Security Target

SevOne Network Management System 5.5.0.1

Common Criteria Evaluation with
Network Device Protection Profile v1.1 Errata #3,

Document Version: 1.6
Date: August 23, 2016

Prepared For:

SevOne Inc.

4550 New linden hill Road, Suite 300
Wilmington, DE 19808, USA

Prepared By:

CGI Global IT Security Labs.

1410 Blair Place, 7th floor
Ottawa, ON K1J 9B9, Canada
www.cgi.com/securitylab

Revision History

Ver #	Description of changes	Modified by	Date
0.1	Draft to the lab and updated document to Errata #3	D Freebourne	March 5, 2015
0.2	Updated following input from SevOne	D Freebourne	March 11, 2015
0.3	ORs updates and corrections from the lab	D Freebourne	June 23, 2015
0.4	ORs, version update and corrections from the lab and SevOne	D Freebourne	June 4, 2015
0.5	ORs updates and corrections from the lab	D Freebourne	August 28, 2015
0.6	ORs update and corrections from the lab and SevOne	D.Freebourne	January 22, 2016
0.7	ORs update and corrections from the lab	D.Freebourne	March 1, 2016
0.8	ORs update and corrections from the lab and SevOne	D.Freebourne	March 10, 2016
0.9	ORs update and corrections from the lab	D.Freebourne	June 10, 2016
1.0	ORs update and corrections from the lab and SevOne	D.Freebourne	July 8, 2016
1.1	Updated Section 6.1.2.9 (cipher suites). Sections 6.1.4.1 and 7.4.1 were updated (use of special characters)	M.Mulligan	July 8, 2016
1.2	Updated Section 6.1.2.3 (cipher suites). Sections 6.1.4.1 was updated (use of special characters)	M.Mulligan	July 19, 2016
1.3	ORs update and corrections from the lab	D.Freebourne	July 27, 2016
1.4	Section 1.5.1.1 updated. Removed 'protocol failure' audit message paragraph below table 7.1. Upgrade process updated in section 7.5.1	M.Mulligan	August 10, 2016
1.5	Section 1.5.4 updated with Installer information. Updated Section 1.5.1.1	M.Mulligan	August 12, 2016
1.6	Updated Sections 1.4 and 7.5.1.	M.Mulligan	August 23, 2016

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

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

1.1 ST Reference

ST Title	SevOne Network Management System 5.5.0.1
ST Revision	1.6
ST Publication Date	August 23, 2016
ST Author	CGI Global IT Security Labs – Canada
	D. Freebourne

1.2 Target of Evaluation Reference

TOE Developer	SevOne Inc.
TOE Name	SevOne Network Management System 5.5.0.1
TOE Version	5.5.0.1

1.3 Conventions

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

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

1.4 TOE Overview

This section primarily addresses the physical and logical components of the TOE included in the evaluation.

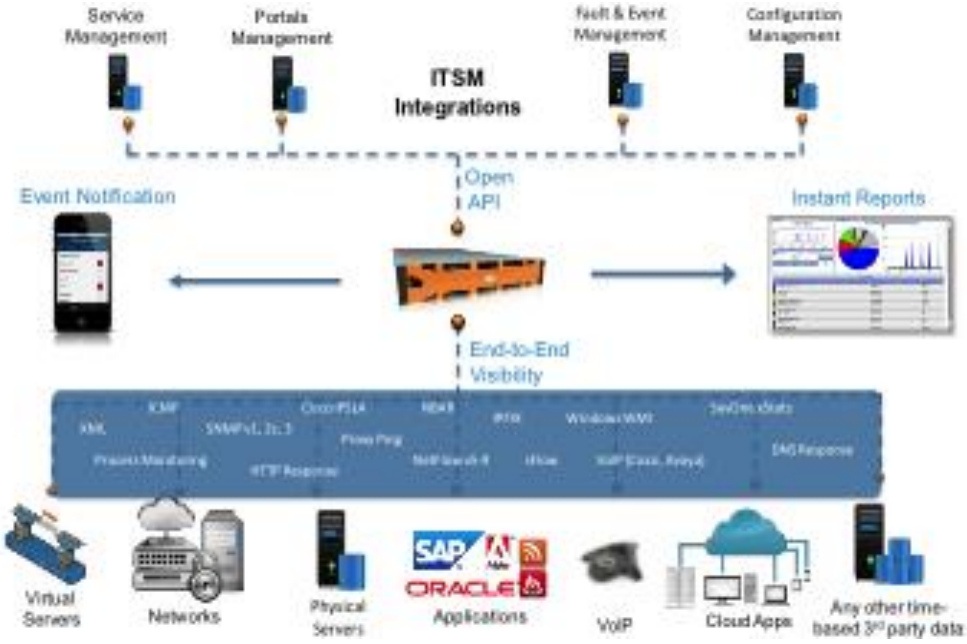
The SevOne Network Management System 5.5.0.1 (NMS) software enables polling and monitoring of current network activity and allows analysis of historic network activity. SevOne NMS 5.5.0.1 maintains a one year detailed history, to establish an audit trail and the baselines necessary to validate service levels, identify trends, and manage costs. SevOne NMS 5.5.0.1 has the ability to capture, store, and manage performance statistics down to the sub-second level, and to dynamically alert network managers for anomalies, which is vital in markets with real time applications. The SevOne NMS 5.5.0.1 can effectively manage networks of all sizes and including all types of network devices, such as routers, switches, servers, printers, etc.

The SevOne Network Management System 5.5.0.1 is a combination of hardware and software used to monitor and administer a network, performing the key collecting, reporting and data storage functions:

1. Monitor health and status of network devices
2. Fault detection
3. Alerts to conditions that impact system performance
4. Collect stats over a period of time to facilitate performance reporting

The Performance Appliance Solution (PAS) appliances monitor and poll network devices, collecting network device application performance information. The information collected is used to generate performance reports and graphs. Each PAS maintains data of those devices assigned to it. Users can administer the local PAS appliance using the web-based user interface. A user must first authenticate to access the user interface. All user interaction with the UI is protected over an HTTPS connection.

