NETSCOUT Arbor Edge Defense and APS Systems

Security Target

ST Version: 1.1
December 12, 2019

NETSCOUT Systems, Inc.
310 Littleton Road
Westford, MA 01886-4105

Prepared By:

Booz | Allen | Hamilton
delivering results that endure

Cyber Assurance Testing Laboratory
1100 West St
Laurel, MD 20707

Table of Contents

Table of Figures

Table of Tables

1 Security Target Introduction

This chapter presents the Security Target (ST) identification information and an overview. An ST contains the Information Technology (IT) security requirements of an identified Target of Evaluation (TOE) and specifies the functional and assurance security measures offered by the TOE.

1.1 ST Reference

This section provides information needed to identify and control this ST and its Target of Evaluation.

1.1.1 ST Identification

ST Title:	NETSCOUT Arbor Edge Defense and APS Systems Security Target
ST Version:	1.1
ST Publication Date:	December 12, 2019
ST Author:	Booz Allen Hamilton

1.1.2 Document Organization

Chapter 1 of this document provides identifying information for the ST and TOE as well as a brief description of the TOE and its associated TOE type.

Chapter 2 describes the TOE in terms of its physical boundary, logical boundary, exclusions, and dependent Operational Environment components.

Chapter 3 describes the conformance claims made by this ST.

Chapter 4 describes the threats, assumptions, objectives, and organizational security policies that apply to the TOE.

Chapter 5 defines extended Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs).

Chapter 6 describes the SFRs that are to be implemented by the TSF.

Chapter 7 describes the SARs that will be used to evaluate the TOE.

Chapter 8 provides the TOE Summary Specification, which describes how the SFRs that are defined for the TOE are implemented by the TSF.

1.1.3 Terminology

This section defines the terminology used throughout this ST. The terminology used throughout this ST is defined in Table 1 and 2. These tables are to be used by the reader as a quick reference guide for terminology definitions.

Term	Definition
Administrator	A user who is assigned an Administrator role on the TOE and has the ability to manage the TSF.

Table 1: Customer Specific Terminology

Term	Definition
Security Administrator	The claimed Protection Profile defines a single Security Administrator role that is authorized to manage the TOE and its data. This TOE defines three separate user roles, but only the most privileged role (System Administrator) is authorized to manage all of the TOE's security functionality and is therefore considered to be the Security Administrator for the TOE.
Trusted Channel	An encrypted connection between the TOE and a system in the Operational Environment.
Trusted Path	An encrypted connection between the TOE and the application a Security Administrator uses to manage it (web browser, terminal client, etc.).
User	In a CC context, any individual who has the ability to access the TOE functions or data.

Table 2: CC Specific Terminology

1.1.4 Acronyms

The acronyms used throughout this ST are defined in Table 3. This table is to be used by the reader as a quick reference guide for acronym definitions.

Acronym	Definition
AGD	Administrative Guidance Document
CC	Common Criteria
CLI	Command-Line Interface
cPP	collaborative Protection Profile
CPU	Central Processing Unit
DDoS	Distributed Denial of Service
ESD	Electronic Software Distribution
FTP	File Transfer Protocol
GUI	Graphical User Interface
HTTPS	Hypertext Transfer Protocol Secure
NDcPP	collaborative Protection Profile for Network Devices, version 2.0 + Errata 20180314
NIAP	National Information Assurance Partnership
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
OS	Operating System
PKCS	Public-Key Cryptography Standards
PP	Protection Profile
RBG	Random Bit Generator
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm

SHS	Secure Hash Standard
SSH	Secure Shell
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Function
UI	User Interface

Table 3: Acronym Definition

1.1.5 References

[1] Collaborative Protection Profile for Network Devices + Errata 20180314, version 2.0E

[2] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model CCMB-2012-09-001, Version 3.1 Revision 4, September 2012

[3] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, CCMB-2012-09-002, Version 3.1 Revision 4, September 2012

[4] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components, CCMB-2012-09-003, Version 3.1 Revision 4, September 2012

[5] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, CCMB-2012-09-004, Version 3.1, Revision 4, September 2012

[6] FIPS PUB 140-2, Security Requirements for cryptographic modules, May 25 2001 with change notices (12-03-2002)

[7] FIPS PUB 186-4, Digital Signature Standard (DSS), July 2013

[8] NIST Special Publication SP800-56B Revision 2: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, May 2013

[9] NIST Special Publication SP800-56B Revision 1: Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography, March 2014 References

[10] NIST Special Publication SP800-90A Deterministic Random Bit Generator Validation System (DRBGVS), October 29, 2015

[11] The Secure Hash Algorithm Validation System (SHAVS), Updated: May 21, 2014

[12] FIPS PUB 140-2 Federal Information Processing Standards Publication Security Requirements for Cryptographic Modules May 25, 2001

[13] FIPS PUB 180-3 Federal Information Processing Standards Publication Secure Hash Standard (SHS) October 2008

[14] FIPS PUB 180-4 Federal Information Processing Standards Publication Secure Hash Standard (SHS) March 2012

[15] FIPS PUB 197 Advanced Encryption Standard November 26, 2012

[16] FIPS PUB 198-1 Federal Information Processing Standards Publication The Keyed-Hash Message Authentication Code (HMAC) July 2008

1.2 TOE Reference

The TOE is the NETSCOUT Arbor Edge Defense and APS Systems family of products, which includes the following appliance models:

- APS2600
- APS2800
- AED2600
- AED2800

Each appliance runs software version 6.2.2

1.3 Each appliance runs software version 6.2.2TOE Overview

The NETSCOUT Arbor Edge Defense and APS Systems (referred to as AED/APS from this point forward) is a network device that includes hardware and software. In the evaluated configuration, the TOE is a standalone device that is not deployed as a distributed TOE. The AED/APS is a family of products that provide different performance and capability specifications; the TOE includes the models listed in section 1.2 above. These devices can be accessed locally via serial port and remotely via SSH CLI and TLS/HTTPS web GUI.

The NETSCOUT AED/APS's primary function is to secure the internet data center's edge from threats against availability, specifically from application-layer distributed denial of service (DDoS) attacks. AED/APS deploys at ingress points to an enterprise to detect, block, and report on key categories of Distributed Denial of Service (DDoS) attacks.

AED/APS uses stateless attack detection that allows it to remain functional even during attacks that are designed to cripple stateful devices on the network. If power failures, hardware failures, or software issues affect the AED/APS appliance, the network traffic will bypass through the appliance unaffected. AED/APS can be connected in-line with or without mitigations enabled (inline mode) or out-of-line through a span port or network tap, with no mitigations (monitor mode).

In monitor mode, the appliance is deployed out-of-line through a span port or network tap, which collectively are referred to as monitor ports. The router or switch sends the traffic along its original path and also copies, or mirrors, the traffic to the appliance. AED/APS analyzes the traffic, detects possible attacks, and suggests mitigations but it does not forward traffic.

In an inline deployment, the appliance acts as a physical cable between the Internet and the protected network. All of the traffic that traverses the network flows through the appliance. AED/APS analyzes the traffic, detects attacks, and mitigates the attacks before it sends the traffic to its destination.
The following figure depicts the TOE boundary:

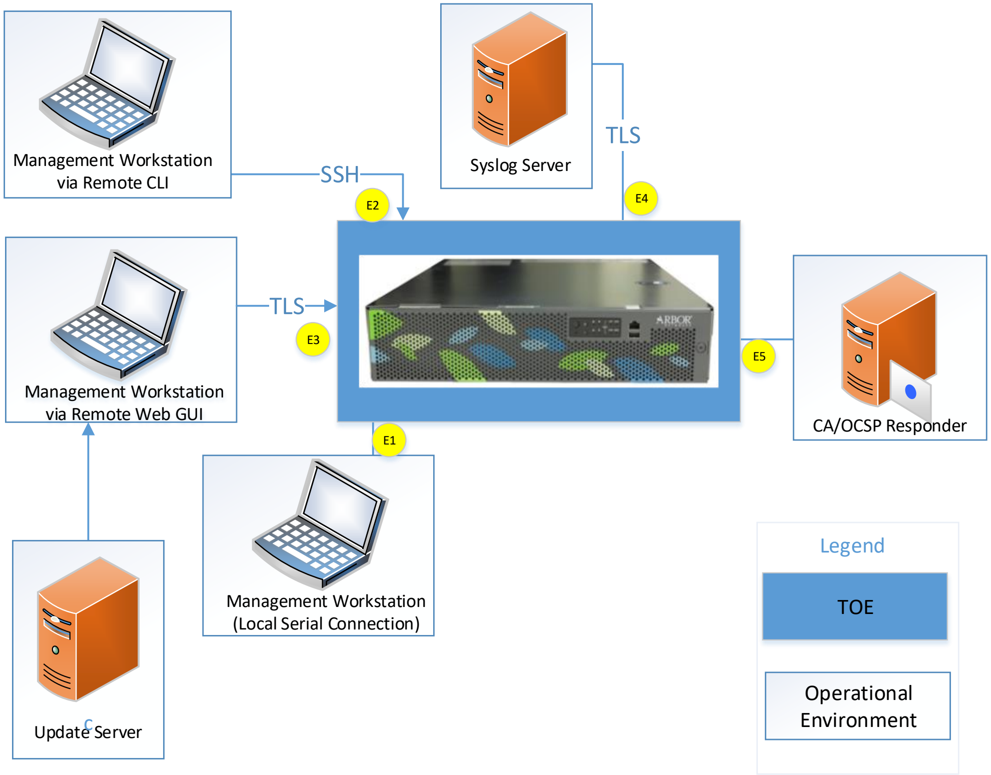
Management Workstation via Remote CLI
SSH
E2
Syslog Server
TLS
E4
Management Workstation via Remote Web GUI
TLS
E3
E5
CA/OCSP Responder
E1
Management Workstation (Local Serial Connection)
Update Server
Legend
TOE
Operational Environment

Figure 1: TOE Boundary for NETSCOUT AED/APS

As illustrated in Figure 1, the TOE is a single hardware device that has management ports, network (or ingress) ports, and tool (or egress) ports. The external interfaces that are relevant to the TOE boundary are the local and remote administrative interfaces, and a syslog server interface (for external audit log storage). The TOE also interfaces with a Certification Authority (CA) for issuance of server certificates and connection to the OCSP Responder to determine the validity of certificates presented to the TOE. An update server (maintained by NETSCOUT), is also required in order for certification testing. The TOE does not directly communicate with the update server and is not considered a TOE external interface.

