Infoblox Trinzic Appliances with NIOS v8.2.6 Security Target

Version 1.0
30 May 2018

Prepared for:

Infoblox

4750 Patrick Henry Drive
Santa Clara, CA 95054

Prepared By:

Leidos

Accredited Testing & Evaluation Labs

6841 Benjamin Franklin Drive
Columbia, MD 21046
USA

TABLE OF CONTENTS

LIST OF TABLES

1 Security Target Introduction

This section introduces the Target of Evaluation (TOE) and provides the Security Target (ST) and TOE identification, ST and TOE conformance claims, ST conventions, glossary and list of abbreviations.

The TOE is Infoblox Trinzic Appliances with NIOS v8.2.6 identified below in Section 1.1. The TOE appliances are a family of network appliances that provide core network services including DNS, DHCP, IPAM, FTP, TFTP, and HTTP.

The Security Target contains the following additional sections:

- TOE Description (Section 2)—provides an overview of the TOE and describes the physical and logical boundaries of the TOE
- Security Problem Definition (Section 3)—describes the threats and assumptions that define the security problem to be addressed by the TOE and its environment
- Security Objectives (Section 4)—describes the security objectives for the TOE and its operational environment necessary to counter the threats and satisfy the assumptions that define the security problem
- IT Security Requirements (Section 5)—specifies the security functional requirements (SFRs) and security assurance requirements (SARs) to be met by the TOE
- TOE Summary Specification (Section 6)—describes the security functions of the TOE and how they satisfy the SFRs
- Rationale (Section 7)—provides mappings and rationale for the security problem definition, security objectives, security requirements, and security functions to justify their completeness, consistency, and suitability.

1.1 Security Target, TOE and CC Identification

ST Title – Infoblox Trinzic Appliances with NIOS v8.2.6 Security Target

ST Version – Version 1.0

ST Date – 30 May 2018

TOE Identification – Infoblox Trinzic Appliances with NIOS v8.2.6

Hardware Appliance Model	Virtual Appliance Model[1]
IB-4015	IB-V4015
IB-2225	IB-V2225
IB-1425	IB-V1425
IB-825	IB-V825

Table 1-1: TOE Appliance Models

TOE Developer – Infoblox Inc.

Evaluation Sponsor – Infoblox Inc.

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, September 2012

[1] The product documentation sometimes refers to the Infoblox Trinzic Virtual Appliances with NIOS as vNIOS.

1.2 Conformance Claims

This ST and the TOE it describes are conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Components, Version 3.1, Revision 4, September 2012.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components, Version 3.1, Revision 4, September 2012.
  - Part 3 Conformant

This ST and the TOE it describes are conformant to the following package:

- EAL2 Augmented (ALC_FLR.2).

1.3 Conventions

The following conventions have been applied in this document:

Extended requirements – Security Functional Requirements not defined in Part 2 of the CC are annotated with a suffix of _EXT.

Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.

Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is identified with a number in parentheses following the base component identifier. For example, iterations of FCS_COP.1 are identified in a manner similar to FCS_COP.1(1) (for the component) and FCS_COP.1.1(1) (for the elements).

Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).

Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).

Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").

Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.4 Abbreviations and Acronyms

The following abbreviations and acronyms are used in this document. A brief definition is provided for abbreviations that are potentially unfamiliar, are specific to the TOE, or not obviously self-explanatory.

AES	Advanced Encryption Standard
API	Application Programming Interface
CBC	Cipher Block Chaining
CC	Common Criteria
EAL	Evaluation Assurance Level
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
NTP	Network Time Protocol
OS	Operating System
OSP	Organizational Security Policy
REST	Representational State Transfer

SSH	Secure Shell
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Function
vNIOS	Infoblox Trinzic Virtual Appliances with NIOS

2 TOE Description

The Target of Evaluation (TOE) is Infoblox Trinzic Appliances with NIOS v8.2.6, hereinafter referred to as Infoblox Trinzic Appliances, Infoblox Trinzic or the TOE. The TOE includes the NIOS v8.2.6 software, hardware and virtual appliances as identified in Table 1-1. The TOE appliances are a family of network appliances that provide core network services including DNS, DHCP, IPAM, FTP, TFTP, and HTTP.

The remainder of this section provides an overview of the TOE and a description of the TOE, including a description of the physical and logical scope of the TOE.

2.1 TOE Overview

The TOE is a network device that consolidates the delivery and management of core IP network services including DNS, DHCP, IPAM, FTP, TFTP, and HTTP. In addition to providing core network services expected from a network device, the TOE also provides Secure Grid functionality. Secure Grid is the capability of Infoblox appliances to work cooperatively in an enterprise deployment. One appliance is designated Master. The Master appliance distributes configuration information to all other Grid devices. Grid communication is secured with OpenVPN.

The TOE NIOS operating system is a hardened version of the Fedora Linux distribution optimized for security and network performance. The appliance models are differentiated by performance, capacity and availability to support various deployment scenarios such as a branch-office or large enterprise.

The TOE provides cryptography in support of Infoblox Trinzic security functionality. All algorithms implemented in support of TLS/HTTPS have been validated against CAVP requirements (http://csrc.nist.gov/groups/STM/cavp/). Infoblox supports both CC Mode and FIPS mode, and either is allowed in the evaluated configuration. CC and FIPS mode both meet the requirements for the CC evaluation. FIPS Mode also complies with FIPS certification standard to include required additional startup steps. In addition to placing the TOE into the FIPS mode, to comply with FIPS certification standards, the tamper evident label must be affixed properly as described in the Infoblox NIOS Administrator Guide.

The TOE provides the following major security features:

- **Secure Management:** Authorized administrators manage the TOE via a TLS protected web GUI, HTTPS/TLS protected API, SSH protected remote access to CLI, or via the local CLI console port. The TOE implements role based access control, password based authentication and auditing of security events. The management functions include (but are not limited to) audit configuration, user accounts, TOE session inactivity settings, and trusted TOE updates. The TOE's Perl API presents a representational state transfer (REST) API that assists in the integration of the Infoblox device into network environments. The API provides interfaces to DHCP, DNS, Grid and IPAM services.
- **Trusted Updates:** The TOE uses digital signatures to verify updates prior to installation.
- **Self Protection:** The TOE performs self-tests at startup to verify the integrity of hardware components and the cryptographic module.
- **Secure Grid:** The TOE uses an SSL/TLS VPN to protect communication between itself and other TOE instances when deployed in a grid.

2.2 TOE Architecture

The TOE is comprised of the following models: IB-4015, IB-2225, IB-1425, IB-825, IB-V4015, IB-V2225, IB-V1425, IB-V825; each with NIOS v8.2.6 software. Each TOE appliance instance is a hardened Linux system running NIOS v8.2.6. The TOE does not expose system interfaces (for example, there is no shell access).

The TOE presents a graphical user interface (GUI), a command line interface (CLI), and application programming interfaces (APIs). The TOE uses FIPS approved OpenSSL cryptographic algorithms version: 1.0.2j with FIPS Object Module v2.0.16.

The functionality is the same across all appliances and processor families. The appliances all run the same code and only differ by performance and capacity.

The following table identifies the hardware appliance models included in the TOE.

Infoblox Model	CPU	CPU Speed	Memory	Storage	Network Connectivity
IB-4015	Intel Xeon E5-2680	2.4 GHz	64GB	1.8TB	4x1Gbe Ethernet, nonaccelerated
IB-2225	Intel Xeon E5-2620	2.1 GHz	64GB	1.8TB	4x1Gbe Ethernet, nonaccelerated
IB-1425	Intel Xeon be E3-1275	3.6 GHz	32GB	900GB	4x1Gbe Ethernet, nonaccelerated
IB-825	Intel Xeon Core i3-6100TE	3.6 GHz	32GB	1TB	4x1Gbe Ethernet, nonaccelerated

Table 2-1: TOE Hardware Models

The virtual appliances in the TOE (IB-V4015, IB-V2225, IB-V1425, IB-V825) run in a virtual machine and will be qualified on a Hewlett Packard Enterprise DL380 G9 device (configured with 1 Xeon E5-2680v3, 128GB of memory, and 4x900gb SAS disk drives). The TOE includes virtual images for VMware. The virtual appliance and hardware appliance are identical from the operating system up. They differ in hardware device drivers only.

2.3 TOE Physical Boundaries

The TOE consists of the appliances and NIOS v8.2.6 software. See Table 1-1 for hardware and virtual appliance models in the TOE. See Table 2-1 for hardware appliance model specifications. The resource requirements for the virtual appliances are specified in Table 2-2.

The TOE can be deployed on a single machine (hereinafter referred to as "stand alone") or as a distributed environment of multiple machines (hereinafter referred to as a "grid"). In a distributed environment, the TOE provides Secure Grid functionality, protecting communication between the appliances using OpenVPN.

The TOE hardware appliances include the NIOS v8.2.6 software and the hardware listed in Table 2-1.

Depending on the administrator defined configuration, the TOE may require the following services to be present in the environment:

- an external log server when the TOE is configured to use an external syslog server,
- Active Directory, LDAP, RADIUS, TACAS+ servers when the TOE is configured to use an external authentication source,
- NTP server when the TOE is configured to use an NTP server,
- SSHv2 client when accessing the CLI remotely across an Ethernet network,
- The GUI can be accessed using the following browsers[2]: Firefox, Internet Explorer, or Chrome.
    - Firefox on Windows, Linux and Mac OS
    - Safari on Mac OS
    - Internet Explorer on Windows
    - Chrome on Windows, Linux and Mac OS.

The TOE virtual appliances include: IB-V4015, IB-V2225, IB-V1425, IB-V825 with NIOS v8.2.6 software. The Infoblox NIOS on VMware software runs on ESXi servers that have DAS (Direct Attached Storage), or iSCSI

[2] For specific supported browsers and versions according to host operating system, please see the NIOS Administrator Guide.