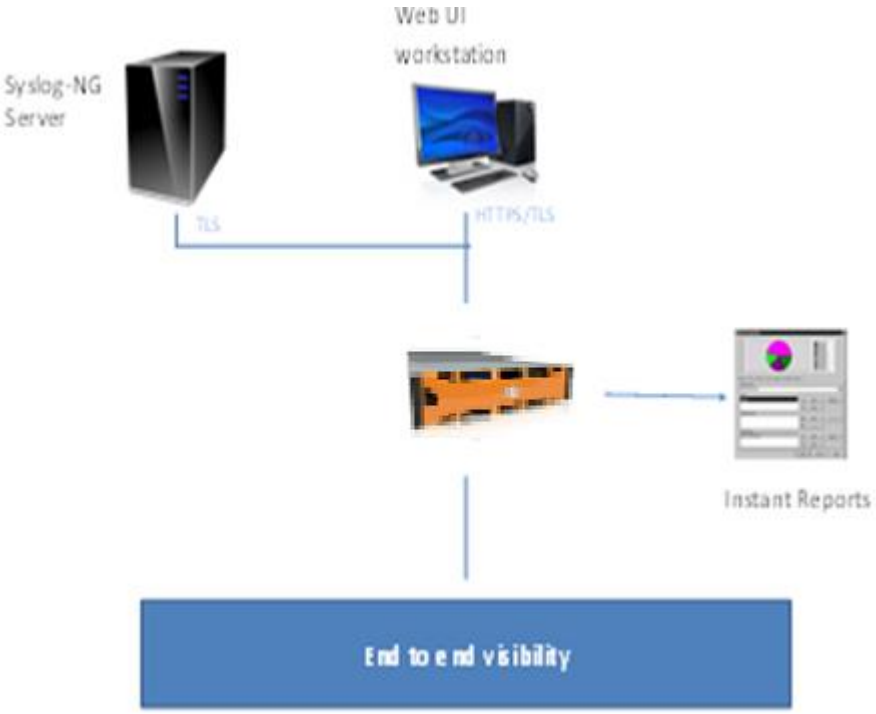
Figure 1 – Typical standalone NMS appliance deployment scenario



The PAS SevOne NMS 5.5.0.1 appliances can operate as a standalone. The in the evaluated configuration the TOE acts as a stand-alone appliance with all functionality deployed off a single system

The following figure (Figure 2) depicts the necessary components in the operational environment to support the evaluated configuration of the NMS deployment (showing a single PAS deployment scenario), which includes the external audit server (syslog-NG).

Figure 2 – NMS evaluated configuration



The SevOne NMS 5.5.0.1 appliances included within the scope of this evaluation are:

- PAS: 10K, 20K, 60K, 200K

The appliance model sizes indicate the number of objects the appliance is capable of monitoring (i.e. the PAS 10K can monitor up to 10,000 objects, where an object can be a fast Ethernet port on a router).

No additional software, hardware, or external databases are needed for the SevOne NMS 5.5.0.1 to operate. The SevOne NMS 5.5.0.1 appliances make use of a variety of protocols to gather information from various devices down the network hierarchy, e.g. SNMP, NetFlow, NBAR, J-Flow, sFlow, IPFIX, JMX (Java Management Extensions), WMI (Windows Management Instrumentation), meaning no SevOne agent is required in network device in order to monitor the device. The appropriate protocol can be selected for the device, depending on the device type and the supported application.

The TOE also includes Installer version "2016-07-22-1778e64" (file seveone-gui-installer-2016-07-22-1778e64.phar) shipped with the Common Criteria appliance. It compares SHA-256 hash value of the downloaded upgrade file with the one in associated manifest file on the SevOne web site to ensure a secure upgrade.

1.5 TOE Description

This section primarily addresses the physical and logical components of the TOE included in the evaluation.

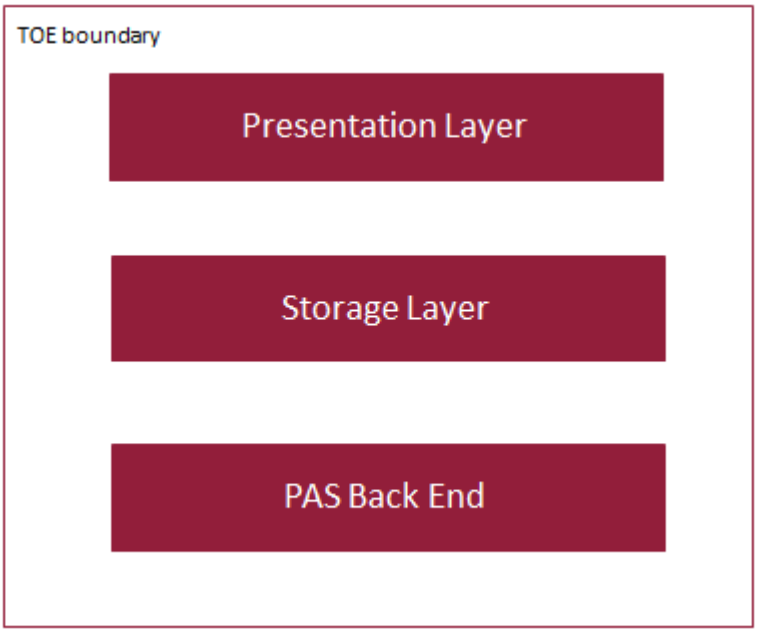
1.5.1 Physical Boundary

The physical scope of the TOE is the SevOne monitoring appliance. The appliance is comprised of three main components:

- PAS backend – interface to rest of network, communicating with network devices
- Storage layer – store data collected from monitored devices
- Presentation layer – web based UI to administer the NMS, including a PDF generator for reports and report scheduling for sending email reports

The physical boundary of the TOE is depicted in Figure 3 below, with the outer box representing the physical boundary of the TOE. The TOE is configured as a standalone appliance (as shown in Figure 1 and Figure 2 above).

Figure 3 – TOE Physical Boundary



The SevOne NMS 5.5.0.1 appliance is based on Linux 3.10.33, using OpenSSL 2.0.4 (FIPS). Apache version 2.4.3 is used for the implementation of the web-based UI.