1.4 TOE Type

The TOE type for this product is Network Device. The product is a hardware appliance whose primary functionality is detect and mitigate DDoS attacks. The *collaborative Protection Profile for Network Devices* [NDcPP] defines a network device as “a device composed of hardware and software that is connected to the network and has an infrastructure role within the network.” The TOE is a network device composed of hardware and software that is connected to the network and accepts traffic, analyzes the traffic, detects attacks, and mitigates the attacks before it sends the traffic to its destination (inline mode). When the appliance is in monitor mode, it will suggest mitigations but will not forward the traffic.

2 TOE Description

This section provides a description of the TOE in its evaluated configuration. This includes the physical and logical boundaries of the TOE.

2.1 Evaluated Components of the TOE

The following table describes the TOE components in the evaluated configuration:

- APS2600
- APS2800
- AED2600
- AED2800

2.2 Components and Applications in the Operational Environment

The following table lists components and applications in the TOE’s operational environment that must be present for the TOE to be operating in its evaluated configuration:

Component	Definition
Certification Authority	A server that acts as a trusted issuer of digital certificates and hosts the OCSP Responders that identifies revoked certificates.
Management Workstation	Any general-purpose computer that is used by an administrator to manage the TOE. The TOE can be managed remotely, in which case the management workstation requires an SSH client to access the CLI or a web browser (Microsoft Internet Explorer 10 or 11, Google Chrome 44, Firefox ESR 31 or 40) to access the web GUI, or locally, in which case the management workstation must be physically connected to the TOE using the serial port and must use a terminal emulator that is compatible with serial communications.
Syslog Server	The syslog server connects to the TOE and allows the TOE to send syslog messages to it for remote storage. This is used to send copies of audit data to be stored in a remote location for data redundancy purposes.
Update Server	A general-purpose computer that is used to store software update packages that can be retrieved by the Security Administrator and downloaded via the management workstation. Software updates, including new versions, are made available to licensed clients through an Electronic Software Distribution system. Access to the ESD server is controlled by the NETSCOUT client services organization and limited to actively licensed clients. Updates are transferred from the management workstation to the TOE via the web GUI upload tool from the management workstation. The TOE does not directly communicate with the update server and is not considered a TOE external interface.

Table 4: Components of the Operational Environment

2.2.1 Excluded from the TOE

The following optional products, components, and/or applications can be integrated with the TOE but are not included in the evaluated configuration. They provide no added security related functionality for the evaluated product. They are separated into three categories: not installed, installed but requires a separate license, and installed but not part of the TSF.

2.2.2 Not Installed

There are no optional components that are omitted from the installation process.

2.2.3 Installed but Requires a Separate License

There are no excluded components that are installed and require a separate license.

2.2.4 Installed but Not Part of the TSF

- **Insecure mode of operation** – AED/APS provides a ‘FIPS mode’ that restricts the cryptographic algorithms and ciphersuites to what is claimed in the Security Target. Operating the product outside of this mode of operation is not within the scope of the TSF.

- **Telnet** – AED/APS supports both Telnet and SSHv2 for remote administration. In the evaluated configuration Telnet will be disabled.

- **SNMP -** The SNMP interface is a read-only interface that is used to output diagnostic information that does not include any TSF data. SNMP will be disabled.

Additionally, the TOE includes a number of functions that are outside the scope of the claimed Protection Profile. These functions are not part of the TSF because there are no SFRs that apply to them.

2.3 Physical Boundary

2.3.1 Hardware

AED/APS is a rack-mounted hardware device. The model specific hardware and their configurations are as follows:

Model	APS2600/AED2600	APS2800/AED2800
Processor	Intel E5-2608L v3 - 2.00GHz	Intel E5-2648L v3 - 1.80GHz
Sockets	2	2
Memory	32 GB	64 GB
OS SSD Capacity	240 GB	240 GB
Cores Per CPU	6	12

Table 5: AED/APS Hardware Specifications

2.3.2 Software

Each TOE appliance operates with AED/APS software version 6.2.2.

Note that the AED/APS software is built on top of Arbux internal OS v7.0 (ArbOS).

2.4 Logical Boundary

The TOE is comprised of the following security features as scoped by the NDcPP:

Security Audit
Cryptographic Support
Identification and Authentication
Security Management

Protection of the TSF
TOE Access
Trusted Path/Channels

2.4.1 Security Audit

Audit records are generated for various types of management activities and events. The audit records include the date and time stamp of the event, the event type and subject identity. In the evaluated configuration, the TSF is configured to transmit audit data to a remote syslog server using TLS. Audit data is also stored locally to ensure availability of the data if communications with the syslog server becomes unavailable. Local audit records are stored in files which are rotated to ensure a maximum limit of disk usage is enforced.

2.4.2 Cryptographic Support

The TOE uses sufficient security measures to protect its data in transmission by implementing cryptographic methods and trusted channels. The TOE uses SSHv2 and TLS/HTTPS to secure the trusted path to the Remote CLI and the web GUI respectively. The TOE also uses TLS to secure the trusted channel to the remote syslog server.

The cryptographic algorithms are provided by a NETSCOUT FIPS Object Module (CERT 3457). Cryptographic keys are generated using the CTR_DRBG provided by this module. The TOE erases all plaintext secret and private keys that reside in both RAM and non-volatile storage by overwriting them with random data. In the evaluated configuration, the TOE operates in “FIPS mode” which is used to restrict algorithms to meet the PP requirements.

The following table contains the CAVP algorithm certificates:

SFR supported	Algorithm	CAVP Cert. #
FCS_CKM.1	ECC (key generation)	1535
FCS_CKM.2	CVL (ECC key establishment)	2056
FCS_COP.1/DataEncryption	AES (encryption/decryption)	5669
FCS_COP.1/SigGen	RSA (signature generation/verification)	3051
FCS_COP.1/SigGen	ECDSA (signature generation/verification)	1535
FCS_COP.1/Hash	SHS (hashing)	4543
FCS_COP.1/KeyedHash	HMAC (keyed-hash message authentication)	3774
FCS_RBG_EXT.1	DRBG (random bit generation)	2291

Table 6: Cryptographic Algorithm Table

2.4.3 Identification and Authentication

All users must be identified and authenticated to the TOE before being allowed to perform any actions on the TOE. This is true of users accessing the TOE via the local console, or protected paths using the remote CLI via SSH or web GUI via TLS 1.2/HTTPS. Users authenticate to the TOE using one of the following methods:

- Username/password (defined on the TOE)

- Username/public key (SSH only)

The TSF provides a configurable number of maximum consecutive authentication failures that are permitted by a user. Once this number has been met, the account is locked until a Security Administrator unlocks it. This behavior is configurable and shared by the CLI and by the web GUI. Passwords that are maintained by the TSF can be composed of upper case, lower case, numbers, and special characters. Password information is never revealed during the authentication process including during login failures. Before a user authenticates to the device, a configurable warning banner is displayed.

As part of establishing trusted remote communications, the TOE provides X.509 certificate functionality. In addition to verifying the validity of certificates, the TSF can check their revocation status using Online Certificate Status Protocol (OCSP). The TSF can also generate a Certificate Signing Request in order to obtain a signed certificate to install for its own use as a TLS server.

2.4.4 Security Management

The TOE defines three roles: System Administrator, DDoS Admin, and System User. Each of these roles has varying levels of fixed privilege to interact with the TSF. The System Administrator role is able to perform all security-relevant management functionality (such as user management, password policy configuration, application of software updates, and configuration of cryptographic settings). Therefore, a user that is assigned this role is considered to be a Security Administrator of the TSF. Management functions can be performed using the local CLI, remote CLI, or web GUI. All software updates to the TOE are performed manually.

2.4.5 Protection of the TSF

The TOE stores usernames and passwords in a password file that cannot be viewed by any user on the TOE regardless of the user's role. The passwords are hashed using SHA-512. Public keys are stored in the configuration database which is integrity checked at boot time. Key data is stored in plaintext on the hard drive but cannot be accessed by any user. The TOE has an underlying hardware clock that is used for keeping time. The time must be manually set in evaluated configuration. Power-on self-tests are executed automatically when the FIPS validated cryptographic module is loaded into memory. The FIPS cryptographic module verifies its own integrity using an HMAC-SHA1 digest computed at build time.

The version of the TOE (both the currently executing version and the installed/updated version, if different) can be verified from any of the administrative interfaces provided by the TSF. All updates are downloaded to a local machine from the vendor website and then loaded on to the TOE. The updated image is verified via a digital signature before installation completes.