(Internet Small Computer System Interface) or FC (Fibre Channel) SAN (Storage Area Network) attached. The TOE software package for virtual appliances is installed on a host with Vmware ESXi 6.5 or 5.5 and then configured as a virtual appliance. The host appliance and Vmware are part of the operational environment and not part of the TOE. The following table lists the required memory, CPU, and disk allocation for each supported Infoblox virtual appliance model:

NIOS Virtual Appliance	Primary Disk (GB)	# of CPU Cores	Memory Allocation (GB)
IB-V4015	250	14	128
IB-V2225	250	8	64
IB-V1425	250	4	32
IB-V825	250	2	16

Table 2-2: Resource Requirements for Virtual Appliances

2.4 TOE Logical Boundaries

This section summarizes the security functions provided by the TOE:

- Security Audit
- Cryptographic Support
- Identification & Authentication
- Security Management
- Protection of the TSF
- TOE Access
- Trusted Path/Channels

2.4.1 Security Audit

The TOE generates audit records for security relevant events and include date and time of the event, subject identity, outcome for security events, and additional content for particular event types. For audit events resulting from actions of identified users, the TOE associates each auditable event with the identity of the user that caused the event.

The TOE protects the stored audit records in the audit trail from unauthorised deletion and prevents unauthorised modifications to the stored audit records in the audit trail. The TOE overwrites the oldest stored audit records when the audit trail is full.

2.4.2 Cryptographic Support

The TOE includes cryptographic functionality that provides random bit-generation, encryption/decryption, digital signature, secure hashing and key-hashing features. These features support cryptographic protocols including SSH, TLS and HTTPS.

SSH and Transport Layer Security protocol (HTTP over TLS) are used to provide protection of the communications surrounding the remote administrative sessions from disclosure and from undetected modification. Communication between the TOE and trusted external entities (syslog and authentication servers) is over TLS. Finally, the TOE uses a TLS protected channel to distribute configuration data when it is transmitted between distributed parts of the TOE.

The TOE supports TLS v1.0, v1.1, and v1.2. The TOE uses OpenSSL and OpenSSH cryptography and has obtained CAVP certificates for all supporting cryptographic algorithms.

2.4.3 Identification & Authentication

The TOE requires all users to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that user. The TOE supports user authentication using a local password mechanism and can be configured to use Active Directory (AD), LDAP, RADIUS or TACACS+ authentication. The TOE provides a mechanism to verify that passwords meet a defined quality metric and provides only obscured feedback to the user while the authentication is in progress.

2.4.4 Security Management

The security functions of the TOE are managed by an authorized administrator using a web-based GUI, SSH protected remote access to CLI, local CLI console port, or using an API. The ST defines the security role of 'superuser'. The superuser is the authorized administrator of the TOE and performs all security functions of the TOE including (but not limited to) managing audit configuration, password and authentication policies, and TOE updates.

2.4.5 Protection of the TSF

When Grid is enabled, communications between the TOE instances utilize TLS VPN to protect against the disclosure and modification of data exchanged between the TOE appliances.

The TOE provides reliable time stamps; and executes self-tests, during initial startup, to determine whether the TOE is operating correctly.

The TOE provides authorized administrators the ability to query the current version of; initiate updates to TOE firmware/software; and provides a digital signature mechanism to verify firmware/software updates to the TOE prior to installing those updates.

2.4.6 TOE Access

The TOE terminates local and remote interactive sessions after an administrator configurable time interval and allows user-initiated termination of the user's own interactive session.

Before establishing a user/administrator session, the TOE displays an administrator configured advisory banner warning message regarding unauthorized use of the TOE.

2.4.1 Trusted Path/Channel

The TOE communicates with authorized remote administrators via a web based GUI that is protected using HTTPS/TLS.

The TOE uses TLS to protect all communications with active directory and LDAP external authentication servers and syslog servers.

2.5 TOE Documentation

Infoblox provides the following administration and configuration guides for the TOE:

- *Infoblox NIOS Administrator Guide Release 8.2,* 400-0700-200 Rev. A, April 23, 2018
- *Infoblox Installation Guide 4005 Series Appliances* (*4015*), 400-0624-000 Rev. B, March 8, 2018
- *Infoblox Installation Guide 1405 Series Appliances,* 400-0671-000 Rev. E, March 8, 2018
- *Infoblox Installation Guide 2205 Series Appliances,* 400-0672-000 Rev. E, March 8, 2018
- *Infoblox Installation Guide 805 Series Appliances, 400-0670-000 Rev. E, March 8, 2018*
- *Infoblox Installation Guide NIOS™ for VMware, 400-0501-002 Rev. E, March 10, 2018*
- *Infoblox CLI Guide Release 8.2, 400-0701-200 Rev. A, July 21, 2017*
- *Infoblox API Documentation Release 8.2, 400-0702-200 Rev. B, May 24, 2018*

Note: Infoblox uses the ‘TE’ (order code prefix) and ‘IB’ (model number prefix) to the appliance models interchangeably. For example, the software itself uses the IB pre-fix, but the price list and administrative docs use the TE pre-fix. .

3 Security Problem Definition

This section defines the security problem to be addressed by the TOE, in terms of threats to be countered by the TOE or its operational environment, assumptions about the intended operational environment of the TOE, and Organizational Security Policies (OSPs) that apply to the TOE.

3.1 Assumptions

This section contains assumptions regarding the operational environment and the intended usage of the TOE.

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

A.NO_THRU_TRAFFIC_PROTECTION A standard/generic network device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the network device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the network device, destined for another network entity, is not covered by the TOE.

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.

A.TRUSTED_ADMINISTRATOR The authorized 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.

3.2 Threats

T.PASSWORD_CRACKING Malicious users or external IT entities may be able to take advantage of weak administrative passwords to gain privileged access to the device.

T.SECURITY_FUNCTIONALITY_FAILURE A component of the network device may fail during start-up or during operations causing a compromise or failure in the security functionality of the network device, leaving the device susceptible to malicious users or external IT entities.

T. UNAUTHORIZED_ADMINISTRATOR_ACCESS A user 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.

T.UNDETECTED_ACTIVITY	Users or external IT entities 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.UNTRUSTED_COMMUNICATION_CHANNELS	Malicious remote users or external IT entities 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.
T.UPDATE_COMPROMISE	Users 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.WEAK_AUTHENTICATION_ENDPOINTS	Malicious remote users or external IT entities may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g., shared password that is guessable or transported as plaintext.
T. WEAK_CRYPTOGRAPHY	Malicious remote users or external IT entities may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space.

3.3 Organizational Security Policies

This section describes the Organizational Security Policies (OSPs) that apply to the TOE. OSPs are used to provide a basis for security objectives that are commonly desired by TOE Owners in this operational environment, but for which it is not practical to universally define the assets being protected or the threats to those assets.

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

4 Security Objectives

This section identifies the security objectives for the TOE and its operational environment. The security objectives identify the responsibilities of the TOE and its environment in addressing the security problem defined in Section 3.

4.1 Security Objectives for the TOE

The following are the TOE security objectives:

O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security relevant events associated with users; and store those audit data locally, or externally if configured.
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE.
O. PROTECTED_COMMUNICATIONS	The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.
O.STRONG_CRYPTOGRAPHY	The TOE will provide strong standards-based cryptographic algorithms and key sizes.
O.TOE_ADMINISTRATION	The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, mechanisms that control a user's logical access to the TOE and mechanisms to ensure strong passwords.
O.TSF_SELF_TEST	The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.
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 cryptographically validated to be from a trusted source.

4.2 Security Objectives for the Environment

The following are the security objectives for the operational environment of the TOE:

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.TRUSTED_ADMIN	TOE administrators are trusted to follow and apply all administrator guidance in a trusted manner.
OE.UPDATES	The TOE firmware and software is updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

5 IT Security Requirements

The security requirements for the TOE have been drawn from Parts 2 and 3 of the Common Criteria. The security functional requirements have been selected to correspond to the actual security functions implemented by the TOE while the assurance requirements have been selected to offer a low to moderate degree of assurance that those security functions are properly realized.

5.1 Extended Component Definition

This Security Target includes Security Functional Requirements (SFRs) that are not drawn from CC Part 2. These Extended SFRs are identified by having a label '_EXT' after the requirement name for TOE SFRs. The structure of the extended SFRs is modeled after the SFRs included in CC Part 2. The structure is as follows:

A. Class – The extended SFRs included in this ST are part of the identified classes of requirements.

B. Family – The extended SFRs included in this ST are part of several SFR families including the new families defined below.

C. Component – The extended SFRs are not hierarchical to any other components, though they may have identifiers terminating on other than "1". The dependencies for the extended components are identified both in this section and in the TOE SFR Dependencies section of this ST (Section 7.3, Requirement Dependency Rationale).

5.1.1 Extended Family Definitions

FPT_TUD_EXT

Family Behavior

This family requires that the TOE provide the ability to query the TOE version, update the TOE, and to verify the updates using a cryptographic mechanism prior to installing those updates.

Management: FPT_TUD_EXT.1

Manage TOE updates.

Audit: FPT_TUD_EXT.1

The following actions should be auditable if FAU_GEN.1 Security audit data generation is included:

Basic Level:

- Initiation of the update; and
- Result of the update attempt (success or failure).

Trusted Update (FPT_TUD_EXT.1)

Hierarchical to: No other components.

Dependencies: FCS_COP.1(2)

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

FPT_TUD_EXT.1.2 The TSF shall provide 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**.**

FPT_TST_EXT

Family Behavior

This family requires that the TOE run a suite of self-tests under certain conditions to demonstrate the correct operation of the TSF.

Management: FPT_TST_EXT.1

There are no management activities foreseen.

Audit: FPT_TST_EXT.1

There are no auditable events foreseen.

TSF Testing (FPT_TST_EXT.1)

Hierarchical to: No other components.

Dependencies: None

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

5.1.2 Extended Requirements Rationale

The following SFRs are modeled from requirements defined by an old (now sunset) protection profile for Network Devices. Earlier versions of this TOE satisfied these requirements prior to the PP being sunset and the SFRs are being retained in this ST to indicate a continuity of product functionality.

- FPT_TUD_EXT.1:

  FPT_TUD_EXT.1 has been created because the TOE is required to support a special method of trusted update. The existing SFRs in FPT class of CC Part 2 cannot meet the requirements.

  This extended requirement is intended to require the TOE provide update functions to include cryptographic verification methods to ensure the updates can be trusted; the ability for administrators to initiate updates; and to query the TOE version.