The SevOne NMS 5.5.0.1 appliances included in the evaluation are comprised of the physical components detailed in Table 1

Table 1 – TOE Appliances[1]

[1] The details of the next-generation hardware platform processors are TBD, but will be using the E5-26xx series chips featuring the Intel RNG optimizations

SevOne NMS 5.5.0.1 Appliance	Processor	Memory	Hard Drive	Network Adaptor
PAS10K	E5-2609v2 Xeon PE R620 2.4Ghz	16GB	4x146GB PERC H710P RAID	4xGigabit Ethernet NIC
PAS20K	E5-2630v2 Xeon PE R720 2.3Ghz	64GB	6x600GB PERC H710P RAID	4xGigabit Ethernet NIC
PAS60K	E5-2630v2 Xeon PE R720 2. 3Ghz	64GB	6x600GB PERC H710P RAID 365GB Fusion-io SSD Drive	4xGigabit Ethernet NIC
PAS200K	E5-2680v2 Xeon PE R720 2.8Ghz	256GB	24x300GB PERC H710P RAID 785GB Fusion-io SSD Drive	4xGigabit Ethernet NIC

1.5.1.1 Guidance Documentation

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

- SevOne NMS 5.5 User Guide 5.5[UG]
- SevOne NMS 5.5 System Administration Guide (Includes Manage Users Workflows) 5.5[AG]
- SevOne NMS 5.5 Installation Guide 5.5[IG]
- SevOne Implementation Guide 5.5[IMG]
- SevOne NMS What's New in 5.5.0? Version 5.4.X vs. Version 5.5.0[WNG]
- SevOne Network Management System v5.5.0.1 Operational User Guidance and Preparative Procedures[AGD]

1.5.2 Logical Boundary

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

1.5.2.1 Security Audit

SevOne NMS 5.5.0.1 generates audit logs for configured security relevant administrative and system actions, including network level events relating to secure connections. The system provides a time stamp that is used to record when each log event was generated. The logs are protected from unauthorized deletion and can be exported to a configured external syslog server.

1.5.2.2 Cryptographic Support and trusted channels

SevOne NMS 5.5.0.1 integrates the FIPS-certified OpenSSL version 2.0.4 cryptographic module for use in all cryptographic services.

Management connections to the web UI are protected by HTTPS/TLS, export of audit logs is performed over TLS protected.

1.5.2.3 User Data Protection

The network interfaces of the SevOne NMS 5.5.0.1 appliances are implemented to ensure leakage of data between network packets through the reuse of memory resources is not possible. This is achieved using the standard Linux driver which utilizes a ring buffer to interface with data going to the physical media. When network data is destined for the NIC driver the kernel creates socket buffer data structures which are of a specific length, and are zeroized upon creation. This socket buffer will then have a copy of the data written to the physical hardware media copied to it via a DMA interrupt.

1.5.2.4 Identification and Authentication, and TOE Access

SevOne NMS 5.5.0.1 supports local and remote authentication of administrators. Administrators must be authenticated before they are permitted access to any administrative function.

An access banner is displayed to the user on initial connection to the web-based UI, presenting a consent warning message regarding use of the UI.

Following a configured period of inactivity the web-based UI session is terminated, requiring re-authentication of the user. The user may also select to terminate the session by selecting the logout option.

1.5.2.5 Security Management

SevOne NMS 5.5.0.1 provides for local GUI access via a dedicated physical connection on one of the 4 NIC ports on the appliance. Access to administrative functions via this connection is protected by password and the cable between the workstation and the appliance is to be physically protected. It also supports a remote web User Interface through which the user can administer the local appliance.

1.5.2.6 Protection of the TSF

An administrator is able to query the current version of the SevOne NMS 5.5.0.1 software and initiate the download of an update from SevOne web servers. The administrator can verify the integrity of the downloaded update, prior to installation, using the published hash associated with the downloaded package.

SevOne NMS 5.5.0.1 checks the integrity of its own operation by performing a set of self-tests during initial start-up. SevOne NMS 5.5.0.1 protects its critical data by ensuring only obfuscated passwords are stored and that keys cannot be read out by administrator or other user of the appliance.

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

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

- External syslog servers
- Client workstation used to access the web UI

1.5.4 Product Physical/Logical Features and Functions not included in the TOE Evaluation

Features/Functions that are not supported in the evaluated configuration of the TOE are:

- LDAP, TACACS+ and Radius authentication
- Distributed communications with remote PAS, DNC or HSA appliances
- Secure interface for SevOne technical support staff to remotely access the TOE (requiring authentication)[2]

[2] This will be disabled in the evaluated configuration.

2 CONFORMANCE CLAIMS

2.1 Common Criteria Conformance Claim

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

2.2 Protection Profile Conformance Claim

The Security Target is conformant to the:

- Network Devices Protection Profile (NDPP) v1.1, June 8, 2013, including the following optional requirements HTTPS, TLS.
- The NDPP Errata #3, 3 November 2014
- Clarification to the Entropy Documentation and Assessment Annex, February 20 2014 due to limited access to the design and raw entropy data of these third-party sources.

3 SECURITY PROBLEM DEFINITION

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

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

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

3.1 Threats

This section identifies the threats to the assets against which protection is required by the TOE or by the security environment. The threat agents are divided into two categories:

- Attackers who are not TOE users: They have public knowledge of how the TOE operates and are assumed to possess a low skill level, limited resources to alter TOE configuration settings or parameters and no physical access to the TOE.
- TOE users: They have extensive knowledge of how the TOE operates and are assumed to possess a high skill level, moderate resources to alter TOE configuration settings or parameters and physical access to the TOE. (TOE users are, however, assumed not to be willfully hostile to the TOE.)

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

Table 2 – Threats

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

3.2 Organizational Security Policies

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

Table 3 – Organizational Security Policies

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

3.3 Assumptions

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

Table 4 – Assumptions

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

4 SECURITY OBJECTIVES

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

4.1 Security Objectives for the TOE

The IT security objectives for the TOE are as follows:

Table 5 – TOE Security Objectives

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

4.2 Security Objectives for the Operational Environment

The security objectives for the operational environment are addressed below:

Table 6 – Operational Environment Security Objectives

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

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

5 EXTENDED SECURITY REQUIREMENT COMPONENTS DEFINITION

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

5.1 Extended TOE Security Functional Requirement Components

This section specifies the extended SFRs for the TOE.