2.4.6 TOE Access

The TOE can terminate inactive local console, remote CLI or web GUI sessions after a specified time period. Users can also terminate their own interactive sessions. Once a session has been terminated, the TOE requires the user to re-authenticate to establish a new session. The TOE displays an administratively configured banner on the local console or remote CLI and the web GUI prior to allowing any administrative access to the TOE.

2.4.7 Trusted Path/Channels

The TOE connects and sends data to IT entities that reside in the Operational Environment via trusted channels. In the evaluated configuration, the TOE connects with a remote syslog server using TLS to encrypt the audit data that traverses the channel. When accessing the TOE remotely, administrators interact with the TSF using a trusted path. The remote CLI is protected via SSHv2 and the web GUI is protected by TLS/HTTPS.

3 Conformance Claims

3.1 CC Version

This ST is compliant with Common Criteria for Information Technology Security Evaluation, Version 3.1 Revision 5 April 2017.

3.2 CC Part 2 Conformance Claims

This ST and Target of Evaluation (TOE) is Part 2 extended to include all applicable NIAP and International interpretations through December 12, 2019.

3.3 CC Part 3 Conformance Claims

This ST and Target of Evaluation (TOE) are conformant to Part 3 to include all applicable NIAP and International interpretations through December 12, 2019.

3.4 PP Claims

This ST claims exact conformance to the following Protection Profile:

- collaborative Protection Profile for Network Devices, version 2.0 + Errata 20180314 [NDcPP]

3.5 Package Claims

The TOE claims following Selection-Based SFRs that are defined in the appendices of the claimed PP:

- FCS_HTTPS_EXT.1
- FCS_SSHS_EXT.1
- FCS_TLSC_EXT.1
- FCS_TLSS_EXT.1
- FIA_X509_EXT.1/Rev
- FIA_X509_EXT.2
- FIA_X509_EXT.3

The TOE claims the following Optional SFRs that are defined in the appendices of the claimed PP:

- FMT_MOF.1/Services
- FMT_MTD.1/CryptoKeys
- FMT_MTD/Services

This does not violate the notion of exact conformance because the NDcPP specifically indicates these as allowable selections and options and provides both the ST author and evaluation laboratory with instructions on how these claims are to be documented and evaluated.

3.6 Package Name Conformant or Package Name Augmented

This ST and TOE are in exact conformance with the NDcPP.

3.7 Technical Decisions

Technical Decisions that effected the SFR wording have been annotated with a Footnote.

Technical Decisions were not considered to be applicable if any of the following conditions were true:

- The Technical Decision does not apply to the NDcPP.
- The Technical Decision does not apply to the current version of the NDcPP.
- The Technical Decision applies to an SFR that was not claimed by the TOE.
- The Technical Decision applies to an SFR selection or assignment that was not chosen for the TOE.
- The Technical Decision only applies to one or more Application Notes in the NDcPP and does not affect the SFRs or how the evaluation of the TOE is conducted.

- The Technical Decision is an affirmation that an existing requirement or Evaluation Activity is correct.
- The Technical Decision was superseded by a more recent Technical Decision.
- The Technical Decision is issued as guidance for future versions of the NDcPP

The following list of the NDcPP2e Technical Decisions apply to the TOE because SFR wording, application notes, or assurance activities were modified for SFRs claimed by the TOE:

		Changes			**Analysis to this evaluation**	
TD #	**Title**	**SFR**	**AA**	**Notes**	**NA**	**SFR**
TD0451	NIT Technical Decision for ITT Comm UUID Reference Identifier				X	Not trying to claim UUID.
TD0448	NIT Technical Decision for Documenting Diffie-Hellman 14 groups		X			AA: TSS
TD0447	NIT Technical Decision for Using 'diffie-hellman-group-exchange-sha256' in FCS_SSHC/S_EXT.1.7				X	Not trying to claim diffie-hellman-group-exchange-sha256
TD0425	NIT Technical Decision for Cut-and-paste Error for Guidance AA		X			AA: AGD No changes to ST
TD0423	NIT Technical Decision for Clarification about application of RfI#201726rev2			X		No changes to ST
TD0412	NIT Technical Decision for FCS_SSHS_EXT.1.5 SFR and AA discrepancy		X			AA: Test No changes to ST
TD0411	NIT Technical Decision for FCS_SSHC_EXT.1.5, Test 1 - Server and client side seem to be confused		X		X	AA: Test Not claiming FCS_SSHC_EXT.1
TD0410	NIT technical decision for Redundant assurance activities associated with FAU_GEN.1		X			TSS: AGD No changes to ST
TD0409	NIT decision for Applicability of FIA_AFL.1 to key-based SSH authentication			X		No changes to ST
TD0408	NIT Technical Decision for local vs. remote administrator accounts	X	X	X		AA: TSS and AGD Changes wording to FIA_UAU_EXT.2.1, FIA_AFL.1.1, and FIA_AFL.1.2
TD0407	NIT Technical Decision for handling Certification of Cloud Deployments				X	Not claiming any cloud platforms
TD0402	NIT Technical Decision for RSA-based FCS_CKM.2 Selection	X			X	Not claiming RSA for key establishment.

TD0401	NIT Technical Decision for Reliance on external servers to meet SFRs			X		No changes to ST
TD0400	NIT Technical Decision for FCS_CKM.2 and elliptic curve-based key establishment			X		No changes to ST
TD0399	NIT Technical Decision for Manual installation of CRL (FIA_X509_EXT.2)				X	Do not use CRL
TD0398	NIT Technical Decision for FCS_SSH*EXT.1.1 RFCs for AES-CTR	X				Changes to FCS_SSHS_EXT.1.1
TD0397	NIT Technical Decision for Fixing AES-CTR Mode Tests		X			AA: Test No changes to ST
TD0396	NIT Technical Decision for FCS_TLSC_EXT.1.1, Test 2		X			AA: Test No changes to ST
TD0395	NIT Technical Decision for Different Handling of TLS1.1 and TLS1.2		X		X	AA: Test Not claiming mutual authentication
TD0394	NIT Technical Decision for Audit of Management Activities related to Cryptographic Keys			X		No changes to ST
TD0343	NIT Technical Decision for Updating FCS_IPSEC_EXT.1.14 Tests				X	AA: TSS, AGD, and Test Not claiming IPSEC
TD0342	NIT Technical Decision for TLS and DTLS Server Tests		X			AA: Test Not claiming DTLS / Claiming TLSS
TD0341	NIT Technical Decision for TLS wildcard checking			X		AA: Test No changes to ST
TD0340	NIT Technical Decision for Handling of the basicConstraints extension in CA and leaf certificates	X				Changes to FIA_X509_EXT.1.1 text
TD0339	NIT Technical Decision for Making password-based authentication optional in FCS_SSHS_EXT.1.2	X	X			AA: TSS and Test Changes to FCS_SSHS_EXT.1.2 text
TD0338	NIT Technical Decision for Access Banner Verification		X			AA: TSS No changes to ST
TD0337	NIT Technical Decision for Selections in FCS_SSH*_EXT.1.6	X	X			AA: Test Changes to FCS_SSHS_EXT.1.4 and FCS_SSHS_EXT.1.6 text
TD0336	NIT Technical Decision for Audit requirements for FCS_SSH*_EXT.1.8		X			AA: Test No changes to ST
TD0335	NIT Technical Decision for FCS_DTLS Mandatory Cipher Suites			X	X	Not claiming DTLSC / Claiming TLSC and TLSS No changes to ST
TD0334	NIT Technical Decision for Testing SSH when password-based authentication is not supported		X			AA: Test No changes to ST

TD0333	NIT Technical Decision for Applicability of FIA_X509_EXT.3	X				AA: AGD and Test Changes to FIA_X509_EXT.3.1 text
TD0324	NIT Technical Decision for Correction of section numbers in SD Table 1		X			AA: FSP Evaluation Activities No changes to ST
TD0323	NIT Technical Decision for DTLS server testing - Empty Certificate Authorities list		X		X	AA: Test Not claiming DTLSC
TD0322	NIT Technical Decision for TLS server testing - Empty Certificate Authorities list		X			AA: Test No changes to ST Supersedes TD0262
TD0321	Protection of NTP communications			X		No change to ST.
TD0291	NIT technical decision for DH14 and FCS_CKM.1	X	X			AA: Test Change to FCS_CKM.1.1 text
TD0290	NIT technical decision for physical interruption of trusted path/channel.		X			AA: TSS and Test No changes to ST
TD0289	NIT technical decision for FCS_TLSC_EXT.x.1 Test 5e		X			AA: Test No changes to ST
TD0281	NIT Technical Decision for Testing both thresholds for SSH rekey		X			AA: Test No changes to ST
TD0259	NIT Technical Decision for Support for X509 ssh rsa authentication IAW RFC 6187	X		X		Changes to FCS_SSHS_EXT.1.5 text
TD0257	NIT Technical Decision for Updating FCS_DTLSC_EXT.x.2/FCS_TLSC_EXT.x.2 Tests 1-4		X			AA: Test Not claiming DTLSC / Claiming TLSC Claiming TLSC No changes to ST
TD0256	NIT Technical Decision for Handling of TLS connections with and without mutual authentication		X			AA: Test No changes to ST
TD0228	NIT Technical Decision for CA certificates - basicConstraints validation		X			AA: Test No changes to ST

Table 7: NDcPP V2.0E Technical Decisions

3.8 Conformance Claim Rationale

The NDcPP states the following: “This is a Collaborative Protection Profile (cPP) whose Target of Evaluation (TOE) is a network device… A network device in the context of this cPP is a device composed of both hardware and software that is connected to the network and has an infrastructure within the network… Examples of network devices that are covered by requirements in this cPP include routers, firewalls, VPN gateways, IDSs, and switches.”