- FPT_TST_EXT.1:

  FPT_TST_EXT.1 has been created because the TOE is required to support special TSF Testing methods. The existing SFRs in FPT class of CC Part 2 cannot meet the requirements.

  This extended requirement is intended to require the TOE to run a suite of self-tests under certain conditions to demonstrate the correct operation of the TSF.

5.2 TOE Security Functional Requirements

This section specifies the security functional requirements (SFRs) for the TOE.

Requirement Class	Requirement Component
FAU: Security Audit	FAU_GEN.1: Audit data generation
	FAU_GEN.2: User identity association
	FAU_STG.1: Protected audit trail storage
	FAU_STG.4: Prevention of audit data loss
FCS: Cryptographic support	FCS_CKM.1: Cryptographic key generation

Requirement Class	Requirement Component
	FCS_CKM.4: Cryptographic key destruction
	FCS_COP.1: Cryptographic operation
FIA: Identification and authentication	FIA_SOS.1: Verification of secrets
	FIA_UID.2: User identification before any action
	FIA_UAU.2: Timing of authentication
	FIA_UAU.5: Multiple authentication mechanisms
	FIA_UAU.7: Protected authentication feedback
FMT: Security management	FMT_MOF.1: Management of security functions behaviour
	FMT_MTD.1: Management of TSF data
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.1: Security roles
FPT: Protection of the TSF	FPT_ITT.1: Basic internal TSF data transfer protection
	FPT_STM.1: Reliable time stamps
	FPT_TST_EXT.1: TSF Testing
	FPT_TUD_EXT.1: Trusted Update
TOE Access	FTA_SSL.3: TSF-initiated termination
	FTA_SSL.4: User-initiated termination
	FTA_TAB.1: Default TOE access banners
FTP: Trusted path/channels	FTP_ITC.1: Inter-TSF trusted channel
	FTP_TRP.1: Trusted path

Table 5-1: TOE Security Functional Components

5.2.1 Security Audit (FAU)

FAU_GEN.1 – Audit data generation

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

a) Start-up and shutdown of the audit functions;
b) All auditable events for the [***not specified***] level of audit; and [
c) **all administrative actions;**
d) **the specifically defined auditable events listed in Table 5-1: Auditable Events].**

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 (if applicable), and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [**information specified in column three of Table 5-1: Auditable Events**].

Table 5-1: Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None	None

FAU_GEN.2	None	None
FAU_STG.1	None	None
FAU_STG.4	None	None
FCS_CKM.1	None	None
FCS_CKM.4	None	None
FCS_COP.1(1)	None	None
FCS_COP.1(2)	None	None
FCS_COP.1(3)	None	None
FCS_COP.1(4)	None	None
FCS_COP.1(5)	None	None
FIA_SOS.1	None	None
FIA_UAU.2	All use of the authentication mechanism.	Origin of the attempt (e.g. IP address).
FIA_UID.2	All use of the identification mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU.5	All use of the external identification/authentication mechanism	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU.7	None	None
FMT_MOF.1	None	None
FMT_MTD.1	None	None
FMT_SMF.1	Use of the management functions.	Identity of the administrator performing these functions.
FMT_SMR.1	None	None
FPT_ITT.1	None	None
FPT_STM.1	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)	None
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None
FTA_SSL.4	The termination of an interactive session.	None
FTA_TAB.1	None	None
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	Identification of the claimed user identity.

FAU_GEN.2 – User identity association	
FAU_GEN.2.1	For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

FAU_STG.1 – Protected audit trail storage	
FAU_STG.1.1	The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.
FAU_STG.1.2	The TSF shall be able to **[*prevent*]** unauthorised modifications to the stored audit records in the audit trail.

FAU_STG.4 – Prevention of audit data loss	
FAU_STG.4.1	The TSF shall **[*overwrite the oldest stored audit records*]** and [**no other action**] if the audit trail is full.

5.2.2 Cryptographic Support (FCS)

FCS_CKM.1 – Cryptographic key generation	
FCS_CKM.1.1	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**RSA-based schemes, elliptic curve-based schemes, finite field-based schemes]** and specified cryptographic key sizes [**2048-bit or greater**] that meet the following [**FIPS PUB 186-4**].

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 [**zeroization**] that meets the following: [**FIPS 140-2**].

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 and 256 bits**] that meets the following: [• **AES as specified in ISO 18033-3,** • **CBC as specified in ISO 10116**].

FCS_COP.1(2) – Cryptographic operation (for cryptographic signature services)	
FCS_COP.1.1(2)	The TSF shall perform [**cryptographic signature services]** in accordance with a specified cryptographic algorithm **[RSA Digital Signature Algorithm (rDSA**)] and cryptographic key sizes **[2048 bits or greater]** that meet the following: **[FIPS PUB 186-4, 'Digital Signature Standard'].**

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-256, SHA-384 and SHA-512**] and ~~cryptographic key~~ **message digest** sizes [**160, 256, 384, 512 bits**] that meet the following: [**ISO/IEC 10118-3:2004**].

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

FCS_COP.1(5) – Cryptographic operation (for Random Bit Generation)

FCS_COP.1.1(5) The TSF shall perform [**Random Bit Generation**] in accordance with a specified cryptographic algorithm [**HMAC_DRBG(any)**] and ~~cryptographic key sizes~~ **minimum entropy seed of** [**256-bits**] that meet the following: [**ISO/IEC 18031:2011**].

5.2.3 Identification and Authentication (FIA)

FIA_SOS.1 – Verification of secrets

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet **[**

- **Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [selection: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")];**
- **Minimum password length shall settable by the authorized administrator, and support passwords of 15 characters or greater;**
- **Passwords shall have a maximum lifetime;**
- **New passwords must contain a minimum of 4 character changes from the previous password]**.

FIA_UID.2 – User identification before any action

FIA_UID.2.1 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

FIA_UAU.2 – User authentication before any action

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

FIA_UAU.5 – Multiple authentication mechanisms

FIA_UAU.5.1 The TSF shall provide [**Local Password-based Authentication, and support for remote authentication via external AD, LDAP, RADIUS and TACACS+ services**] to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the [

- **The user is authenticated using the mechanism configured for that user.**

].

FIA_UAU.7 – Protected authentication feedback

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

5.2.4 Security Management (FMT)

FMT_MOF.1 – Management of security functions behaviour

FMT_MOF.1.1 The TSF shall restrict the ability to **[*modify the behaviour of*]** the functions **[the list of management functions identified in FMT_SMF.1]** to **[superuser]**.

FMT_MTD.1 – Management of TSF Data

FMT_MTD.1.1 The TSF shall restrict the ability to [***[manage]***] the [**TSF data**] to [**authorized administrators**].

FMT_SMF.1 – Specification of Management Functions

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

- **Manage authentication policy**
- **Manage password policy**
- **Manage user creation/modification**
- **Manage the TOE banner**
- **Manage TOE updates**
- **Manage TOE session Inactivity**
- **Manage audit configuration**
- **Manage TOE system time**
- **Manage passwords**

].

FMT_SMR.1 – Security roles	
FMT_SMR.1.1	The TSF shall maintain the roles: [**superuser**].
FMT_SMR.1.2	The TSF shall be able to associate users with roles.

5.2.5 Protection of the TSF (FPT)

FPT_ITT.1 – Basic internal TSF data transfer protection	
FPT_ITT.1.1	The TSF shall protect TSF data from **[*disclosure and modification*]** when it is transmitted between separate parts of the TOE.

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

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: [• **memory test,** • **cryptographic library test,** • **cryptographic known answer test,** • **Random number generation test**].

FPT_TUD_EXT.1 – Trusted Update	
FPT_TUD_EXT.1.1	The TSF shall provide administrators the ability to query the current version of the TOE firmware/software.
FPT_TUD_EXT.1.2	The TSF shall provide 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 **[*digital signature mechanism*]** prior to installing those updates**.**

5.2.6 TOE Access (FTA)

FTA_SSL.3 – TSF-initiated termination	
FTA_SSL.3.1	The TSF shall terminate an interactive session after a **[an administrator configurable time interval of 60 and 31536000 seconds (one minute – one year)]**.

FTA_SSL.4 – User-initiated termination	
FTA_SSL.4.1	The TSF shall allow user-initiated termination of the user's own interactive session.

FTA_TAB.1 – Default TOE access banners	
FTA_TAB.1.1	Before establishing a user session, the TSF shall display an advisory warning message regarding unauthorised use of the TOE.

5.2.7 Trusted Path/channels (FTP)

FTP_ITC.1 Inter-TSF trusted channel	
FTP_ITC.1.1	The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from **detection of** modification or disclosure.
FTP_ITC.1.2	The TSF shall permit [***the TSF***] to initiate communication via the trusted channel.
FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for **[Active Directory and LDAP Authentication, Sending Audit Events to Syslog]**.

FTP_TRP.1 – Trusted Path	
FTP_TRP.1.1	The TSF shall provide a communication path between itself and [***remote***] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [***disclosure, [undetected modification]***].
FTP_TRP.1.2	The TSF shall permit [***remote users***] to initiate communication via the trusted path.
FTP_TRP.1.3	The TSF shall require the use of the trusted path for [***initial user authentication, [all remote administrative interactions with the TSF]***].

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 2 augmented with ALC_FLR.2 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Requirement Class	Requirement Component
ADV: Development	ADV_ARC.1: Security architecture description
	ADV_FSP.2: Security-enforcing functional specification
	ADV_TDS.1: Basic design
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.2: Use of a CM system
	ALC_CMS.2 Parts of the TOE CM coverage
	ALC_DEL.1: Delivery procedures
	ALC_FLR.2: Flaw remediation
ASE: Security Target evaluation	ASE_CCL.1: Conformance claims
	ASE_ECD.1: Extended components definition
	ASE_INT.1: ST introduction
	ASE_OBJ.2: Security objectives
	ASE_REQ.2: Derived security requirements

Requirement Class	Requirement Component
	ASE_SPD.1: Security problem definition
	ASE_TSS.1: TOE summary specification
ATE: Tests	ATE_COV.1: Evidence of coverage
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.2 Vulnerability analysis

Table 5-2: EAL2 Augmented with ALC_FLR.2 Assurance Components

5.3.1 Development (ADV)

ADV_ARC.1 – Security architecture description

ADV_ARC.1.1D The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2D The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3D The developer shall provide a security architecture description of the TSF.