5.1.1 FAU_STG_EXT.1 External Audit Trail Storage

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

Management: FAU_STG_EXT.1

There are no management activities foreseen.

Audit: FAU_STG_EXT.1

There are no auditable events foreseen.

FAU_STG_EXT.1 External Audit Trail Storage

Hierarchical to:	No other components
Dependencies:	FAU_GEN.1 Audit data generation FTP_ITC.1 Inter-TSF trusted channel
FAU_STG_EXT.1.1	The TSF shall be able to [selection: transmit the generated audit data to an external IT entity, receive and store audit data from an external IT entity] using a trusted channel implementing the [selection: IPsec, TLS, TLS/HTTPS] protocol.

5.1.2 FCS_CKM_EXT.4 Cryptographic Key Zeroization

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

Management: FCS_CKM_EXT.4

There are no management activities foreseen.

Audit: FCS_CKM_EXT.4

There are no auditable events foreseen.

FCS_CKM_EXT.4Cryptographic Key Zeroization

Hierarchical to:	No other components
Dependencies:	FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or

	FCS_CKM.1 Cryptographic key generation
FCS_CKM_EXT.4.1	The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

5.1.3 FCS_HTTPS_EXT.1 Extended: HTTPS

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

Management: FCS_HTTPS_EXT.1

There are no management activities foreseen.

Audit: FCS_HTTPS_EXT.1

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

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

FCS_HTTPS_EXT.1 Extended: HTTPS

Hierarchical to:	No other components
Dependencies:	FCS_TLS_EXT.1 Extended: TLS
FCS_HTTPS_EXT.1.1	The TSF shall implement the HTTPS protocol that complies with RFC 2818.
FCS_HTTPS_EXT.1.2	The TSF shall implement the HTTPS protocol using TLS as specified in FCS_TLS_EXT.1.

5.1.4 FCS_TLS_EXT.1 Extended: TLS

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

Management: FCS_TLS_EXT.1

There are no management activities foreseen.

Audit: FCS_TLS_EXT.1

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

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

FCS_TLS_EXT.1 Extended: TLS

Hierarchical to:	No other components
Dependencies:	FCS_COP.1(1) Cryptographic operation (for data encryption/decryption) FCS_COP.1(2) Cryptographic operation (for cryptographic signature) FCS_COP.1(3) Cryptographic operation (for cryptographic hashing)

FCS_COP.1(4) Cryptographic operation (for keyed-hash message authentication)
FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM_EXT.4 Cryptographic Key Zeroization

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

Mandatory Ciphersuites:
TLS_RSA_WITH_AES_128_CBC_SHA

Optional Ciphersuites:
[selection:
None

TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_ SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
].

5.1.5 FCS_RBG_EXT.1 Extended: Random Bit Generation

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

Management: FCS_RBG_EXT.1

There are no management activities foreseen.

Audit: FCS_RBG_EXT.1

There are no auditable events foreseen.

FCS_RBG_EXT.1 Extended: Random Bit Generation

Hierarchical to: No other components

Dependencies: None

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with [selection, choose one of: NIST16 Special Publication 800-90 using [selection: Hash_DRBG(any), HMAC DRBG (any), CTR_DRBG (AES20), Dual_EC_DRBG (any)]; FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using AES] seeded by an entropy source that accumulated entropy from [selection, one or both of: a software-based noise source; a TSF-hardware-based noise source].

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

5.1.6 FIA_PMG_EXT.1 Password Management

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

Management: FIA_PMG_EXT.1

There are no management activities foreseen.

Audit: FIA_PMG_EXT.1

There are no auditable events foreseen.

FIA_PMG_EXT.1 Password Management

Hierarchical to: No other components

Dependencies: None

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

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

5.1.7 FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism

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

Management:FIA_UAU_EXT.2

There are no management activities foreseen.

Audit:FIA_UAU_EXT.2

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

a) All use of the authentication mechanisms.

FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism

Hierarchical to:	No other components
Dependencies:	None
FIA_UAU_EXT.2.1	The TSF shall provide a local password-based authentication mechanism, [selection: [assignment: *other authentication mechanism(s)*], none] to perform user authentication.

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

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

Management: FIA_UIA_EXT.1

There are no management activities foreseen.

Audit: FIA_UIA_EXT.1

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

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

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

Hierarchical to:	FIA_UID.1 Timing of identification FIA_UAU.1 Timing of Authentication
Dependencies:	FTA_TAB.1
FIA_UIA_EXT.1.1	The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process: o Display the warning banner in accordance with FTA_TAB.1; o [selection: no other actions, [assignment: *list of services, actions performed by the TSF in response to non-TOE requests.*]]
FIA_UIA_EXT.1.2	The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

5.1.9 FPT_APW_EXT.1 Extended: Protection of Administrator Passwords

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

Management: FPT_APW_EXT.1

There are no management activities foreseen.

Audit: FPT_APW_EXT.1

There are no audit activities foreseen.

FPT_APW_EXT.1 Extended: Protection of Administrator Passwords

Hierarchical to:	No other components
Dependencies:	None
FPT_APW_EXT.1.1	The TSF shall store passwords in non-plaintext form.
FPT_APW_EXT.1.2	The TSF shall prevent the reading of plaintext passwords.

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

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

Management: FPT_SKP_EXT.1

There are no management activities foreseen.

Audit: FPT_SKP_EXT.1

There are no audit activities foreseen.

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

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

5.1.11 FPT_TST_EXT.1 Extended: TSF testing

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

Management: FPT_TST_EXT.1

There are no management activities foreseen.

Audit: FPT_TST_EXT.1

There are no audit activities foreseen.

FPT_TST_EXT.1 TSF testing

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

5.1.12 FPT_TUD_EXT.1 Extended: Management of TSF Data

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

Management: FPT_ TUD_EXT.1

There are no management activities foreseen.

Audit: FPT_ TUD_EXT.1

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

a) Initiation of update.