The TOE is a network device composed of hardware and software that is designed to provide in-line network protection from DDoS attacks. Therefore, the conformance claim is appropriate.

4 Security Problem Definition

4.1 Threats

This section identifies the threats against the TOE. These threats have been taken from the NDcPP.

Threat	Threat Definition
T.UNAUTHORIZED_ADMINISTRATOR_ACCESS	Threat agents may attempt to gain administrator access to the network device by nefarious means such as masquerading as an administrator to the device, masquerading as the device to an administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between network devices. Successfully gaining administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.
T.WEAK_CRYPTOGRAPHY	Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.
T.UNTRUSTED_COMMUNICATION_CHANNELS	Threat agents may attempt to target network devices that do not use standardized secure tunneling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-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.
T.WEAK_AUTHENTICATION_ENDPOINTS	Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the network device itself could be compromised.
T.UPDATE_COMPROMISE	Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.
T.UNDETECTED_ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the network

	device without administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the administrator would have no knowledge that the device has been compromised.
T.SECURITY_FUNCTIONALITY_COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the network device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the administrator or device credentials for use by the attacker.
T.PASSWORD_CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other network devices.
T.SECURITY_FUNCTIONALITY_FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.

Table 8: TOE Threats

4.2 Organizational Security Policies

This section identifies the organizational security policies which are expected to be implemented by an organization that deploys the TOE. These policies have been taken from the NDcPP.

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

Table 9: Organizational Security Policies

4.3 Assumptions

The specific conditions listed in this section are assumed to exist in the TOE's Operational Environment. These assumptions have been taken from the NDcPP.

Assumption	Assumption Definition
A.PHYSICAL_PROTECTION	The network device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security and/or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP will not include any requirements on physical tamper protection or other physical attack mitigations. The cPP will not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device.
A.LIMITED_FUNCTIONALITY	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality).
A.NO_THRU_TRAFFIC_PROTECTION	A standard/generic network device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the

	network device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the network device, destined for another network entity, is not covered by the NDcPP. It is assumed that this protection will be covered by cPPs for particular types of network devices (e.g., firewall).
A.TRUSTED_ADMINISTRATOR	The Security Administrator(s) for the network device are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The network device is not expected to be capable of defending against a malicious administrator that actively works to bypass or compromise the security of the device.
A.REGULAR_UPDATES	The network device firmware and software is assumed to be updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_CREDENTIALS_SECURE	The administrator's credentials (private key) used to access the network device are protected by the platform on which they reside.
A.RESIDUAL_INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

Table 10: TOE Assumptions

4.4 Security Objectives

This section identifies the security objectives of the TOE and its supporting environment. The security objectives identify the responsibilities of the TOE and its environment in meeting the security needs.

4.4.1 TOE Security Objectives

The NDcPP does not define any security objectives for the TOE.

4.4.2 Security Objectives for the Operational Environment

The TOE's operational environment must satisfy the following objectives:

Objective	Objective Definition
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.
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.NO_THRU_TRAFFIC_PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.RESIDUAL_INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.
OE.TRUSTED_ADMIN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner.

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

Table 11: Operational Environment Objectives

4.5 Security Problem Definition Rationale

The assumptions, threats, OSPs, and objectives that are defined in this ST represent the assumptions, threats, OSPs, and objectives that are specified in the Protection Profile to which the TOE claims conformance.

5 Extended Components Definition

5.1 Extended Security Functional Requirements

The extended Security Functional Requirements that are claimed in this ST are taken directly from the PP to which the ST and TOE claim conformance. These extended components are formally defined in the PP in which their usage is required.

5.2 Extended Security Assurance Requirements

There are no extended Security Assurance Requirements in this ST.

6 Security Functional Requirements

6.1 Conventions

The CC permits four functional component operations—assignment, refinement, selection, and iteration—to be performed on functional requirements. This ST will highlight the operations in the following manner:

- **Assignment:** allows the specification of an identified parameter. Indicated with *italicized* text.
- **Refinement:** allows the addition of details. Indicated with **bold** text. Note that conversion of British spelling to American spelling is not marked as a refinement (e.g. ‘authorisation’ changed to ‘authorization’).
- **Selection:** allows the specification of one or more elements from a list. Indicated with <u>underlined</u> text.
- **Iteration:** allows a component to be used more than once with varying operations. Indicated with a sequential number in parentheses following the element number of the iterated SFR and/or separated by a “/” with a notation that references the function for which the iteration is used, e.g. “/TrustedUpdate” for an SFR that relates to update functionality

When multiple operations are combined, such as an assignment that is provided as an option within a selection or refinement, a combination of the text formatting is used.

If SFR text is reproduced verbatim from text that was formatted in a claimed PP (such as if the PP’s instantiation of the SFR has a refinement or a completed assignment), the formatting is not preserved. This is so that the reader can identify the operations that are performed by the ST author as opposed to the PP author.

6.2 Security Functional Requirements Summary

The following table lists the SFRs claimed by the TOE:

Class Name	Component Identification	Component Name
Security Audit (FAU)	FAU_GEN.1	Audit Data Generation
	FAU_GEN.2	User Identity Association
	FAU_STG_EXT.1	Protected Audit Event Storage
Cryptographic Support (FCS)	FCS_CKM.1	Cryptographic Key Generation
	FCS_CKM.2	Cryptographic Key Establishment
	FCS_CKM.4	Cryptographic Key Destruction
	FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)
	FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)
	FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)
	FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)
	FCS_HTTPS_EXT.1	HTTPS Protocol
	FCS_RBG_EXT.1	Random Bit Generation
	FCS_SSHS_EXT.1	SSH Server Protocol
	FCS_TLSC_EXT.1	TLS Server Protocol
	FCS_TLSS_EXT.1	TLS Server Protocol
Identification and Authentication (FIA)	FIA_AFL.1	Authentication Failure Management
	FIA_PMG_EXT.1	Password Management
	FIA_UAU_EXT.2	Password-based Authentication Mechanism
	FIA_UAU.7	Protected Authentication Feedback
	FIA_UIA_EXT.1	User Identification and Authentication

	FIA_X509_EXT.1/Rev	X509 Certificate Validation
	FIA_X509_EXT.2	X509 Certificate Authentication
	FIA_X509_EXT.3	X509 Certificate Requests
Security Management (FMT)	FMT_MOF.1/ManualUpdate	Management of Security Functions Behavior
	FMT_MOF.1/Services	Management of Security Functions Behavior
	FMT_MTD.1/CoreData	Management of TSF Data
	FMT_MTD.1/CryptoKeys	Management of TSF Data
	FMT_SMF.1	Specification of Management Functions
	FMT_SMR.2	Restrictions on Security Roles
Protection of the TSF (FPT)	FPT_APW_EXT.1	Protection of Administrator Passwords
	FPT_SKP_EXT.1	Protection of TSF Data (for reading of all symmetric keys)
	FPT_STM_EXT.1	Reliable Time Stamps
	FPT_TST_EXT.1	TSF Testing
	FPT_TUD_EXT.1	Trusted Update
TOE Access (FTA)	FTA_SSL_EXT.1	TSF-Initiated Session Locking
	FTA_SSL.3	TSF-Initiated Termination
	FTA_SSL.4	User-Initiated Termination
	FTA_TAB.1	Default TOE Access Banners
Trusted Path/Channels (FTP)	FTP_ITC.1	Inter-TSF Trusted Channel
	FTP_TRP.1/Admin	Trusted Path

Table 12: Security Functional Requirements for the TOE

6.3 Security Functional Requirements

6.3.1 Class FAU: Security Audit

6.3.1.1 *FAU_GEN.1 Audit Data Generation*

FAU_GEN.1.1

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

a) Start-up and shut-down of the audit functions;

b) All auditable events for the not specified level of audit; and

c) All administrative actions comprising:

- Administrative login and logout (name of user account shall be logged if individual user accounts are required for administrators).
- Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).
- Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).
- Resetting passwords (name of related user account shall be logged).
- [[Starting and stopping services]];

d) Specifically defined auditable events listed in Table **13**.

FAU_GEN.1.2

The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, information specified in column three of Table **13**.

Requirement	Auditable Event(s)	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_HTTPS_EXT.1	Failure to establish an HTTPS session.	Reason for failure.
FCS_RBG_EXT.1	None.	None.
FCS_SSHS_EXT.1	Failure to establish an SSH session.	Reason for failure.
FCS_TLSC_EXT.1	Failure to establish a TLS session.	Reason for failure.
FCS_TLSS_EXT.1	Failure to establish a TLS session.	Reason for failure.
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP Address).
FIA_PMG_EXT.1	None.	None.
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP Address).
FIA_UAU.7	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP Address).
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate.	Reason for failure.
FIA_X509_EXT.2	None.	None.
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update**.**	None.
FMT_MOF.1/Services	Starting and stopping of services	None
FMT_MTD.1/CoreData	All management activities of TSF data.	None.
FMT_MTD.1/CryptoKeys	Management of cryptographic keys.	None.
FMT_SMF.1	None.	None.
FMT_SMR.2	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process.	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure).	No additional information.
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_SSL_EXT.1	The termination of a remote session by the session locking mechanism.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None.