ADV_ARC.1.1C The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2C The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3C The security architecture description shall describe how the TSF initialization process is secure.

ADV_ARC.1.4C The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5C The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

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

ADV_FSP.2 – Security-enforcing functional specification

ADV_FSP.2.1D The developer shall provide a functional specification.

ADV_FSP.2.2D The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.2.1C The functional specification shall completely represent the TSF.

ADV_FSP.2.2C The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.2.3C The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.2.4C For each SFR-enforcing TSFI, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.

ADV_FSP.2.5C For each SFR-enforcing TSFI, the functional specification shall describe direct error messages resulting from processing associated with the SFR-enforcing actions.

ADV_FSP.2.6C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

ADV_TDS.1 – Basic design	
ADV_TDS.1.1D	The developer shall provide the design of the TOE.
ADV_TDS.1.2D	The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.
ADV_TDS.1.1C	The design shall describe the structure of the TOE in terms of subsystems.
ADV_TDS.1.2C	The design shall identify all subsystems of the TSF.
ADV_TDS.1.3C	The design shall describe the behaviour of each SFR-supporting or SFR-non-interfering TSF subsystem in sufficient detail to determine that it is not SFR-enforcing.
ADV_TDS.1.4C	The design shall summarise the SFR-enforcing behaviour of the SFR-enforcing subsystems.
ADV_TDS.1.5C	The design shall provide a description of the interactions among SFR-enforcing subsystems of the TSF, and between the SFR-enforcing subsystems of the TSF and other subsystems of the TSF.
ADV_TDS.1.6C	The mapping shall demonstrate that all TSFIs trace to the behaviour described in the TOE design that they invoke.
ADV_TDS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_TDS.1.2E	The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

5.3.2 Guidance Documents (AGD)

AGD_OPE.1 – Operational user guidance	
AGD_OPE.1.1D	The developer shall provide operational user guidance.
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 fulfil 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.
AGD_OPE.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1 – Preparative procedures	
AGD_PRE.1.1D	The developer shall provide the TOE including its preparative procedures.
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.
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.

5.3.3 Life-cycle Support (ALC)

ALC_CMC.2 – Use of a CM system	
ALC_CMC.2.1D	The developer shall provide the TOE and a reference for the TOE.
ALC_CMC.2.2D	The developer shall provide the CM documentation.
ALC_CMC.2.3D	The developer shall use a CM system.
ALC_CMC.2.1C	The TOE shall be labelled with its unique reference.
ALC_CMC.2.2C	The CM documentation shall describe the method used to uniquely identify the configuration items.
ALC_CMC.2.3C	The CM system shall uniquely identify all configuration items.
ALC_CMC.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ALC_CMS.2 – Parts of the TOE CM coverage	
ALC_CMS.2.1D	The developer shall provide a configuration list for the TOE.
ALC_CMS.2.1C	The configuration list shall include the following: The TOE itself; the evaluation evidence required by the SARs; and the parts that comprise the TOE.
ALC_CMS.2.2C	The configuration list shall uniquely identify the configuration items.
ALC_CMS.2.3C	For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.
ALC_CMS.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ALC_DEL.1 – Delivery procedures	
ALC_DEL.1.1D	The developer shall document and provide procedures for delivery of the TOE or parts of it to the consumer.
ALC_DEL.1.2D	The developer shall use the delivery procedures.
ALC_DEL.1.1C	The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.
ALC_DEL.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ALC_ FLR.2 – Flaw reporting procedures	
ALC_FLR.2.1D	The developer shall document and provide flaw remediation procedures addressed to TOE developers.
ALC_FLR.2.2D	The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.
ALC_FLR.2.3D	The developer shall provide flaw remediation guidance addressed to TOE users.
ALC_FLR.2.1C	The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.
ALC_FLR.2.2C	The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.2.3C	The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.
ALC_FLR.2.4C	The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.
ALC_FLR.2.5C	The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.
ALC_FLR.2.6C	The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.
ALC_FLR.2.7C	The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.
ALC_FLR.2.8C	The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

5.3.4 Security Target Evaluation (ASE)

ASE_CCL.1 – Conformance claims

ASE_CCL.1.1D	The developer shall provide a conformance claim.
ASE_CCL.1.2D	The developer shall provide a conformance claim rationale.
ASE_CCL.1.1C	The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.
ASE_CCL.1.2C	The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.
ASE_CCL.1.3C	The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.
ASE_CCL.1.4C	The CC conformance claim shall be consistent with the extended components definition.
ASE_CCL.1.5C	The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.
ASE_CCL.1.6C	The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.
ASE_CCL.1.7C	The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.
ASE_CCL.1.8C	The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.
ASE_CCL.1.9C	The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.
ASE_CCL.1.10C	The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.
ASE_CCL.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_ECD.1 – Extended components definition

ASE_ECD.1.1D	The developer shall provide a statement of security requirements.
ASE_ECD.1.2D	The developer shall provide an extended components definition.
ASE_ECD.1.1C	The statement of security requirements shall identify all extended security requirements.
ASE_ECD.1.2C	The extended components definition shall define an extended component for each extended security requirement.

ASE_ECD.1.3C	The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.
ASE_ECD.1.4C	The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.
ASE_ECD.1.5C	The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.
ASE_ECD.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ASE_ECD.1.2E	The evaluator shall confirm that no extended component can be clearly expressed using existing components.

ASE_INT.1 – ST introduction

ASE_INT.1.1D	The developer shall provide an ST introduction.
ASE_INT.1.1C	The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.
ASE_INT.1.2C	The ST reference shall uniquely identify the ST.
ASE_INT.1.3C	The TOE reference shall identify the TOE.
ASE_INT.1.4C	The TOE overview shall summarise the usage and major security features of the TOE.
ASE_INT.1.5C	The TOE overview shall identify the TOE type.
ASE_INT.1.6C	The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.
ASE_INT.1.7C	The TOE description shall describe the physical scope of the TOE.
ASE_INT.1.8C	The TOE description shall describe the logical scope of the TOE.
ASE_INT.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ASE_INT.1.2E	The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

ASE_OBJ.2 – Security objectives

ASE_OBJ.2.1D	The developer shall provide a statement of security objectives.
ASE_OBJ.2.2D	The developer shall provide a security objectives rationale.
ASE_OBJ.2.1C	The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.
ASE_OBJ.2.2C	The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.
ASE_OBJ.2.3C	The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.
ASE_OBJ.2.4C	The security objectives rationale shall demonstrate that the security objectives counter all threats.
ASE_OBJ.2.5C	The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.
ASE_OBJ.2.6C	The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.
ASE_OBJ.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_REQ.2 – Derived security requirements

ASE_REQ.2.1D	The developer shall provide a statement of security requirements.

ASE_REQ.2.2D	The developer shall provide a security requirements rationale.
ASE_REQ.2.1C	The statement of security requirements shall describe the SFRs and the SARs.
ASE_REQ.2.2C	All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.
ASE_REQ.2.3C	The statement of security requirements shall identify all operations on the security requirements.
ASE_REQ.2.4C	All operations shall be performed correctly.
ASE_REQ.2.5C	Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.
ASE_REQ.2.6C	The security requirements rationale shall trace each SFR back to the security objectives for the TOE.
ASE_REQ.2.7C	The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.
ASE_REQ.2.8C	The security requirements rationale shall explain why the SARs were chosen.
ASE_REQ.2.9C	The statement of security requirements shall be internally consistent.
ASE_REQ.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_SPD.1 – Security problem definition	
ASE_SPD.1.1D	The developer shall provide a security problem definition.
ASE_SPD.1.1C	The security problem definition shall describe the threats.
ASE_SPD.1.2C	All threats shall be described in terms of a threat agent, an asset, and an adverse action.
ASE_SPD.1.3C	The security problem definition shall describe the OSPs.
ASE_SPD.1.4C	The security problem definition shall describe the assumptions about the operational environment of the TOE.
ASE_SPD.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_TSS.1 – TOE summary specification	
ASE_TSS.1.1D	The developer shall provide a TOE summary specification.
ASE_TSS.1.1C	The TOE summary specification shall describe how the TOE meets each SFR.
ASE_TSS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ASE_TSS.1.2E	The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

5.3.5 Tests (ATE)

ATE_COV.1 – Evidence of coverage	
ATE_COV.1.1D	The developer shall provide evidence of the test coverage.
ATE_COV.1.1C	The evidence of the test coverage shall show the correspondence between the tests in the test documentation and the TSFIs in the functional specification.
ATE_COV.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_FUN.1 – Functional testing	
ATE_FUN.1.1D	The developer shall test the TSF and document the results.
ATE_FUN.1.2D	The developer shall provide test documentation.
ATE_FUN.1.1C	The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2C	The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.
ATE_FUN.1.3C	The expected test results shall show the anticipated outputs from a successful execution of the tests.
ATE_FUN.1.4C	The actual test results shall be consistent with the expected test results.
ATE_FUN.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2 – Independent testing - sample

ATE_IND.2.1D	The developer shall provide the TOE for testing.
ATE_IND.2.1C	The TOE shall be suitable for testing.
ATE_IND.2.2C	The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.
ATE_IND.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ATE_IND.2.2E	The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.
ATE_IND.2.3E	The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.3.6 Vulnerability Assessment (AVA)

AVA_VAN.2 – Vulnerability analysis

AVA_VAN.2.1D	The developer shall provide the TOE for testing.
AVA_VAN.2.1C	The TOE shall be suitable for testing.
AVA_VAN.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_VAN.2.2E	The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.
AVA_VAN.2.3E	The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.
AVA_VAN.2.4E	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.

6 TOE Summary Specification

This chapter describes the following security functions:

- Security audit
- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Protection of the TSF
- TOE Access

6.1 Security Audit

The TOE generates audit records for the following auditable events:

- Start-up and shutdown of the TOE
- All administrative actions
- All auditable events as specified in the following table.