FPT_ TUD_EXT.1 Extended: Trusted Update

Hierarchical to:	No other components
Dependencies:	[selection: FCS_COP.1(2) Cryptographic operation (for cryptographic signature), FCS_COP.1(3) Cryptographic operation (for cryptographic hashing)]
FPT_TUD_EXT.1.1	The TSF shall provide security administrators the ability to query the current version of the TOE firmware/software.
FPT_TUD_EXT.1.2	The TSF shall provide security administrators the ability to initiate updates to TOE firmware/software.
FPT_TUD_EXT.1.3	The TSF shall provide a means to verify firmware/software updates to the TOE using a [selection: digital signature mechanism, published hash] prior to installing those updates.

5.1.13 FTA_SSL_EXT.1 Extended: TSF-initiated Session Locking

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

Management: FTA_SSL_EXT.1

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

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

Audit: FTA_SSL_EXT.1

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

a) Any attempts at unlocking an interactive session.

FTA_SSL_EXT.1 Extended: TSF-initiated Session Locking

Hierarchical to:	No other components
Dependencies:	FIA_UIA_EXT.1 Password-based Authentication and Identification Mechanism
FTA_SSL_EXT.1.1	The TSF shall, for local interactive sessions, [selection:

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

after a Security Administrator-specified time period of inactivity.

5.2 Extended TOE Security Assurance Requirement Components

There are no extended TOE Security Assurance Requirement Components.

6 SECURITY REQUIREMENTS

This section defines the Security Functional Requirements (SFRs) and Security Functional Assurance Requirements (SARs) met by the TOE. All the components have been drawn from the Network Device Protection Profile (NDPP) v1.1, Errata #3 of the NDPP and clarifications and interpretations made with NDPP Errata #3.

6.1 Security Functional Requirements

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

Table 7 – TOE Security Functional Requirements

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

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

6.1.1 Security Audit (FAU)

6.1.1.1 FAU_GEN.1 Audit Data Generation

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

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

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

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

Table 8 – Auditable Events

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

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

6.1.1.2 FAU_GEN.2 User Identity Association

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

6.1.1.3 FAU_STG_EXT.1 External Audit Trail Storage

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

6.1.2 Cryptographic Support (FCS)

6.1.2.1 FCS_CKM.1 Cryptographic Key Generation

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

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

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

6.1.2.2 FCS_CKM_EXT.4 Cryptographic Key Zeroization

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

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

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

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

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

FCS_COP.1.1(2) The TSF shall perform [*cryptographic signature services*] in accordance with a **[**

1) RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater

that meets the following:

Case: RSA Digital Signature Algorithm

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

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

FCS_COP.1.1(3) The TSF shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm **SHA-1, SHA-224, SHA-256, SHA-384, SHA-512] and message digest sizes [160, 224, 256, 384, 512] bits** that meet the following: [*FIPS Pub 180-3, "Secure Hash Standard."*]

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

FCS_COP.1.1(4) The TSF shall perform [*keyed-hash message authentication*] in accordance with a specified cryptographic algorithm HMAC**-[SHA-1,SHA-256, SHA-512], key size [160, 256, 512], and message digest sizes [160, 256, 512] bits** that meet the following: [*FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-3, "Secure Hash Standard."*]

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

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

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

6.1.2.8 FCS_HTTPS_EXT.1 Extended: HTTPS

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

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

6.1.2.9 FCS_TLS_EXT.1 Extended: TLS

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

Mandatory Ciphersuites:
TLS_RSA_WITH_AES_128_CBC_SHA

Optional Ciphersuites:
[TLS_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256].

6.1.3 User Data Protection (FDP)

6.1.3.1 FDP_RIP.2 Full Residual Information Protection

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

6.1.4 Identification and Authentication (FIA)

6.1.4.1 FIA_PMG_EXT.1 Password Management

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

1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“`”, “~”,

“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, “_”, “-“, “+”, “=”, “[“, “{“, “]”, “}”, “\”, “|”, “;”, “:”, “‘”, "'", "/", “?”, “>”, ".", “,”, “<”]];

2. Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater;

6.1.4.2 FIA_UIA_EXT.1 User Identification and Authentication

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

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

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

6.1.4.3 FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism

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

6.1.4.4 FIA_UAU.7 Protected Authentication Feedback

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

6.1.5 Security Management (FMT)

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

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

6.1.5.2 FMT_SMF.1 Specification of Management Functions

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

- *Ability to administer the TOE locally and remotely;*
- *Ability to update the TOE, and to verify the updates using [published hash] capability prior to installing those updates;*
- *[No other capabilities.]*]

6.1.5.3 FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1 The TSF shall maintain the roles: [

- *Authorized Administrator*]

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

FMT_SMR.2.3 The TSF shall ensure that the conditions [

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

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

are satisfied.

6.1.6 Protection of the TSF (FPT)

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

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

6.1.6.2 FPT_APW_EXT.1 Extended: Protection of Administrator Passwords

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

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

6.1.6.3 FPT_STM.1 Reliable Time Stamps

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

6.1.6.4 FPT_TUD_EXT.1 Extended: Trusted Update

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

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

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

6.1.6.5 FPT_TST_EXT.1: TSF Testing

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

6.1.7 TOE Access (FTA)

6.1.7.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

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

- terminate the session]

after a Security Administrator-specified time period of inactivity.

6.1.7.2 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*].

6.1.7.3 FTA_SSL.4 User-initiated Termination

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

6.1.7.4 FTA_TAB.1 Default TOE Access Banners

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

6.1.8 Trusted Path/Channels (FTP)

6.1.8.1 FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 The TSF shall **use [TLS]** to provide a **trusted** communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [none]** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data **from disclosure and detection of modification of the channel data**.

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

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

6.1.8.2 FTP_TRP.1 Trusted Path

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

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

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

6.2 Security Assurance Requirements

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

Table 9 – Security Assurance Requirements

Assurance Classes	Assurance Component	Description
Security Target	ASE_INT.1	ST Introduction

	ASE_CCL.1	Conformance Claims
	ASE_OBJ.1	Security Objectives for the operational environment
	ASE_ECD.1	Extended Components Definition
	ASE_REQ.1	Stated security requirements
	ASE_TSS.1	TOE summary specification
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative Procedures
Lifecycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM Coverage
Tests	ATE_IND.1	Independent Testing – Conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability Analysis

7 TOE SUMMARY SPECIFICATION

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

7.1 Security Audit

7.1.1 FAU_GEN.1, FAU_GEN.2, FPT_STM.1

Auditing of events is performed using the standard Linux syslog utility. All log events are associated with a time-stamp provided by the appliance. The TOE stores the audit events locally in the appropriate log files, and can be optionally configured to export the log files to an external source over TLS. Local logs files are stored in folders under **/var/log**. The Apache httpd daemon is also configured to log events to syslog.