Table 13: Auditable Events

6.3.1.2 *FAU_GEN.2 User Identity Association*

FAU_GEN.2.1

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

6.3.1.3 *FAU_STG_EXT.1 Protected Audit Event Storage*

FAU_STG_EXT.1.1

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

FAU_STG_EXT.1.2

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

FAU_STG_EXT.1.3

The TSF shall [overwrite previous audit records according to the following rule: [*delete the oldest of 11 syslog files*]] when the local storage space for audit data is full.

6.3.2 Class FCS: Cryptographic Support

6.3.2.1 *FCS_CKM.1 Cryptographic Key Generation*

FCS_CKM.1.1[1]

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

- 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 Diffie-Hellman group 14 that meet the following: RFC 3526, Section 3].

6.3.2.2 *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: [

[1] TD0291

- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”;
- Key establishment scheme using Diffie-Hellman group 14 that meet the following: RFC 3526, Section 3].

6.3.2.3 *FCS_CKM.4 Cryptographic Key Destruction*

FCS_CKM.4.1

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

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

that meets the following: No Standard.

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

FCS_COP.1.1/DataEncryption

The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in [CBC, CTR, 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, CTR as specified in ISO 10116, GCM as specified in ISO 19772].

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

FCS_COP.1.1/SigGen

The TSF shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [

- RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [*2048 bits*]]
- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [*256 bits and 384 bits*]]

That meet the following: [

- For RSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital **S**ignature scheme 2 or Digital Signature scheme 3
- For ECDSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing “NIST curves” [P-256, P-384, P512]; ISO/IEC 14888-3, Section 6.4].

6.3.2.6 *FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)*

FCS_COP.1.1/Hash

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

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

FCS_COP.1.1/KeyedHash

The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm [HMAC-SHA-1, HMAC-SHA-256, HMAC-384, HMAC-SHA-512] and cryptographic key sizes [*160 bits, 256 bits, 384 bits, 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".

6.3.2.8 *FCS_HTTPS_EXT.1 HTTPS Protocol*

FCS_HTTPS_EXT.1.1

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

FCS_HTTPS_EXT.1.2

The TSF shall implement the HTTPS protocol using TLS.

FCS_HTTPS_EXT.1.3

If a peer certificate is presented, the TSF shall [not require client authentication] if the peer certificate is deemed invalid.

6.3.2.9 *FCS_RBG_EXT.1 Cryptographic Operation (Random Bit Generation)*

FCS_RBG_EXT.1.1

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

FCS_RBG_EXT.1.2

The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [[*2*] software-based noise source**s**] 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.

6.3.2.10 *FCS_SSHS_EXT.1 SSH Server Protocol*

FCS_SSHS_EXT.1.1[2]

The TSF shall implement the SSH protocol that complies with RFC(s) [4251, 4252, 4253, 4254, 4344, 5656, 6668].

FCS_SSHS_EXT.1.2[3]

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 [*262,144*] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4[4]

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

[2] TD0398
[3] TD0339
[4] TD0337

FCS_SSHS_EXT.1.5[5]

The TSF shall ensure that the SSH public-key based authentication implementation uses [ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6[6]

The TSF shall ensure that the SSH transport implementation uses [hmac-sha1, hmac-sha2-256, hmac-sha2-512, implicit] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7

The TSF shall ensure that [diffie-hellman-group14-sha1, 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.

6.3.2.11 *FCS_TLSC_EXT.1 TLS Client Protocol*

FCS_TLSC_EXT.1.1

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

- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_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_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289]

FCS_TLSC_EXT.1.2

The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125 section 6.

FCS_TLSC_EXT.1.3

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

FCS_TLSC_EXT.1.4

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

[5] TD0259

[6] TD0337

6.3.2.12 ***FCS_TLSS_EXT.1 TLS Server Protocol***

FCS_TLSS_EXT.1.1

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

- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_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_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289]

FCS_TLSS_EXT.1.2

The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and [TLS 1.1].

FCS_TLSS_EXT.1.3

The TSF shall [generate EC Diffie-Hellman parameters over NIST curves [secp256r1, secp384r1, secp521r1] and no other curves; generate Diffie-Hellman parameters of size [2048 bits]].

6.3.3 **Class FIA: Identification and Authentication**

6.3.3.1 ***FIA_AFL.1 Authentication Failure Management***

FIA_AFL.1.1[7]

The TSF shall detect when an Administrator configurable positive integer within [*1 to 999,999,999*] unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password.

FIA_AFL.1.2[8]

When the defined number of unsuccessful authentication attempts has been met, the TSF shall [prevent the offending Administrator from successfully establishing remote session using any authentication method that involves a password until [*a manual unlock of the account*] is taken by an Administrator].

6.3.3.2 ***FIA_PMG_EXT.1 Password Management***

FIA_PMG_EXT.1.1

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

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

b) Minimum password length shall be configurable to [*7 characters*] and [*72 characters*].

[7] TD0408

[8] TD0408

6.3.3.3 *FIA_UAU_EXT.2 Password-Based Authentication Mechanism*

FIA_UAU_EXT.2.1[9]

The TSF shall provide a local [password-based] authentication mechanism to perform local administrative user authentication.

6.3.3.4 *FIA_UAU.7 Protected Authentication Feedback*

FIA_UAU.7.1

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

6.3.3.5 *FIA_UIA_EXT.1 User Identification and Authentication*

FIA_UIA_EXT.1.1

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

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

FIA_UIA_EXT.1.2

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

6.3.3.6 *FIA_X509_EXT.1/Rev X.509 Certificate Validation*

FIA_X509_EXT.1.1/Rev[10]

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

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

FIA_X509_EXT.1.2/Rev

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

[9] TD0408

[10] TD0340

6.3.3.7 *FIA_X509_EXT.2 X.509 Certificate Authentication*

FIA_X509_EXT.2.1

The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [TLS], 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 [allow the Administrator to choose whether to accept the certificate in these cases].

6.3.3.8 *FIA_X509_EXT.3 X.509 Certificate Requests*

FIA_X509_EXT.3.1[11]

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

FIA_X509_EXT.3.2

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

6.3.4 Class FMT: Security Management

6.3.4.1 *FMT_MOF.1/ManualUpdate Management of Security Functions Behavior*

FMT_MOF.1.1/ManualUpdate

The TSF shall restrict the ability to enable the functions to perform manual update to Security Administrators.

6.3.4.2 *FMT_MOF.1/Services Management of Security Functions Behavior*

FMT_MOF.1.1/Services

The TSF shall restrict the ability to enable and disable functions and services to Security Administrators.

6.3.4.3 *FMT_MTD.1/CoreData Management of TSF Data*

FMT_MTD.1.1/CoreData

The TSF shall restrict the ability to manage the TSF data to the Security Administrators.

6.3.4.4 *FMT_MTD.1/CryptoKeys Management of TSF Data*

FMT_MTD.1.1/CryptoKeys

The TSF shall restrict the ability to manage the cryptographic keys to Security Administrators.

6.3.4.5 *FMT_SMF.1 Specification of Management Functions*

FMT_SMF.1.1

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

- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;

[11] TD0333

- Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- [Ability to configure the cryptographic functionality;
- Ability to re-enable an Administrator account;
- Ability to set the time which is used for time-stamps].

6.3.4.6 *FMT_SMR.2 Restrictions on Security Roles*

FMT_SMR.2.1

The TSF shall maintain the roles:

- Security Administrator.

FMT_SMR.2.2

The TSF shall be able to associate users with roles.

FMT_SMR.2.3

The TSF shall ensure that the conditions

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

are satisfied.

6.3.5 Class FPT: Protection of the TSF

6.3.5.1 *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.

6.3.5.2 *FPT_SKP_EXT.1 Protection of TSF Data (For Reading of All Pre-shared, Symmetric and Private Keys)*

FPT_SKP_EXT.1.1

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

6.3.5.3 *FPT_STM_EXT.1 Reliable Time Stamps*

FPT_STM_EXT.1.1

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

FPT_STM_EXT.1.2

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

6.3.5.4 *FPT_TST_EXT.1 TSF Testing*

FPT_TST_EXT.1.1

The TSF shall run a suite of the following self-tests [during initial start-up (on power on)] to demonstrate the correct operation of the TSF: [*software integrity, cryptographic module integrity, cryptographic known-answer tests, continuous RNG test*].

6.3.5.5 *FPT_TUD_EXT.1 Trusted Update*

FPT_TUD_EXT.1.1

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

FPT_TUD_EXT.1.2

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

FPT_TUD_EXT.1.3

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

6.3.6 Class FTA: TOE Access

6.3.6.1 *FTA_SSL_EXT.1 TSF-initiated Session Locking*

FTA_SSL_EXT.1.1

The TSF shall, for local interactive sessions,

- [terminate the session]

after a Security Administrator-specified time period of inactivity.

6.3.6.2 *FTA_SSL.3 TSF-initiated Termination*

FTA_SSL.3.1

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

6.3.6.3 *FTA_SSL.4 User-initiated Termination*

FTA_SSL.4.1

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

6.3.6.4 *FTA_TAB.1 TOE Access Banner*

FTA_TAB.1.1

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

6.3.7 Class FTP: Trusted Path/Channels

6.3.7.1 *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, [no other capabilities] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_ITC.1.2

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

FTP_ITC.1.3

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

6.3.7.2 ***FTP_TRP.1/Admin Trusted Path***

FTP_TRP.1.1/Admin

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/Admin

The TSF shall permit remote Administrators to initiate communication via the trusted path.