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FAU_STG.1	None	None
FAU_STG.4	None	None
FCS_CKM.1	None	None
FCS_CKM.4	None	None
FCS_COP.1(1)	None	None
FCS_COP.1(2)	None	None
FCS_COP.1(3)	None	None
FCS_COP.1(4)	None	None
FCS_COP.1(5)	None	None
FIA_SOS.1	None	None
FIA_UAU.2	All use of the authentication mechanism.	Origin of the attempt (e.g. IP address).
FIA_UID.2	All use of the identification mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU.5	All use of the external identification/authentication mechanism	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU.7	None	None
FMT_MOF.1	None	None
FMT_MTD.1	None	None
FMT_SMF.1	Use of the management functions.	Identity of the administrator performing these functions.

FMT_SMR.1	None	None
FPT_ITT.1	None	None
FPT_STM.1	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)	None
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None
FTA_SSL.4	The termination of an interactive session.	None
FTA_TAB.1	None	None
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	Identification of the claimed user identity.

Table 6-1: Auditable Events

Each audit record includes the date and time of the event, type of event, subject identity (if applicable), the outcome (success or failure) of the event; and for each audit event type, based on the auditable event definitions of the functional components, information specified in column three of Table 6-1: Auditable Events. For audit events resulting from actions of identified users, the TOE associates each auditable event with the identity of the user that caused the event.

The audit log is stored locally by default, and the TOE protects the stored audit records in the audit trail from unauthorized deletion and prevents unauthorized modifications to the stored audit records in the audit trail.

Specifically, the security related audit logs are kept in two separate log files:

- Audit Log – Maintains audit records for all admin management functionality;
- syslog (this is an internal system log file, not a Syslog server)– Maintains audit records of DNS and network traffic;

The Audit and syslog are accessible to users with superuser admin privileges. The maximum size of the log file is 100 MB. The TOE deletes the oldest log file when audit storage space is full.

The local time source supports the reliable time stamp for the audit function.

The Audit Log and syslog records are transmitted to an external Syslog Server. Audit records are transmitted through a secure TLS connection immediately after they are generated

The TOE does not offer the ability to start and stop the audit function independently from the starting and stopping of the TOE. Audit startup and shutdown are implied by system start and shutdown, both of which are audited. The administrator may choose brief or detailed audit.

The Security Audit function is designed to satisfy the following security functional requirements:

- FAU_GEN.1: Audit records are generated for the appropriate security relevant events and include the date and time of the event, type of event, subject identity (if applicable) and outcome of the event.
- FAU_GEN.2: The TOE associates each auditable event resulting from actions of identified users with the identity of the user that caused the event.

- FAU_STG.1: The TOE protects the stored audit records in the audit trail from unauthorised deletion and prevents unauthorised modifications to the stored audit records in the audit trail.
- FAU_STG.4: The TOE overwrites the oldest stored audit records when the audit trail is full.

6.2 Cryptographic Support

The TOE uses cryptography to support integrity checks of TOE update images and for the protection of the following types of communication pathways:

- **Grid communication.** The TOE may be configured to communicate with other TOE instances in a grid. This communication is protected via an SSL/TLS VPN.
- **Remote Administration.** Remote administrators configure the TOE via a web based GUI that is protected using TLS/HTTPS or via CLI protected using SSH.
- **Application Programming Interface.** The TOE provides a Perl API to assist integration of the Infoblox device into network environments. The API is protected using TLS/HTTPS. The Perl API provides interfaces to DHCP, DNS, Grid and IPAM services.
- **Trusted external entities: remote syslog and authentication servers.** The TOE initiates outbound TLS tunnels to transmit audit logs to remote syslog servers. In addition, TLS is used to secure the session between the TOE and the Active Directory and LDAP authentication servers.

The TOE uses RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits for integrity checking of TOE update images prior to installing those updates and during the boot process. The TOE verifies the digital signature associated with the TOE update and will not load an image that fails an integrity check.

The TOE uses the OpenVPN/OpenSSL implementation of TLS to establish a VPN for Grid communication between instances of the TOE (when so configured). This implementation has the following characteristics:

- Key exchange (256 bit) is as per TLS RFC 2246 with OpenVPN specified as the transport protocol.
- All packets sent over the VPN after the key exchange is encrypted with AES-256-CBC.
- Authentication and integrity are provided using HMAC as per IPsec Authentication Header (AH) described in RFC 2402. Note: The TOE does not implement the full IPsec protocol.

Remote administrative management sessions are initiated by a login and occurs over HTTPS using TLS or SSH. Remote administrative GUI or API sessions use HTTPS/TLS. A remote administrative session to the CLI occurs over SSH protected ethernet connection. TOE to TOE communication occurs for the purpose of distributing configuration information from one instance of the TOE to another. The TOE ensures that such communication occurs only over a TLS protected communication pathway. The TOE initiates outbound TLS tunnels to transmit audit logs to remote syslog servers. In addition, TLS is used to secure the session between the TOE and the Active Directory and LDAP authentication servers. TLS provides protection of the communications pathways from disclosure and from undetected modification.

The TOE implements the HTTPS protocol that complies with RFC 2818 and implements TLS versions 1.0, 1.1 and 1.2, supporting the following ciphersuites:

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA (1.0, 1.1)
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA (1.0, 1.1)
- TLS_RSA_WITH_AES_128_CBC_SHA (1.0, 1.1)
- TLS_RSA_WITH_AES_256_CBC_SHA (1.0, 1.1)
- TLS_RSA_WITH_AES_128_CBC_SHA256 (1.2)
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (1.2)
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (1.2)
- TLS_RSA_WITH_AES_128_CBC_SHA256 (1.2)

The ciphersuite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA is used for Secure Grid Communications.
The TOE uses OpenSSL cryptography to implement all cryptographic functions.

The following ciphersuites are used for connections with AD, LDAP and Syslog:

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA (1.0, 1.1)
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA (1.0, 1.1)
- TLS_RSA_WITH_AES_128_CBC_SHA (1.0, 1.1)
- TLS_RSA_WITH_AES_256_CBC_SHA (1.0, 1.1)

The TOE implements the SSHv2 protocol for remote administrators accessing the CLI and supports the following algorithms.

- Encryption: aes128-cbc and aes256-cbc
- Key exchange: ecdh-sha2-nistp256; ecdh-sha2-nistp384; ecdh-sha2-nistp521; diffie-hellman group exchange sha256; diffie-hellman group exchange sha1; diffie-hellman group 14 sha1; diffie-hellman group 1 sha1
- MAC: hmac-sha1; hmac-sha1-etm@openssh.com
- Public key authentication: ssh-rsa

The CAVP certificate numbers are listed in the Table below.

Algorithm	FIPS Certification Number
AES	#4805
rDSA	#2633
SHS	#3953
HMAC	#3215
CTR_DRBG (AES)	#1671

Table 6-2: OpenSSL FIPS Object Module Certificates

In support of secure cryptographic protocols, the TOE supports key establishment schemes, as specified in FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendices B.1, B.3, and B.4.

The TOE zeroizes all plaintext secret and private cryptographic keys and Cryptographic Critical Security Parameters (CSPs) when no longer required. While the administrator could directly read RAM or persistent memory to view CSPs, they are trusted not to do so.

The TOE performs encryption and decryption in accordance with cryptographic algorithm AES operating in CBC mode and cryptographic key sizes 128-bits and 256-bits. AES is performed as specified in ISO 18033-3, and CBC as specified in ISO 10116. AES is implemented in the following protocols: TLS, SSH.

The TOE performs cryptographic signature services in accordance with RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater. It meets the requirements of RSA Digital Signature in FIPS PUB 186-4, “Digital Signature Standard (DSS)”.

The TOE performs cryptographic hashing services in accordance with SHA-1, SHA-256, SHA-384, and SHA-512 and message digest sizes 160, 256, 384, and 512 bits. ISO/IEC 10118-3:2004 is met for SHA implementation.

The TOE performs keyed-hash message authentication in accordance with HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512, key size 128, 256 and 512 bits, and message digest sizes 160, 256, 384, 512. ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2” is met for the HMAC implementation.

The TOEs random numbers are generated in accordance with ISO/IEC 18031:2011 using a Deterministic Random Bit Generator HMAC_DRBG (any). The deterministic RBG is seeded by one entropy source that accumulates entropy from one hardware -based noise source with a minimum of 256 bits of entropy at least equal to the greatest

security strength, according to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”, of the keys and hashes that it generates.

The table below identifies all secret and private keys and CSPs used to generate keys, the related zeroization procedures. No interfaces are provided to view the plaintext key.

Key/CSP	Location	Zeroization Procedure
HTTPS server private key: 2048 bit RSA	Encrypted database file	Overwritten when no longer in use; three times with a random pattern, and once with zeroes.
Static symmetric key for decrypting software updates (AES-256-CBC)	Compiled into a library binary.	None. Key remains static.
Static symmetric key for encrypting/decrypting database backups (AES-128-CBC)	Compiled into a library binary.	None. Key remains static.
Administration session cookie HMAC key: HMAC-SHA1	File	Overwritten when no longer in use; three times with a random pattern, and once with zeroes.
TLS/SSH session keys	Stored in memory	Overwritten with zeroes when no longer in use.
RBG state seed	Process memory	The generator state is overwritten with zeroes when the generator process exits, at system shutdown.

The Cryptographic support function is designed to satisfy the following security functional requirements:

- FCS_CKM.1: The TOE generates asymmetric cryptographic keys for use in key establishment. These keys meet the recommendations of FIPS PUB 186-4 for RSA-based schemes, elliptic curve-based schemes, and finite field-based key establishment schemes using key sizes of 2048 bits or greater.
- FCS_CKM.4: The TOE clears, by overwriting with zeros, plaintext secret and private cryptographic keys and Cryptographic Critical Security Parameters (CSPs) when no longer required.
- FCS_COP.1(1): The TOE implements AES for encryption and decryption of data as described above to meet the standards: AES as specified in ISO 18033-3, CBC as specified in ISO 10116, with the bit sizes and mode described above and in the OpenSSL security policy.
- FCS_COP.1(2): The TOE performs RSA Digital Signature Algorithm (rDSA) with cryptographic key sizes 2048 bits or greater that meet FIPS PUB 186-4.
- FCS_COP.1(3): The TOE implements SHA-1, SHA-256, SHA-384, SHA-512 for hashing services as described above that meet ISO/IEC 10118-3:2004 with the required message digest sizes.
- FCS_COP.1(4): The TOE implements HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 for keyed-hash authentication as described above that meet ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2” with the required key sizes.
- FCS_COP.1(5): The TOE implements HMAC_DRBG(any) with a minimum entropy seed of 256-bits that meet ISO/IEC 18031:2011.