Table 10 - Audit Event Specifications

Requirements	Auditable Events	Location
FAU_GEN.1	Start-up and shutdown of the audit functions	/var/log/messages
FAU_GEN.1	All administrative actions	/var/log/apache2.log
FCS_TLS_EXT.1	Failure to establish a TLS Session Establishment/Termination of a TLS session	/var/log/apache2/error.log /var/log/apache2/access.log
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session Establishment/Termination of a HTTPS session	/var/log/apache2/error.log /var/log/apache2/access.log
FIA_UIA_EXT.1	All use of the identification and authentication mechanism	/var/log/apache2.log
FIA_UAU_EXT.2	All use of the authentication mechanism	/var/log/apache2.log
FPT_STM.1	Changes to the time	/var/log/messages
FPT_TUD_EXT.1	Initiation of update	/var/SevOne/upgrade-appliance.log
FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session	/var/log/apache2.log
FTA_SSL.3	The termination of a remote session by the session locking mechanism	/var/log/apache2.log
FTA_SSL.4	The termination of an interactive session	/var/log/apache2.log
FTP_TRP.1	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions	/var/log/apache2/error.log /var/log/apache2/access.log
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions	As for FCS_TLS_EXT.1 above

There are no additional protocol failure audit messages recorded.

7.1.2 FAU_STG_EXT.1

Local log files are rotated on a time based rotation every 24 hours. At fixed periods the TOE compresses (gz) the current logfile, having started a new logfile. In the event of audit storage being exhausted, the TOE will delete the oldest records and continue storing the new audit logs. The maximum size of an audit log is 1Gb and a maximum of 10 log files can exist before they are rotated. Logs can also be exported in real-time over TLS protected syslog.

Audit logs are protected through the Web Interface by not allowing any functionality on the UI to modify the logs. Thus only trusted, authorized administrators might have access to the TOE audit log files.

7.2 Cryptographic Support

The TOE uses the FIP-approved cryptographic functions provided by OpenSSL v2.0.4 (**certificate #1747**. The method of use of cryptographic functions is hard-coded in the implementation of the TOE and cannot be configured by administrators.

7.2.1 FCS_HTTPS_EXT.1, FTP_TRP.1

HTTPS (using TLS 1.2) is used to protect interaction between the TOE and the Authorized Administrator's web-browser when connecting to the web-based UI to manage the TOE. TLS version 1.2 is implemented in accordance with RFC 5246. RSA certificates (generated in accordance with NIST SP800-56B, see Section 7.2.3 below) are used in the TLS handshake to authenticate the connection prior to administrator authentication.

7.2.2 FCS_TLS_EXT.1, FTP_ITC.1

TLS v1.2 is implemented to protect communication between the TOE and the external audit server, using supporting the following ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256

TLS is used to protect the export of syslog events between the TOE and the external log system. This is performed using a specific configuration and build of syslog-NG.

7.2.3 FCS_CKM.1

The TOE uses OpenSSL to generate asymmetric cryptographic keys using a domain parameter generator and a random number generator that meet ANSI X9.31 with an equivalent key strength of at least 112 bits (rDSA keys).

Domain parameters used in RSA-based key establishment schemes meet NIST SP800-56B. While the TOE fulfills all of the NIST SP800-56B requirements without extensions, of relevance to this PP the TOE uses FIPS 186-2 for key establishment as documented by the CAVP validations listed in table 11.

7.2.4 FCS_CKM_EXT.4, FPT_SKP_EXT.1

The TOE uses the OpenSSL FIPS module to support the generation and zeroization of keys. The OpenSSL module provides the following functions in support of key handling:

> "Keys residing in internally allocated data structures can only be accessed using the Module defined API. The operating system protects memory and process space from unauthorized access. Zeroization of sensitive data is performed automatically by API function calls for intermediate data items, and on demand by the calling process using Module provided API function calls provided for that purpose.
>
> Only the process that creates or imports keys can use or export them. No persistent storage of key data is performed by the Module. All API functions are executed by the invoking process in a non-overlapping sequence such that no two API functions will execute concurrently.
>
> The calling process can perform key zeroization of keys by calling an API function."3 OpenSSL uses DRBG to overwrite the memory space with random bits.

The TOE handles zeroization for all CSP, plaintext secret and private cryptographic keys according to Table 11.

Table 11 – Zeroization

Data	Generation/Algorithm	Description	Zeroization method
RSA Public/private keys	ANSI X9.31/RSA	Plaintext self-signed certificates used for HTTPS/TLS, stored in RAM	RAM scrubbed using OpenSSL method
RSA Public/private keys	ANSI X9.31/RSA	Plaintext self-signed certificates used for HTTPS/TLS, stored on hard disk	Administrator performed low-level format of TOE device
User password	User generated	Plain text value held in RAM as entered by user.	RAM scrubbed using OpenSSL method following authentication request completion
RNG		Plaintext seed key and state of RNG held in RAM	Memory scrubbed by OpenSSL following seed passage to OpenSSL RNG. RNG scrubbed using OpenSSL method during normal shutdown.

The TOE does not provide any method of viewing stored keys within persistent memory through any of the TSFI's by design of the UI and the Linux security permissions present within the OS.
Copies of ephemeral keys generated during session negotiation as well as copies of the persistent keys present on the persistent storage of the TOE are loaded into memory by OpenSSL and scrubbed according to the security policy of the module.

Keys such as RSA public/private keypairs are capable of being zeroized through a low level format of the The zeroization process does require that the system administrator reformat all drives in the RAID array through other means as no ability to perform this function is present within the TOE. The TOE does support booting from the enclosed DVDROM drive, or the disks in the RAID array can be accessed and removed by removing physically secured SevOne faceplate. This allows the purchaser of the TOE to ensure that the information is scrubbed using a suitable method for their data security requirements.