FTP_TRP.1.3/Admin

The TSF shall require the use of the trusted path for initial Administrator authentication and all remote administration actions.

6.4 Statement of Security Functional Requirements Consistency

The Security Functional Requirements included in the ST represent all required SFRs specified in the claimed PP, a subset of the optional requirements, and all applicable selection-based requirements that have been included as specified for the claimed PP.

7 Security Assurance Requirements

This section identifies the Security Assurance Requirements (SARs) that are claimed for the TOE. The SARs which are claimed are in exact conformance with the NDcPP.

7.1 Class ADV: Development

7.1.1 Basic Functional Specification (ADV_FSP.1)

7.1.1.1 *Developer action elements:*

ADV_FSP.1.1D

The developer shall provide a functional specification.

ADV_FSP.1.2D

The developer shall provide a tracing from the functional specification to the SFRs.

7.1.1.2 *Content and presentation elements:*

ADV_FSP.1.1C

The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2C

The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3C

The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

ADV_FSP.1.4C

The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

7.1.1.3 *Evaluator action elements:*

ADV_ FSP.1.1E

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_ FSP.1.2E

The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

7.2 Class AGD: Guidance Documentation

7.2.1 Operational User Guidance (AGD_OPE.1)

7.2.1.1 *Developer action elements:*

AGD_OPE.1.1D

The developer shall provide operational user guidance.

7.2.1.2 *Content and presentation elements:*

AGD_OPE.1.1C

The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C

The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C

The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C

The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C

The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C

The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C

The operational user guidance shall be clear and reasonable.

7.2.1.3 *Evaluator action elements:*

AGD_OPE.1.1E

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

7.2.2 Preparative Procedures (AGD_PRE.1)

7.2.2.1 *Developer action elements:*

AGD_PRE.1.1D

The developer shall provide the TOE including its preparative procedures.

7.2.2.2 *Content and presentation elements:*

AGD_ PRE.1.1C

The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_ PRE.1.2C

The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

7.2.2.3 *Evaluator action elements:*

AGD_ PRE.1.1E

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_ PRE.1.2E

The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

7.3 Class ALC: Life Cycle Supports

7.3.1 Labeling of the TOE (ALC_CMC.1)

7.3.1.1 *Developer action elements:*

ALC_CMC.1.1D
The developer shall provide the TOE and a reference for the TOE.

7.3.1.2 *Content and presentation elements:*

ALC_CMC.1.1C
The TOE shall be labeled with its unique reference.

7.3.1.3 *Evaluator action elements:*

ALC_CMC.1.1E
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

7.3.2 TOE CM Coverage (ALC_CMS.1)

7.3.2.1 *Developer action elements:*

ALC_CMS.1.1D
The developer shall provide a configuration list for the TOE.

7.3.2.2 *Content and presentation elements:*

ALC_CMS.1.1C
The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

ALC_CMS.1.2C
The configuration list shall uniquely identify the configuration items.

7.3.2.3 *Evaluator action elements:*

ALC_CMS.1.1E
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

7.4 Class ATE: Tests

7.4.1 Independent Testing - Conformance (ATE_IND.1)

7.4.1.1 *Developer action elements:*

ATE_IND.1.1D
The developer shall provide the TOE for testing.

7.4.1.2 *Content and presentation elements:*

ATE_IND.1.1C
The TOE shall be suitable for testing.

7.4.1.3 *Evaluator action elements:*

ATE_IND.1.1E

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.1.2E

The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

7.5 Class AVA: Vulnerability Assessment

7.5.1 Vulnerability Survey (AVA_VAN.1)

7.5.1.1 *Developer action elements:*

AVA_VAN.1.1D

The developer shall provide the TOE for testing.

7.5.1.2 *Content and presentation elements:*

AVA_VAN.1.1C

The TOE shall be suitable for testing.

7.5.1.3 *Evaluator action elements:*

AVA_VAN.1.1E

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.1.2E

The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.1.3E

The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

8 TOE Summary Specification

The following sections identify the security functions of the TOE and describe how the TSF meets each claimed SFR. They include Security Audit, Cryptographic Support, Identification and Authentication, Security Management, Protection of the TSF, TOE Access, and Trusted Path/Channels.

8.1 Security Audit

8.1.1 FAU_GEN.1

The TOE contains mechanisms to generate audit data based on the behavior that occurs with the TSF. For example, the TOE generates audit records for startup and shutdown of audit functions (this equates to startup and shutdown of system), administrative functions performed on the TOE, and communication handling. For a full list of the evaluated audit records that the TOE produces, see Table 13 above. Each audit record contains identifying information of the subject performing the action. The audit records are generated and stored in the form of syslog records which are sent to a remote syslog server. The transmission of this data to the syslog server is protected from unauthorized modification and deletion using TLSv1.2.

The audit records that the TOE creates include the following information: date and time of the event, event type, subject identity, success or failure of the event, source of the event and any additional audit information as specified in the right column of Table 13.

Below is an example of an audit record that is generated by the TOE for a cryptographic key generation and a brief description of each field.

```
Apr 11 11:32:30 arbor-aps2800.catl.local auth: COMMAND services ssh key generate BY admin FROM cli VIA 192.168.1.3
Apr 11 11:32:30 arbor-aps2800.catl.local auditcomsh: User: admin  Command: / services ssh key generate
Apr 11 11:32:32 arbor-aps2800.catl.local ssh: User admin successfully generated host key
```

Date and Time – Apr 11 11:32:30
Event Type - services ssh key generate
Subject Identity - User: admin
Success or Failure of the event - User admin successfully generated host key
Source of the Event - FROM cli VIA 192.168.1.3

Note: For a full list of the audit events samples generated by the TOE, please refer to the Supplemental Administrative Guidance Document (AGD).

8.1.2 FAU_GEN.2

The TOE records the identity of the user (e.g. username, system name, IP address) associated with each audited event in the audit record.

8.1.3 FAU_STG_EXT.1

In the evaluated configuration, the TOE will send audit records to a remote syslog server via an encrypted TLS channel that is provided by the OpenSSL FIPS cryptographic module v2.0.8. When the TOE is configured to send data to the syslog server, the audit records immediately pushed to the syslog server. If syslog server connectivity is unavailable, audit records will only be stored locally. Upon re-establishment of communications with the syslog server, new audit records will resume being transmitted to it but the audit records that were generated during the time the syslog server connection was down remain stored locally and are not sent to the syslog server.

The TSF has a fixed audit log rotation method for storing and automatically deleting old records. The TOE allocates 64 MB for up to 11 separate log files to store audit data. When the first audit log is full, a second is automatically created. This process is repeated until 11 files have been created. Once the 11th file is full, the first file is deleted, and another is created to be populated with the newest audit data.

8.2 Cryptographic Support

The TOE contains its own cryptographic module software called NETSCOUT FIPS Object Module v1.0. The NETSCOUT FOM is based on OpenSSL v2.0.8. This cryptographic module is the same on both models of the NETSCOUT AED/APS and performs the functionality described within this section.

8.2.1 FCS_CKM.1

The TOE generates Elliptic Curve (ECC) keys using NIST curve P-256, P-384 and P-521 in accordance with FIPS PUB 186-4. The ECC keys are generated in support of TLS key establishment. Additionally, the TOE uses FFC Diffie-Hellman group 14 for SSH key establishment that meets RFC 3526, Section 3.

The TOE’s key generation function has the ECC certificate #1535.

8.2.2 FCS_CKM.2

The TOE implements a NIST SP 800-56A conformant key establishment mechanism for Elliptic Curve. Specifically, the TOE complies with the NIST SP 800-56A key agreement scheme (KAS) primitives that are defined in section 6.1.2.2 of the SP. These key establishment schemes used for the establishment of TLS sessions, for which the TOE can act as both a client and a server. In addition, the TOE implements Diffie-Hellman group 14 for SSH key establishment, for which the TOE is a server. The group, prime, hexadecimal value, and generator are all consistent with RFC 3526, section 3.

The TOE’s implementation of NIST SP 800-56A has CVL certificate #2056.

8.2.3 FCS_CKM.4

The TOE destroys all plaintext secret and private cryptographic keys in persistent storage by overwriting the storage location the keys occupy with random data. This is accomplished using a file system API that overwrites the memory with random data before deallocating the file in which the key data is stored.

Keys stored in volatile memory are immediately destroyed upon deallocation using the function *cleanse()*, to overwrite the keys with random data. This combined approach protects the keys from being compromised. There are no known instances where key destruction does not happen as defined. The following table identifies the keys and CSPs that are applicable to the TOE as well as their usage, storage location, and method of destruction:

Key Material	Origin	Storage Location	Clearing of Key Material
SSH keys	SSH server/ client application	Non-volatile storage/file system	Synchronously write random data to the file.
Authentication keys	X.509 certificates	Non-volatile storage/ file system	Synchronously write random data to the file.
TLS session keys	syslogtls, radsec applications	RAM	For RAM, overwrite with random data.

Table 14: Cryptographic Materials, Storage, and Destruction Methods

8.2.4 FCS_COP.1/DataEncryption

The TOE performs encryption and decryption using the AES algorithm in CBC, CTR and GCM mode with key sizes of 128 and 256 bits. The AES algorithm meets ISO 18033-3 and the CBC and CTR mode implementations meet ISO 10116.

This algorithm implementation has CAVP AES certificates #5669.