6.3 Identification and Authentication

The TOE provides the administrator access to the TOE via local console port, SSH and HTTPS. The TOE provides a password-based logon mechanism for authorized access to the TOE. The authentication policy can be configured to

specify whether authentication uses a local store or through invoking authentication services from a remote Active Directory, LDAP, RADIUS or TACACS+ server.

Before establishing a user/administrator session, the TOE displays an Administrator-specified advisory notice and consent warning message regarding unauthorized use of the TOE. The TOE requires each user to be successfully identified and authenticated before allowing any other actions on behalf of that user. The TSF provides only obscured feedback to the user while the authentication is in progress at the local console and remote access methods (SSH, and HTTPS).

The TOE enforces the following password policy for administrative passwords:

- Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and special characters (that include: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")");
- Minimum password length shall settable by the Authorized Administrator, and support passwords of 15 characters or greater;
- Passwords shall have a maximum lifetime;
- New passwords must contain a minimum of 4 character changes from the previous password.

The passwords composition rules apply to locally defined admins and are configurable by the superuser admin.

The TOE requires users with expired passwords to create a new password after correctly entering the expired password. The TOE re-authenticates the user when the user changes their password, or following session locking.

For remote access, after the user is authenticated, the TOE checks for the user roles before the dashboard is displayed. The available options on the dashboard are determined by the user role.

For local console access, only users with the superuser admin privilege may use this interface. Once authenticated, the superuser admin is provided an Infoblox console prompt.

The Identification and Authentication function is designed to satisfy the following security functional requirements:

- FIA_SOS.1: The TOE provides a mechanism to verify that secrets meet a defined quality metric.
- FIA_UAU.2: The TOE requires all users to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
- FIA_UID.2: The TOE requires all users to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.
- FIA_UAU.5: The TOE supports user authentication using a local password mechanism and can be configured to use Active Directory, LDAP, RADIUS, TACACS+ authentication.
- FIA_UAU.7: The TOE provides only obscured feedback to the user while the authentication is in progress.

6.4 Security Management

A user must have an admin account to log in to the TOE. Each admin account belongs to an admin group, which contains roles and permissions that determine the tasks a user can perform.

The TOE provides interfaces to manage its security functions and data. The GUI interfaces and API's are accessed remotely through HTTPS (using TLS) and the CLI is accessed via local console or remotely using SSH. All users accessing the TOE must be identified and authenticated. Access to security management functions is restricted to specific user roles.

Administrative users inherit access privileges from the admin group that they are member of based on the roles and permissions assigned to that group. There are three types of admin groups:

• Superuser – Superuser admin groups provide their members with unlimited access and control of all the operations that a NIOS appliance performs. There is a default superuser admin group, called admin-group, with one superuser administrator, admin. Only superusers can create admin groups.

• Limited-Access – Limited-access admin groups provide their members with read-only or read/write access to specific resources.

• Default – When upgrading from previous NIOS releases, the appliance converts the ALL USERS group to the Default Group when the ALL USERS Group contains admin accounts. The appliance does not create the Default Group if there is no permission in the ALL USERS group. The permissions associated with the ALL USERS group are moved to a newly created role called Default Role. Supported in previous NIOS releases, the ALL USERS group was a default group in which you defined global permissions for all limited-access users. This group implicitly included all limited-access users configured on the appliance.

The TOE provides the following system-defined admin roles:

- superuser admin
- DHCP Admin
- DNS Admin
- DTC Admin
- File Distribution Admin
- Grid Admin
- Load Balancer Admin
- PKI Admin
- DHCP Fingerprint Admin

The superuser admin privilege gives full access to the TOE. The other roles can be assigned to the Limited-Access group and provide privileges to manage resources or services such as DNS and DHCP. These roles do not provide any permissions to the TOE management functions. The scope of the evaluation does not include the system-defined admin roles that can be assigned to the Limited-Access group since all of the security management functions are performed by the superuser admin role which belongs to the superuser group. The Default group consists of Limited-Access users imported at upgrade with previously defined permissions, The evaluated configuration requires a fresh install and therefore the Default role will not be assigned. The TOE construct of Authorized Administrator equates to a TOE administrative user with the superuser admin role. All security management functions are performed by the superuser admin.

The TSF is capable of performing the following management functions:

- Manage authentication policy
- Manage password policy
- Manage user creation/modification
- Manage the TOE banner
- Manage TOE updates
- Manage TOE session Inactivity
- Manage audit configuration
- Manage TOE system time
- Manage passwords

Management and modification of the behavior of the functions is restricted to the superuser admin.

Management functions description:

Manage password policy – Gives the administrator the ability to define the global policy for password metrics.

Manage user creation/modification – Gives the administrator the ability to create, modify, and delete user accounts and user groups.

Manage authentication policy – Gives the administrator the ability to define method of authentication whether local or remote and identifying the remote authentication server. This is a group policy setting for all users within the specified user group.

Manage the TOE banner – Gives the administrator the ability to configure the warning message that users see at the login display.

Manage TOE updates- Gives the administrator the ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates.

Manage TOE Session Inactivity – Gives the administrator the ability to define interactive session timers.

Manage audit configuration– Gives the administrator the ability to configure the location of the external syslog server and which logs are to be transmitted. Gives the administrator the ability to configure the level of detail recorded in the audit logs: brief or detailed. Detailed level must be chosen for the TOE to generate all of the specified audit records.

Manage TOE system time – Gives the administrator the ability to change the local system time and configure the use of a NTP server.

Manage passwords – Gives the administrator the ability to change their own passwords in addition to the passwords of other administrators. All users can change their own passwords.

The Security Management function is designed to satisfy the following security functional requirements:

- FMT_MOF.1: The TOE restricts the ability to modify the behaviour of the management functions to the superuser.
- FMT_MTD.1: The TOE ensures that only authorized administrators can modify the TSF configuration data.
- FMT_SMF.1: The TOE provides management functions identified in the text above to support an administrator's ability to securely configure and operate the TOE as described in the above section.
- FMT_SMR.1: The TOE maintains the security role of superuser and is able to associate users with this role.

6.5 Protection of the TSF

When Secure Grid is configured, all communication between TOE instances is protected using TLS VPN. The TOE uses the OpenVPN/OpenSSL implementation of TLS to establish a VPN for Grid communication between instances of the TOE. Authentication and integrity are provided using HMAC as per IPsec Authentication Header (AH) described in RFC 2402.

The TOE provides a source of date and time information used in audit event timestamps. The clock function is reliant on the system clock provided by the underlying hardware. The TOE can optionally be set to receive clock updates from an NTP server. This date and time is used as the time stamp that is applied to TOE generated audit records and used to track inactivity of administrative sessions.

The TOE implements self-test, during initial startup, to determine whether the TOE is operating correctly. The self-test includes:

- Memory test using the MemBIST test from the Intel memory reference code (MRC); at the end of the test faulty isolated memory are disabled;
- Cryptographic libraries test where library content is compared to a stored checksum;
- Crypto algorithms known answer tests are executed for all algorithms;
- Random number generation test, which continuously test the seed entropy using a Repetition Count test and an Adaptive Proportion test.

If any of these tests fail, the system startup will be aborted and an error message will be displayed on the serial console. Otherwise, the login prompt will be displayed showing that the system is operating correctly.

An authorized administrator can query the software version running on the TOE, and can initiate updates to the TOE software images. When software updates are made available by Infoblox, an administrator can obtain, verify the integrity of, and install those updates. The updates can be downloaded from the support.infoblox.com. The TOE image files are digitally signed so their integrity can be verified using a digital signature mechanism. The integrity checking is performed prior to installing those updates and during the boot process. An image that fails an integrity check will not be loaded. The digital certificates used by the update verification mechanism are contained on the TOE. Specifically, Infoblox generates RSA digital signatures for TOE updates to ensure that the update can be trusted. The TOE verifies the digital signature associated with the TOE update. The certificate used for validation is stored in a protected file on the appliance. Detailed instructions for how to perform verification are provided in the administrator guidance for this evaluation.

The CLI shows the version of the TOE on login and provides a command to show the version of TOE and serial number of unit. Upgrade functionality allows for updating the TOE software after validating a digital signature on the software.

The Protection of the TSF function is designed to satisfy the following security functional and assurance requirement:

- FPT_ITT.1: The TOE protects TSF data from disclosure and modification when it is transmitted between separate parts of the TOE.
- FPT_STM.1: The TOE provides reliable time stamps.
- FPT_TST_EXT.1: The TOE implements self-test, during initial startup, to determine whether the TOE is operating correctly.
- FPT_TUD_EXT.1: The TOE provides administrators the ability to: query the current version of the TOE firmware/software; and initiate updates to TOE firmware/software. The TOE provides a digital signature mechanism to verify firmware/software updates to the TOE prior to installing those updates.

6.6 TOE Access

The TOE terminates local and remote interactive sessions after an administrator configurable time interval of 60 and 31536000 seconds (one minute – one year). The default session timeout is 600 seconds (10 minutes). The TOE allows user-initiated termination of the user's own interactive session.

Before establishing a user/administrator session, the TOE displays an administrator configured advisory banner and consent warning message regarding unauthorized use of the TOE.

The TOE Access function is designed to satisfy the following security functional requirements:

- FTA_SSL.3: The TOE terminates an interactive session after an administrator configurable time interval.
- FTA_SSL.4: The TOE allows user-initiated termination of the user's own interactive session.
- FTA_TAB.1: Before establishing a user session, the TSF displays an advisory warning message regarding unauthorised use of the TOE.