7.2.5 FCS_COP.1(1/2/3/4)

All cryptographic functions in the TOE are FIPS approved, as detailed in the following table:

Table 12 – CAVP Certificates

Algorithm	Certificate Number
AES	#2394
DRBG	#316
DSA	#748
RSA	#1237
SHS	#2056
Triple-DES	#1492
ECDSA	#394
CVL	#71
HMAC	#1485

7.2.6 FCS_RBG_EXT.1

The TOE leverages OpenSSL to perform random number generation using AES which is seeded from a hardware entropy source within the Intel processor which provides at least 256 bits of entropy to seed the RNG. The TOE has OpenSSL configured to use the rdrand engine which calls the rdrand function native to the Intel processors pulling entropy from a hardware entropy source to produce the entropy required for the initial seeding of the RNG[3]. Intel makes no claim of the entropy of their NDRNG other than the rate at which it is reseeded in their public documentation, but during the design of the system the developer considered the entropy rate to be no less than 0.5. In the event that insufficient entropy is available the CPU CF (Carry Flag) will be set in order to signal to OpenSSL that insufficient entropy was available during the OpenSSL get_random_bytes call to the rdrand engine which was used to seed the PRNG. The hardware on-processor entropy source passes the randomly generated bits to the OpenSSL AES container (in CBC-MAC mode) to distill the entropy into non-deterministic random numbers, in compliance with NIST SP 800-90 and FIPS Pub 140-2. During instantiation of the PRNG the TOE calls the get_entropy function which pulls the TOE configured number of random bits into the PRNG to use as the entropy seed to ensure that 256 bit cryptographically strong random numbers can be generated by the TOE.

[3] This is detailed in http://software.intel.com/en-us/articles/intel-digital-random-number-generator-drng-software-implementation-guide

7.3 User Data Protection

7.3.1 FDP_RIP.2

The network interfaces of the SevOne NMS 5.5.0.1 appliances are implemented to ensure leakage of data between network packets through the reuse of memory resources is not possible. This is achieved using the standard Linux driver interfaces. The Linux Kernel creates two ring buffers called tx_ring and rx_ring which contain a dually linked list of socket buffers representing the packet's payload as well as any control information. These buffers represent both the transmit and receive memory buffers of the kernel. These socket buffers are of a specific length as defined by the kernel networking parameters to include both the packet data and any additional memory space required for headers or tail room. In the Linux kernel these packet buffers are of a specific flat size and allocated via a call to get_zeroized_page for the appropriate length.

When the NIC hardware has a packet (to transmit or receive) fully loaded into the hardware's memory space a DMA interrupt is generated and an amount of memory up to the amount of space allocated in the skb_buff is copied via mem_copy. Data from the tx_ring and rx_ring are then passed up/down to a higher layer (i.e. IP) in the networking stack for further processing of the contents of that data packet.

7.4 Identification and Authentication

7.4.1 FIA_UAU_EXT.1, FIA_UAU_EXT.2, FIA_UAU.7, FIA_PMG_EXT.1, FTA_TAB.1, FPT_APW_EXT.1

When initiating a connection to the web-based UI from their client browser, the administrator is presented with an advisory and consent message about use of the TOE on the login screen. The administrator is then required to authenticate, providing username and password.

The username/password will then be verified against the local password database (which is stored on the appliance using a salted hash from bcrypt). If the password is valid for the entered username, the administrator is authenticated to gain access to the web-based UI. If the username is not valid or the password is not successfully verified for the entered username a general login failure message is presented to the user.

Passwords must meet the following complexity options:

- The administrator will be able to configure a minimum of password length of 15 characters
- Include at least one special character (“`”, “~”, “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, “_”, “-“, “+”, “=”, “[“, “{“, “]”, “}”, “\”, “|”, “;”, “:”, “‘”, """, "/", “?”, “>”, ".", “,”, “<”)and at least two of the following three types of characters: Lower case letters, UPPERCASE LETTERS, and numbers. In addition, passwords cannot contain more than two of a given type of character in succession (upper and lower case letters count as the same type). (This is enforced by selecting the “Require strong passwords” option.)

In addition, the administrator can also select the options to further increase the on-going strength of password.

The same requirements are applied to the authentication credentials (local username/password) used to gain local access to the local appliance GUI via the (physically protected) local connection. A similar advisory banner is displayed to the user when establishing the local GUI connection, prior to entering the authentication credentials. During entry of authentication credentials at the local GUI, the

characters entered at the username prompt are reflected to the screen, but characters entered at the password prompt are masked with an asterisk ("*") character.

7.5 Security Management

7.5.1 FMT_SMR.1, FMT_MTD.1, FMT_SMF.1, FPT_TUD_EXT.1

There is only one user role defined for the TOE: Authorized Administrator, which is represented in the TOE by a user who is a member of the "Admins" user group, with all user permissions selected. The Authorized Administrator is responsible for creating and maintaining administrator accounts. These accounts are managed through the User Manager screen, which is accessed by clicking the **Administration** menu from the navigation bar, selecting **Access Configuration**, and then select **User Manager** (details of user management are provided in (Includes Manage Users Workflows) 5.5[AG]).

When connected to the web user Interface (locally or remotely) the Authorized Administrator is able to administer the TOE they are directly logged on to. Details of devices monitoring and reporting are provided in 5.5[UG].

Should the appliance need upgrading, the Administrator would Contact SevOne Support to arrange for delivery of a version of the update files that are Common Criteria compatible.

The TOE software comes pre-installed with Common Criteria safe Installer version "2016-07-22-1778e64" (file seveone-gui-installer-2016-07-22-1778e64.phar) shipped with the Common Criteria appliance. The Installer interacts with NMS to perform secure software/patch upgrade.

7.6 Protection of the TSF

7.6.1 FPT_TST_EXT.1

7.6.1.1 Hardware BIOS Power On Self Tests (POST) and Ongoing Hardware Health Checks

The TOE performs BIOS power-on self-tests (from Dell), then on-going self-monitoring during operation for hardware failures and process failures. Tests demonstrating the correct operation of the TOE hardware include tests for the entropy health of the Intel hardware based entropy source per NIST SP 800-90.

These tests also include a verification of the BIOS image against a known SHA-256 hash to ensure the authenticity of the binary and provide tamper-resistance of the TOE firmware and its health tests.