8.2.5 FCS_COP.1/SigGen

In accordance with FIPS PUB 186-4, the TOE provides cryptographic digital signature verification using RSA Digital Signature Algorithm (rDSA). The TOE supports rDSA with a key size of 2048 bits. The TOE also performs digital signature generation and verification using ECDSA in accordance with FIPS PUB 186-4. The TOE supports ECDSA key sizes of 256 and 384 bits and NIST curves of P-256, P-384 and P-521.

This implementation has CAVP RSA certificate #3051 and ECDSA certificate #1535.

8.2.6 FCS_COP.1/Hash

The TOE provides cryptographic hashing services using SHA-1, SHA-256, SHA-384 and SHA-512 with message digest sizes of 160, 256, 384, and 512 bits respectively, as specified in FIPS PUB 180-4. The TSF uses hashing services the following functions:

- SHA-1, SHA-256, and SHA-512 used for SSH data integrity
- SHA-1, 256, 384 used for TLS support
- SHA-512 used for password hashing

The SHA algorithm meets ISO/IEC 10118-3:2004 and has CAVP SHS certificate #4543.

8.2.7 FCS_COP.1/KeyedHash

The TOE provides keyed-hashing message authentication services using HMAC-SHA-1 (160-bit output MAC), HMAC-SHA-256 (256-bit output MAC), HMAC-SHA-384 (384-bit output MAC) and HMAC-SHA-512 (512-bit output MAC). The HMAC implementation supports key sizes that are equal to block sizes. HMAC is implemented as specified in FIPS PUB 198-1 and FIPS PUB 180-3.

The algorithm meets ISO/IEC 9797-2:2011 and has CAVP HMAC certificate #3774.

8.2.8 FCS_HTTPS_EXT.1

The TOE implements HTTPS in order to facilitate remote administration over the web GUI. The HTTPS implementation conforms to RFC 2818 and uses the TLS server implementations specified in FCS_TLSS_EXT.1. Since the HTTPS server does not enforce TLS mutual authentication, the only prerequisite to establishment of a TLS connection is that the peer initiates the communication.

8.2.9 FCS_RBG_EXT.1

The TOE implements a counter mode deterministic random bit generator (CTR_DRBG (AES)). The DRBG used by the TOE is in accordance with ISO/IEC 18031:2011. The TOE models uniformly provide two software-based entropy sources as described in the proprietary entropy specification. The DRBG is seeded with a minimum of 256 bits of entropy so that it is sufficient to ensure full entropy for 256-bit keys, which are the largest keys generated by the TSF.

The NETSCOUT FIPS Object Module v1.0 collects entropy from /dev/random, which is a blocking entropy source. The /dev/random entropy pool is protected by being in kernel memory and is not accessible from user space. The entropy source is described in greater detail in the proprietary Entropy Assessment Report (EAR).

The TOE’s DRBG implementation is validated under CAVP, certificate #2291.

8.2.10 FCS_SSHS_EXT.1

SSHv2 is used to secure the remote CLI management connection (SSH Server). The traffic for the SSH connection is sent via the Ethernet Management Port and by default, the SSHv2 port used is port 22. The TOE implements the SSHv2 protocol that complies with the following RFCs: 4251, 4252, 4253, 4254, 4344, 5656, and 6668. The TOE supports password based and public-key based authentication methods as described in RFC 4252; both username/password and public key authentication methods are supported on the remote CLI. The TOE supports ssh-rsa, ecdsa-sha2-nistp256 and ecdsa-sha2-nistp384, ecdsa-sha2-nistp521 for the public key algorithm.

Session keys are created when the TOE establishes an SSHv2 connection. The TOE will monitor the time period during which the SSHv2 session keys are active and how much data has been transmitted using them. The TOE performs an SSH rekey no longer than 1 hour elapsing or after no more than 1 GB of data is transferred whichever comes first.

As SSH packets are being received, the TOE uses a buffer to build all packet information. All SSHv2 connections will be dropped upon detection of any packet greater than 262,144 bytes being transported, as described in RFC 4253. Data encryption is provided by the aes128-ctr, aes256-ctr, aes128-gcm@openssh.com, and aes256-gcm@openssh.com algorithms.

The hmac-sha1, hmac-sha2-256, and hmac-sha2-512 algorithms are used for data integrity. When aes*-gcm@openssh.com is negotiated as the encryption algorithm, the MAC algorithm field is ignored and GCM is implicitly used as the MAC.

The key exchange methods used in SSHv2 are diffie-hellman-group14-sha1, ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521.

The TOE has a “FIPS mode” to limit the SSH connection parameters to those defined in the evaluated configuration.

8.2.11 FCS_TLSC_EXT.1/FCS_TLSS_EXT.1

The TOE uses the TLS 1.2 protocol to secure the following connections and channels: web GUI management interface connection using HTTPS (TLS Server) and for the secure connection to the external syslog server. When the TOE is operating in “FIPS mode”, TLS uses only the following ciphersuites:

- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_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_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

Note: The TOE supports this full list. However, when an RSA certificate is loaded, only the RSA ciphers will be available. When ECDSA certificate is loaded only the ECDSA ciphers will be available.

When the TOE uses TLS client functionality, the presented identifier for the server certificate has to match the reference identifier in order to establish the connection. The TOE supports either the hostname or IP address as reference identifiers. Wildcards cannot be used when defining the reference identifier on the TOE. In the evaluated configuration, the TOE’s TLS implementation is configured to present the Supported Elliptic Curves Extension in the Client Hello using NIST curves secp256r1, secp384r1, and secp521r1. Certificate pinning is not supported. When certificate validation fails, the connection is not established.

When the TOE uses TLS server functionality, it will reject all connection attempts from TLS versions other than 1.2. The TOE generates EC Diffie-Hellman parameters over NIST curves secp256r1, secp384r1, secp521r1 and generates Diffie-Hellman parameters of 2048 bits for key establishment.

8.3 Identification and Authentication

8.3.1 FIA_AFL.1

The TSF provides configurable counters for consecutive failed authentication attempts that will result in a user account being locked due to the failure counter being reached. The CLI and the web GUI have a single shared counter that adds to the total count. This setting is configurable only in the CLI and can be set between 1 and 999,999,999 attempts. The default setting is 5 attempts. While the user account is locked, no authentication is possible. The account will remain locked until a Security Administrator manually unlocks it.

All remote accounts are subject to the account lockout mechanism. To prevent total lockout, a local Security Administrator can manually unlock any locked out accounts.

8.3.2 FIA_PMG_EXT.1

Passwords maintained by the TSF can be composed using any combination of upper case and lower case letters, numbers, and special characters including the following: “!”,”@”,”#”,”$”,”%”,”^”,”&”,”*”,”(“,”)”. The password policy is configurable by the Security Administrator via the CLI and supports the minimum password length of 7 characters and a maximum password length of 72 characters.

8.3.3 FIA_UAU.7

While authenticating to the TOE, whether it be via the CLI or web GUI, the TOE does not echo any password data that is entered. In the case that a user enters invalid credentials (valid/invalid username or valid/invalid password), the TOE does not reveal any information about the invalid component of the credential.

8.3.4 FIA_UIA_EXT.1/FIA_UAU_EXT.2

Users can authenticate to the TOE locally or remotely. Local users log onto the local console via a workstation, using a terminal emulator that is compatible with serial communications, physically connected to the serial port, using a username and password. Remote users can log in to the TOE using SSH to access the remote CLI using either username and password or SSH public key via the Ethernet Management Port. User authentication information used is sent remotely and is protected using SSHv2. Users may also authenticate remotely to a web GUI that is protected using TLS/HTTPS via the Ethernet Management Port.

All authentication credentials are verified using the TOE’s local authentication mechanism. When public key authentication is used, the TOE authenticates users by verifying the message the TOE receives from the SSH client using the message's associated public key stored on the TOE.

In the evaluated configuration, the warning banner is displayed prior to the user authenticating to the TOE via the local console, the remote CLI, and the web GUI. The display of the warning banner is the only service that can be run prior to authentication and thus, the TOE does not allow a user to perform any other actions prior to authentication, regardless of the interface used.

8.3.5 FIA_X509_EXT.1/ FIA_X509_EXT.2/ FIA_X509_EXT.3

The TOE performs certificate validity checking for outbound TLS connections to the syslog server. The TSF determines the validity of certificates to use by ensuring that the certificate and the certificate path are valid in accordance with RFC 5280. As per RFC5280, the TOE supports the minimum length of three certificates in the path. In addition, the certificate path is terminated in a trusted CA certificate, the basicConstraints extension is present, and the CA flag is set to TRUE for all CA certificates. The TSF also ensures that the extendedKeyUsage field includes the correct purpose for its intended use. The only supported use of X.509 certificates is for Server Authentication for TLS server certificates used in FTP_ITC.1 defined connections. If the validation of the certificate fails, the certificate is rejected, and the connection is not established.

In addition to the validity checking that is performed by the TOE, the TSF will validate certificate revocation status using Online Certificate Status Protocol (OCSP) as specified in RFC 6960. The Security Administrator can configure the TOE to accept or reject a certificate in the event the OCSP responder is unavailable via the CLI. By default, if the OCSP responder is unavailable to determine the validity of the certificate, the TOE rejects the certificate.

In order to support TLS/HTTPS connectivity over the web GUI, the TSF provides the ability, via the CLI, to generate a Certificate Request Message as specified by RFC 2986 so that its server certificate can be signed by a Certification Authority. The message includes Common Name, Organization, Organizational Unit, and Country values. The certificate chain of the Certificate Response is validated by the TSF prior to being installed as the TOE's server certificate.