6.7 Trusted Path Channels

The TOE communicates with other network devices, as well as administrators, over the network. The critical communication paths and their related protection mechanisms are as follows:

- **Remote Administration -** Remote administrators manage the TOE via SSH or a web based GUI that is protected using HTTPS/TLS.
- **Syslog Server** – All communication with the syslog server is protected with TLS
- **Active Directory and LDAP Servers -** All communication with these external authentication servers is protected with TLS.

The Protection of the TSF function is designed to satisfy the following security functional and assurance requirements:

- FTP_ITC.1: The TOE utilizes TLS to protect all data transmitted between the TOE and trusted external third-party IT entities from unauthorised disclosure and detection of modification during transmission.
- FTP_TRP.1: Remote Administrators connect to the TOE using SSH or HTTPS/TLS to use the administrative CLI or GUI (respectively) for management of the TOE. The initial administrator authentication operation, as well as all subsequent remote administration actions, occur through these channels.

7 Rationale

This section provides the rationale for completeness and consistency of the Security Target. The rationale addresses the following areas:

- Security Objectives
- Security Functional Requirements
- Security Assurance Requirements
- Requirement Dependencies
- TOE Summary Specification.

7.1 Security Objectives Rationale

This section shows that all secure usage assumptions, organizational security policies, and threats are completely covered by security objectives. In addition, each objective counters or addresses at least one assumption, policy or threat.

7.1.1 Security Objectives Rationale for the TOE and Environment

This section shows that all secure usage assumptions and threats are completely covered by security objectives for the TOE or operational environment. In addition, each objective counters or addresses at least one assumption or threat.

	T.PASSWORD_CRACKING	T.SECURITY_FUNCTIONALITY_FAILURE	T. UNAUTHORIZED_ADMINISTRATOR_ACCESS	T.UNDETECTED_ACTIVITY	T.UNTRUSTED_COMMUNICATION_CHANNELS	T.UPDATE_COMPROMISE	T.WEAK_AUTHENTICATION_ENDPOINTS	T. WEAK_CRYPTOGRAPHY	P.ACCESS_BANNER	A.ADMIN_CREDENTIALS_SECURE	A.LIMITED_FUNCTIONALITY	A.NO_THRU_TRAFFIC_PROTECTION	A. TRUSTED_ADMINISTRATOR	A.PHYSICAL_PROTECTION	A. .REGULAR_UPDATES
O.AUDIT_GENERATION				X											
O.DISPLAY_BANNER			X						X						
O.PROTECTED_COMMUNIATIONS					X		X								
O.STRONG_CRYPTOGRAPHY					X	X	X	X							
O.TOE_ADMINISTRATION	X		X												
O.TSF_SELF_TEST		X													
O.VERIFIABLE_UPDATES						X									
OE.ADMIN_CREDENTIALS_SECURE										X					
OE. NO_GENERAL_PURPOSE											X				
OE. NO_THRU_TRAFFIC_PROTE												X			

	T.PASSWORD_CRACKING	T.SECURITY_FUNCTIONALITY_FAILURE	T. UNAUTHORIZED_ADMINISTRATOR_ACCESS	T.UNDETECTED_ACTIVITY	T.UNTRUSTED_COMMUNICATION_CHANNELS	T.UPDATE_COMPROMISE	T.WEAK_AUTHENTICATION_ENDPOINTS	T. WEAK_CRYPTOGRAPHY	P.ACCESS_BANNER	A.ADMIN_CREDENTIALS_SECURE	A.LIMITED_FUNCTIONALITY	A.NO_THRU_TRAFFIC_PROTECTION	A. TRUSTED_ADMINISTRATOR	A.PHYSICAL_PROTECTION	A. .REGULAR_UPDATES
CTION															
OE.PHYSICAL														X	
OE. TRUSTED_ADMIN													X		
OE.UPDATES															X

Table 7-1: Security Problem Definition to Security Objective Correspondence

T.PASSWORD_CRACKING

Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device.

This threat is countered by ensuring that:

- O.TOE_ADMINISTRATION: The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, mechanisms that control a user's logical access to the TOE and mechanisms to ensure strong passwords.

T.SECURITY_FUNCTIONALITY_FAILURE

A component of the network device may fail during start-up or during operations causing a compromise or failure in the security functionality of the network device, leaving the device susceptible to attackers.

This threat is satisfied by ensuring that:

- O.TSF_SELF_TEST: The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.

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.

This threat is countered by ensuring that:

- 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, mechanisms that control a user's logical access to the TOE and mechanisms to ensure strong passwords.

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.

This assumption is addressed by ensuring that:

- O.AUDIT_GENERATION: The TOE will provide the capability to detect and create records of security relevant events associated with users; and store those audit data locally, or externally if configured.

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.

This threat is countered by ensuring that:

- O. PROTECTED_COMMUNICATIONS: The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.
- O.STRONG_CRYPTOGRAPHY: The TOE will provide strong standards-based cryptographic algorithms and key sizes.

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.

This threat is countered by ensuring that:

- 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 cryptographically validated to be from a trusted source.
- O.STRONG_CRYPTOGRAPHY: The TOE will provide strong standards-based cryptographic algorithms and key sizes.

T. WEAK_AUTHENTICATION_ENDPOINTS

Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g., shared password that is guessable or transported as plaintext.

This threat is countered by ensuring that:

- O. PROTECTED_COMMUNICATIONS: The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.
- O.STRONG_CRYPTOGRAPHY: The TOE will provide strong standards-based cryptographic algorithms and key sizes.

T. WEAK_CRYPTOGRAPHY

Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space.

This threat is countered by ensuring that:

- O.STRONG_CRYPTOGRAPHY: The TOE will provide strong standards-based cryptographic algorithms and key sizes.

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.

This policy is covered by ensuring that:

- O.DISPLAY_BANNER: The TOE will display an advisory warning regarding use of the TOE.

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.

This assumption is addressed by ensuring that:

- 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.

A. LIMITED_FUNCTIONALITY

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

This assumption is addressed by ensuring that:

- 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.

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 TOE.

This assumption is addressed by ensuring that:

- 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.

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.

This assumption is addressed by ensuring that:

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

A. TRUSTED_ADMINISTRATOR

The Authorized 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.

This assumption is addressed by ensuring that:

- OE. TRUSTED_ADMIN: TOE Administrators are trusted to follow and apply all guidance documentation in a trusted manner.

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.

This assumption is addressed by ensuring that:

- 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.

7.2 Security Requirements Rationale

All security functional requirements identified in this Security Target are fully addressed in this section and each is mapped to the objective it is intended to satisfy. Table 7-2: Objective to Requirement Correspondence summarizes the correspondence of functional requirements to TOE security objectives.

7.2.1 Security Functional Requirements Rationale

All of the Security Functional Requirements (SFRs) identified in this Security Target are fully addressed in this section and each SFR is mapped to the objective it is intended to satisfy.

	O.AUDIT_GENERATION	O.DISPLAY_BANNER	O.PROTECTED_COMMUNICATIONS	O.STRONG_CRYPTOGRAPHY	O.TOE_ADMINISTRATION	O.TSF_SELF_TEST	O.VERIFIABLE_UPDATES
FAU_GEN.1	X						
FAU_GEN.2	X						
FAU_STG.1	X						
FAU_STG.4	X						
FCS_CKM.1			X	X			
FCS_CKM.4			X	X			
FCS_COP.1(1)			X	X			
FCS_COP.1(2)			X	X			X
FCS_COP.1(3)			X	X			
FCS_COP.1(4)			X	X			
FCS_COP.1(5)			X	X			
FIA_SOS.1					X		
FIA_UAU.2					X		
FIA_UAU.5					X		
FIA_UAU.7					X		
FIA_UID.2					X		
FMT_MOF.1					X		

	O.AUDIT_GENERATION	O.DISPLAY_BANNER	O.PROTECTED_ COMMUNICATIONS	O.STRONG_CRYPTOGRAPHY	O.TOE_ADMINISTRATION	O.TSF_SELF_TEST	O.VERIFIABLE_UPDATES
FMT_MTD.1					X		
FMT_SMF.1					X		
FMT_SMR.1					X		
FPT_ITT.1			X				
FPT_STM.1	X						
FPT_TST_EXT.1						X	
FPT_TUD_EXT.1							X
FTA_SSL.3					X		
FTA_SSL.4					X		
FTA_TAB.1		X					
FTP_ITC.1			X				
FTP_TRP.1			X				

Table 7-2: Objective to Requirement Correspondence

O.AUDIT_GENERATION

The TOE will provide the capability to detect and create records of security relevant events associated with users; and store those audit data locally, or externally if configured.

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.1: The TOE is required to provide a set of events that it is capable of recording. Among these events the TOE is able to audit must be security relevant events occurring within the TOE. This requirement also defines the information that must be recorded for each auditable event.
- FAU_GEN.2: The TOE is required to associate a user identity resulting from actions of identified users with the identity of the user that caused the event.
- FAU_STG.1: The TOE is required to protect the stored audit records in the audit trail from unauthorised deletion and prevents unauthorised modifications to the stored audit records in the audit trail.
- FAU_STG.4: The TOE is required to overwrite the oldest stored audit records when the audit trail is full.
- FPT_STM.1: The TOE is required to provide reliable time stamps for its own use. The timestamps are used in the audit function.

O. DISPLAY_BANNER

The TOE will display an advisory warning regarding use of the TOE.

This TOE Security Objective is satisfied by ensuring that:

- FTA_TAB.1: Before establishing a user session, the TOE is required to display an advisory warning message regarding unauthorised use of the TOE.

O.PROTECTED_COMMUNICATIONS

The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.