7.6.1.2 OpenSSL

OpenSSL FIPS provides the following power-up self-tests at module initialization[4] and continuous condition tests during operation to demonstrate the cryptographic operations of the TOE are operating correctly. If the power-up self-tests fail, subsequent calls to the module will fail and are logged disallowing further cryptographic operations.

- Power-up tests:

[4]The FIPS mode initialization is performed when the application invokes the FIPS_mode_set() call which returns a "1" for success and "0" for failure.

Table 13 – OpenSSL Power-up Self-tests

Algorithm	Test
AES	KAT
RSA	KAT
RNG	KAT
HMAC-SHA-1	KAT
SHA-1	KAT
SHA-256	KAT
SHA-512	KAT
module integrity	HMAC-SHA-1

- Conditional self-tests:

Table 14 – OpenSSL Conditional Self-tests

Algorithm	Test
RSA	pairwise consistency
PRNG	continuous test

7.7 TOE Access

7.7.1 FTA_SSL_EXT.1, FTA_SSL.3, FTA_SSL.4

After an Authorized Administrator configurable period of inactivity the TOE will automatically terminate the interactive web-based administrative session. This period of time can be configured to a value between 5 minutes and 86400 minutes. To resume administration of the TOE the administrator must re-authenticate to the UI.

Similarly, after an Authorized Administrator configurable period of inactivity the TOE will automatically requiring the user to re-authenticate to start a new session.

8 RATIONALE

This ST claims Exact Compliance to Network Devices Protection Profile v1.1 and the NDPP Errata #3. Hence, conformance claim rationale, security objectives rationale, extended SFR rationale, and security requirements rationale (including SAR choice rationale) are explicitly addressed by the Protection Profile and the Extended Package, without further elaboration in this ST, with the following exceptions.

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

8.1 Dependency Rationale

The following table provides the rationale of the satisfaction of SFR dependencies. A justification has been provided where dependencies do not appear to be satisfied directly.

Table 15 – SFR Dependency Rationale

SFR	Dependency	Satisfaction of dependency
FAU_GEN.1	FPT_STM.1 Reliable time stamps	FPT_STM.1
FAU_GEN.2	FAU_GEN.1 Audit data generation FIA_UID.1 Timing of identification	FAU_GEN.1 FIA_UID.1 dependency satisfied by FIA_UIA_EXT.1 which authenticates administrator identity prior to interaction with TSF.
FAU_STG_EXT.1	FAU_GEN.1 Audit data generation	FAU_GEN.1
FCS_CKM.1	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1 (1-4) FCS_CKM.4 dependency met by FCS_CKM_EXT.4
FCS_CKM_EXT.4	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	FCS_CKM.1
FCS_COP.1(1)	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1 FCS_CKM.4 dependency met by FCS_CKM_EXT.4

SFR	Dependency	Satisfaction of dependency
FCS_COP.1(2)	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1 FCS_CKM.4 dependency met by FCS_CKM_EXT.4
FCS_COP.1(3)	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1 FCS_CKM.4 dependency met by FCS_CKM_EXT.4
FCS_COP.1(4)	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1 FCS_CKM.4 dependency met by FCS_CKM_EXT.4
FCS_RBG_EXT.1	None	n/a
FCS_HTTPS_EXT.1	FCS_TLS_EXT.1 Extended: TLS	FCS_TLS_EXT.1
FCS_TLS_EXT.1	FCS_COP.1(1) Cryptographic operation (for data encryption/decryption) FCS_COP.1(2) Cryptographic operation (for cryptographic signature) FCS_COP.1(3) Cryptographic operation (for cryptographic hashing) FCS_COP.1(4) Cryptographic operation (for keyed-hash message authentication) FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation) FCS_CKM.1 Cryptographic Key Generation FCS_CKM_EXT.4 Cryptographic Key Zeroization	FCS_COP.1(1) FCS_COP.1(2) FCS_COP.1(3)) FCS_COP.1(4) FCS_RBG_EXT.1 FCS_CKM.1 FCS_CKM_EXT.4
FDP_RIP.2	None	n/a

SFR	Dependency	Satisfaction of dependency
FIA_PMG_EXT.1	None	n/a
FIA_UIA_EXT.1	FTA_TAB.1	FTA_TAB.1
FIA_UAU_EXT.2	None	n/a
FIA_UAU.7	FIA_UAU.1 Timing of authentication	FIA_UIA_EXT.1 which authenticates administrator identity prior to interaction with TSF.
FMT_MTD.1	FMT_SMF.1 Specification of Management Functions FMT_SMR.1 Security roles	Although FMT_SMR.1 is not included, FMT_SMR.2, which is hierarchical to FMT_SMR.1, is included. This satisfies the dependency. FMT_SMF.1
FMT_SMF.1	None	n/a
FMT_SMR.2	FIA_UID.1 Timing of identification	FIA_UIA_EXT.1 which authenticates administrator identity prior to interaction with TSF.
FPT_SKP_EXT.1	None	n/a
FPT_APW_EXT.1.1	None	n/a
FPT_STM.1	None	n/a
FPT_TUD_EXT.1	FCS_COP.1(3) (cryptographic hashing).	FCS_COP.1(3)
FPT_TST_EXT.1	None	n/a
FTA_SSL_EXT.1	FIA_UIA_EXT.1 Password-based Authentication and Identification Mechanism	FIA_UIA_EXT.1
FTA_SSL.3	None	n/a
FTA_SSL.4	None	n/a
FTA_TAB.1	None	n/a
FTP_ITC.1	None	n/a
FTP_TRP.1	None	n/a

9 ACRONYMS

Table 16 – Acronyms

Acronym	Definition
CBC	Cipher Block Chaining
CC	Common Criteria
CEM	Common Evaluation Methodology
CFB	Cipher Feedback
CSP	Critical Security Parameters
DNC	Data NetFlow Collector
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
FIPS	Federal Information Processing Standard
HSA	Hot Standby Appliance
NDPP	Network Device Protection Profile
NMS	Network Management System
OFB	Output Feedback
OSP	Organizational Security Policy
PAS	Performance Appliance Solution
PP	Protection Profile
RBG	Random Bit Generator
RNG	Random Number Generator
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
UI	User Interface