8.4 Security Management

8.4.1 FMT_MOF.1/ManualUpdate

Software updates on the TOE can only be performed using the CLI. TOE software updates are published to the customer portal on the NETSCOUT website and downloaded by the customer to their local system. Once downloaded, the update is loaded onto the TOE and applied manually. Security Administrator is the only user capable of performing this function. Updating the software of the TOE is only able to be performed via the CLI.

8.4.2 FMT_MOF.1/Services

The TOE restricts the ability to enable and disable the services to the Security Administrator. Section 8.4.4 describes the TOE's definition of the Security Administrator.

8.4.3 FMT_MTD.1/CoreData

The TOE provides three user roles: System Administrator, DDoS Admin, and System User. The TSF uses role-based access control to assign each user account to one or more roles, each of which has a fixed set of privileges to interact with the product. Of these roles, only the System Administrator role is authorized to perform each of the management functions associated with the TSF. The DDoS Admin is able to perform a subset of the management functions. The System Administrator role is therefore functionally identical to the 'Security Administrator' as defined by the NDcPP.

The only security-relevant TOE functionality that is available to a user prior to authentication is the display of the warning banner.

8.4.4 FMT_MTD.1/CryptoKeys

The Security Administrator is the only role that is permitted to manipulate cryptographic data on the TOE. Within the TSF, this behavior is limited to the:

- generation of SSH host keys for the TOE SSH server and
- generation and import/removal of X.509 certificates.

8.4.5 FMT_SMF.1

Security Administrators are capable of performing management functions on the TOE locally and remotely. Security Administrators can perform management functions locally via a workstation, using a terminal emulator that is compatible with serial communications, physically connected to the serial port, or remotely via the remote CLI or web GUI. The following table lists the TSF management functions and identifies the interface(s) that can be used to perform them:

Management Function	Local CLI	Remote CLI	Remote GUI
Configure TLS Connection Parameters	X	X	
Configure SSH Connection Parameters	X	X	
Configure Failed Lockout Threshold	X	X	
Create Users	X	X	X
Unlock User Account	X		
Modify User Passwords	X	X	X
Modify Password Policy	X	X	
Generate X.509 Certificate Request Message	X	X	
Initiate Manual Update	X	X	
Configure System Time	X	X	
Configure Idle Session Timeout	X	X	
Configure Banner Text			X

Table 15: Management Functions by Interface

Only the Security Administrator can modify the text displayed in the TOE's login banners, set the values for session inactivity before termination, and initiate manual updates to the TOE's software after verifying the digital signature of the update.

The Security Administrator is also able to configure the TOE’s cryptographic functionality. This is accomplished by entering a command into the CLI which places the TOE into "FIPS mode" of operation. When this command is entered, the TOE limits the cryptographic algorithms to meet the requirements defined within the Security Target. The administrator will then need to complete the configuration of the SSH, TLS, and certificate functionality in order to bring the TOE into complete conformance.

8.4.6 FMT_SMR.2

The security management functions available to authorized users of the TOE are mediated by a role-based access control system. The role-based access control system is enforced via the local console and remotely via the remote CLI or web GUI. The TOE has three claimed user roles: System Administrator, DDoS Admin, and System User. Only the System Administrator and DDoS Admin are capable of performing management functions on the TOE. All SFR relevant management activity is performed by the System Administrator, role which corresponds to the NDcPP’s definition of Security Administrator. Users in the System User role only have read access to view events and blocked host queries via web GUI.

Custom user roles are created based on privileges by the Security Administrator via the CLI. The Security Administrator can assign users to the custom role on either the web GUI or the CLI. Only one role can be assigned to a user, however, a role can be assigned to multiple users.

8.5 Protection of the TSF

8.5.1 FPT_APW_EXT.1

While the TOE is in FIPS mode, the TOE stores usernames and passwords in a password file. All passwords stored on the TOE are stored in hashed form using SHA-512. The password file cannot be viewed by any user on the TOE regardless of the user's role.

8.5.2 FPT_SKP_EXT.1

All key data (public/secret/private) is stored in plaintext on the hard drive but cannot be accessed via the local console, remote CLI (secure shell does not provide root level access to OS), or the web GUI by any user.

8.5.3 FPT_STM_EXT.1

The TOE has an underlying hardware clock that is used for keeping time. A Security Administrator must set the clock’s time manually in the evaluated configuration. The TSF uses time data for the following purposes:

- Audit record timestamps
- Inactivity timeout for administrative sessions
- Expiration checking for certificates

8.5.4 FPT_TST_EXT.1

The TOE relies on the standard OpenSSL functionality for FIPS Module and Integrity Tests as defined in Sections 2.2, 2.3, 2.4 and Section 6.3 Self Tests of the *OpenSSL User Guide for the Open/SSL FIPS object Module v2.0*.

During the start-up of the TOE, the TSF runs a cryptographic module integrity test as well as a cryptographic known-answer test and continuous RNG tests. These tests verify the software integrity and that the cryptographic module is operating correctly and has not been tampered with. If a cryptographic self-test fails, the device will become inoperable. The only remediation to this test failure is to call the technical support team.

The TOE also performs a file system self-test during start-up to ensure the underlying hardware does not have any anomalies that would cause the software to be executed in an unpredictable or inconsistent manner. This test calls the standard test method in the ext3 file system to check the file system data structures and confirm that they are in a consistent state.

In the event that a power on self-test fails, the boot process will terminate. The TOE will need to be rebooted to attempt to clear the error. If the TOE has been corrupted or the hardware has failed such that rebooting will not resolve the issue, a Security Administrator will need to contact NETSCOUT support. These tests and their response to failures is sufficient to ensure that the TSF behaves as described in the ST because it would detect any unauthorized modifications to the TOE, failures or tampering of the hardware (which could be an attempt to compromise its storage or take the TOE out of the range of operating conditions specified for its entropy source), and any cryptographic failures that could result in the establishment of insecure trusted channels.

8.5.5 FPT_TUD_EXT.1

The Security Administrator can query the current version of the TOE's firmware/software on the local/remote CLI interface and the web GUI interface. On the local console and remote CLI interface, a user can enter the “`system version`” command to view the currently installed version. Within the web GUI interface, the user can select "About” link to display the currently installed version.

Software updates, including new versions, are made available to licensed clients through an Electronic Software Distribution system (ESD). Access to the ESD server is controlled by the NETSCOUT client services organization and limited to actively licensed clients.

In order to update the TOE, the Security Administrator manually downloads the update from the customer portal on the NETSCOUT website to their local management workstation and then applied to the TOE following documented procedures. Once downloaded, the update is loaded onto the TOE using the web GUI. At this point, the update is not installed on the TOE only loaded into the filesystem. Packages uploaded onto the TOE can be displayed using the “`system files show`” from the CLI. The Security Administrator then authenticates to the TOE via the CLI and manually performs installation of the update. Updates are signed using OpenSSL and verified against a public key which is shipped with the TOE. The signature is checked automatically during the installation process. Packages with invalid or missing signatures cannot be installed.

8.6 TOE Access

8.6.1 FTA_SSL_EXT.1

The TOE is designed to terminate a local session after a specific period of time. By default, this setting is disabled. In the evaluated configuration, this setting must be enabled by the Security Administrator. The inactivity timeout can be configured between 0-999 minutes. The value of 0 means that this setting is disabled and there is no timeout configured. Once a session has been terminated, the local user must re-authenticate to start a new session.

8.6.2 FTA_SSL.3

The TOE can be configured to terminate remote interactive sessions that are inactive in two different ways. In the event that the inactivity setting is met while users are logged into the CLI, the TOE tears down the SSH connection. This setting can be configured between 0-999 minutes. The value of 0 means that this setting is disabled and there is no timeout configured.

In the event that the inactivity setting is reached while a user is logged into the web GUI, the user’s session will end. The setting is configurable; the value of 0 means that this setting is disabled. There is no timeout configured by default.

The Security Administrator users authenticated to the local console or remote CLI may configure this setting for the local console, remote CLI, and web GUI. However, Security Administrator users authenticated to the web GUI can only configure the timeout setting for the web GUI.

8.6.3 FTA_SSL.4

A Security Administrator is able to terminate their own session by entering the "exit" command when logged into the local console or remote CLI. A Security Administrator can terminate their own session by clicking on the "Log Out" link when logged into the web GUI.

8.6.4 FTA_TAB.1

There are three possible ways to authenticate to the TOE: local console, remote CLI, and web GUI. Each of these interfaces has a configurable login banner that is displayed prior to the user authenticating to the TOE.

The pre-login banner is configured through the web GUI and is applied to all authentication options.

8.7 Trusted Path/Channels

8.7.1 FTP_ITC.1

The TOE connects and sends audit data to a syslog server that resides in the Operational Environment via trusted channel. In the evaluated configuration, the TOE connects with a syslog server using TLS v1.2 (only TLS v1.2 is supported) to encrypt the audit data that traverses the channel. The remote syslog server is authenticated using TLS server certificates.

The TOE initiates communication as the client using the TLS cryptographic protocol in the manner described by FCS_TLSC_EXT.1. This protocol is used to protect the data traversing the channel from disclosure and/or modification.

8.7.2 FTP_TRP.1/Admin

Security Administrator users are required to authenticate to the TOE in order to be able to perform any management functions. By initiating the trusted path via the web GUI or remote CLI, Security Administrator users can perform management activities remotely. The web GUI path is protected by TLS/HTTPS (only TLS v1.2 is supported) and the remote CLI is protected using only SSHv2. These protocols are used to protect the data traversing the channel from disclosure and/or modification.