This TOE Security Objective is satisfied by ensuring that:

- FCS_CKM.1: The TOE is required to generate asymmetric cryptographic keys for use in key establishment. These keys meet the recommendations of FIPS PUB 186-4 for RSA-based key establishment schemes using key sizes of 2048 bits or greater.
- FCS_CKM.4: The TOE is required to clear, by overwriting with zeros, plaintext secret and private cryptographic keys and Cryptographic Critical Security Parameters (CSPs) when no longer required.
- FCS_COP.1(1): The TOE is required to implement AES for encryption and decryption of data as described above to meet the standards: AES as specified in ISO 18033-3, CBC as specified in ISO 10116, with the bit sizes and mode described above and in the OpenSSL security policy.
- FCS_COP.1(2): The TOE is required to performs RSA Digital Signature Algorithm (rDSA) with cryptographic key sizes 2048 bits or greater that meet FIPS PUB 186-4.
- FCS_COP.1(3): The TOE is required to implement SHA-1, SHA-256, SHA-384, SHA-512 for hashing services as described above that meet ISO/IEC 10118-3:2004 with the required message digest sizes.
- FCS_COP.1(4): The TOE is required to implement HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 for keyed-hash authentication as described above that meet ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2" with the required key sizes.
- FCS_COP.1(5): The TOE is required to implement HMAC_DRBG(any) with a minimum entropy seed of 256-bits that meet ISO/IEC 18031:2011.
- FPT_ITT.1: The TOE is required to protect TSF data from disclosure and modification when it is transmitted between separate parts of the TOE.
- FTP_ITC.1: The TOE is required to utilize TLS to protect all data transmitted between the TOE and trusted external third-party IT entities from unauthorised disclosure and detection of modification during transmission.
- FTP_TRP.1: The TOE requires remote Administrators to connect to the TOE using HTTPS/TLS in order to use the administrative GUI for management of the TOE. The initial administrator authentication operation, as well as all subsequent remote administration actions, occurs through this channel.

O.STRONG_CRYPTOGRAPHY

The TOE will provide strong standards-based cryptographic algorithms and key sizes.

This TOE Security Objective is satisfied by ensuring that:

- FCS_CKM.1: The TOE is required to generate asymmetric cryptographic keys for use in key establishment. These keys meet the recommendations of FIPS PUB 186-4 for RSA-based key establishment schemes using key sizes of 2048 bits or greater.
- FCS_CKM.4: The TOE is required to clear, by overwriting with zeros, plaintext secret and private cryptographic keys and Cryptographic Critical Security Parameters (CSPs) when no longer required.
- FCS_COP.1(1): The TOE is required to implement AES for encryption and decryption of data as described above to meet the standards: AES as specified in ISO 18033-3, CBC as specified in ISO 10116, with the bit sizes and mode described above and in the OpenSSL security policy.

- FCS_COP.1(2): The TOE is required to performs RSA Digital Signature Algorithm (rDSA) with cryptographic key sizes 2048 bits or greater that meet FIPS PUB 186-4.
- FCS_COP.1(3): The TOE is required to implement SHA-1, SHA-256, SHA-384, SHA-512 for hashing services as described above that meet ISO/IEC 10118-3:2004 with the required message digest sizes.
- FCS_COP.1(4): The TOE is required to implement HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 for keyed-hash authentication as described above that meet ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2" with the required key sizes.
- FCS_COP.1(5): The TOE is required to implement HMAC_DRBG(any) with a minimum entropy seed of 256-bits that meet ISO/IEC 18031:2011.

O.TOE_ ADMINISTRATION

The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, mechanisms that control a user's logical access to the TOE and mechanisms to ensure strong passwords.

This TOE Security Objective is satisfied by ensuring that:

- FIA_SOS.1: The TOE is required to provide a mechanism to verify that secrets meet a defined quality metric.
- FIA_UAU.2: The TOE requires all users to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
- FIA_UID.2: The TOE requires all users to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.
- FIA_UAU.5: The TOE is required to support user authentication using a local password mechanism and can be configured to use Active Directory, LDAP, RADIUS, TACACS+ authentication.
- FIA_UAU.7: The TOE is required to provide only obscured feedback to the user while the authentication is in progress.
- FMT_MOF.1: The TOE is required to restrict the ability to modify the behaviour of the management functions to the superuser.
- FMT_MTD.1: The TOE is required to ensure that only authorized administrators can modify the TSF configuration data.
- FMT_SMF.1: The TOE is required to provide management functions to support an administrator's ability to securely configure and operate the TOE.
- FMT_SMR.1: The TOE is required to maintain the security role of superuser and is able to associate users with this role.
- FTA_SSL.3: The TOE is required to terminate an interactive session after an administrator configurable time interval.
- FTA_SSL.4: The TOE allows user-initiated termination of the user's own interactive session.

O. TSF_SELF_TEST

The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.

This TOE Security Objective is satisfied by ensuring that:

- FPT_TST_EXT.1: The TOE is required to implement self-tests, during initial startup, to determine whether the TOE is operating correctly.

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 cryptographically validated to be from a trusted source.

This TOE Security Objective is satisfied by ensuring that:

- FPT_TUD_EXT.1: The TOE is required to provide administrators the ability to: query the current version of the TOE firmware/software; and initiate updates to TOE firmware/software. The TOE provides a digital signature mechanism to verify firmware/software updates to the TOE prior to installing those updates.
- FCS_COP.1(2): The TOE is required to perform RSA Digital Signature Algorithm (rDSA) with cryptographic key sizes 2048 bits or greater that meet FIPS PUB 186-4.

7.2.2 Security Assurance Requirements Rationale

The security assurance requirements for the TOE are the EAL 2 augmented with the ALC_FLR.2 component as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

EAL 2 augmented with ALC_FLR.2 was selected as the assurance level because the TOE is a commercial product whose users require a low to moderate degree of independently assured security. ALC_FLR.2 was selected to exceed EAL2 assurance objectives in order to ensure that identified flaws are addressed. The TOE is targeted at a relatively benign environment with good physical access security and competent administrators. Within such environments it is assumed that attackers will have little attack potential. As such, EAL 2 augmented with ALC_FLR.2 is appropriate to provide the assurance necessary to counter the limited potential for attack.

7.3 Requirement Dependency Rationale

The following table demonstrates the dependencies among the claimed security requirements. It shows that all dependencies are satisfied. Therefore the requirements work together to accomplish the overall objectives defined for the TOE.

ST Requirement	CC Dependencies	ST Dependencies
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1 and FIA_UID.1	FAU_GEN.1 and FIA_UID.2
FAU_STG.1	FAU_GEN.1	FAU_GEN.1
FAU_STG.4	FAU_STG.1	FAU_STG.1
FCS_CKM.1	(FCS_CKM.2 or FCS_COP.1) and FCS_CKM.4	FCS_COP.1(1), FCS_COP.1.1(2) and FCS_CKM.4
FCS_CKM.4	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	FCS_CKM.1
FCS_COP.1(1)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM.4
FCS_COP.1(2)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM.4
FCS_COP.1(3)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM.4
FCS_COP.1(4)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM.4
FCS_COP.1(5)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	See DRBG Note Below.
FIA_SOS.1	None	None
FIA_UAU.2	FIA_UID.1	FIA_UID.2
FIA_UID.2	None	None
FIA_UAU.5	None	None
FIA_UAU.7	FIA_UAU.1	FIA_UAU.2
FMT_MOF.1	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1

ST Requirement	CC Dependencies	ST Dependencies
FMT_SMF.1	None	None
FMT_SMR.1	FIA_UID.1	FIA_UID.2
FPT_ITT.1	None	None
FPT_STM.1	None	None
FPT_TST_EXT.1	None	None
FPT_TUD_EXT.1	FCS_COP.1(2)	FCS_COP.1(2)
FTA_SSL.3	None	None
FTA_SSL.4	None	None
FTA_TAB.1	None	None
FTP_ITC.1	None	None
FTP_TRP.1	None	None
ADV_ARC.1	ADV_FSP.1, ADV_TDS.1	ADV_FSP.2, ADV_TDS.1
ADV_FSP.2	ADV_TDS.1	ADV_TDS.1
ADV_TDS.1	ADV_FSP.2	ADV_FSP.2
AGD_OPE.1	ADV_FSP.1	ADV_FSP.2
AGD_PRE.1	None	None
ALC_CMC.2	ALC_CMS.1	ALC_CMS.2
ALC_CMS.2	None	None
ALC_DEL.1	None	None
ALC_FLR.2	None	None
ASE_CCL.1	ASE_INT.1, ASE_ECD.1, ASE_REQ.1	ASE_INT.1, ASE_ECD.1, ASE_REQ.2
ASE_ECD.1	None	None
ASE_INT.1	None	None
ASE_OBJ.2	ASE_SPD.1	ASE_SPD.1
ASE_REQ.2	ASE_ECD.1, ASE_OBJ.2	ASE_ECD.1, ASE_OBJ.2
ASE_SPD.1	None	None
ASE_TSS.1	ADV_FSP.1, ASE_INT.1, ASE_REQ.1	ADV_FSP.2, ASE_INT.1, ASE_REQ.2
ATE_COV.1	ADV_FSP.2 and ATE_FUN.1	ADV_FSP.2 and ATE_FUN.1
ATE_FUN.1	ATE_COV.1	ATE_COV.1
ATE_IND.2	ADV_FSP.2 and AGD_OPE.1 and AGD_PRE.1 and ATE_COV.1 and ATE_FUN.1	ADV_FSP.2 and AGD_OPE.1 and AGD_PRE.1 and ATE_COV.1 and ATE_FUN.1
AVA_VAN.2	ADV_ARC.1 and ADV_FSP.2 and ADV_TDS.1 and AGD_OPE.1 and AGD_PRE.1	ADV_ARC.1 and ADV_FSP.2 and ADV_TDS.1 and AGD_OPE.1 and AGD_PRE.1

Note: The DRBG algorithm defined in FCS_COP.1(5) is a keyless operation and as such, has no dependency for generation or zeroization of cryptographic keys. DRBG has a random seed, but that is generated from a source of entropy, not from a key generation algorithm.

Table 7-3: Requirement Dependencies

7.4 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all

necessary for the required security functionality in the TSF. **Table 7-4: Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

	Security audit	Cryptographic support	Identification and authentication	Security management	Protection of the TSF	TOE Access	Trusted Path/Channels
FAU_GEN.1	X						
FAU_GEN.2	X						
FAU_STG.1	X						
FAU_STG.4	X						
FCS_CKM.1		X					
FCS_CKM.4		X					
FCS_COP.1(1)		X					
FCS_COP.1(2)		X					
FCS_COP.1(3)		X					
FCS_COP.1(4)		X					
FCS_COP.1(5)		X					
FIA_SOS.1			X				
FIA_UAU.2			X				
FIA_UID.2			X				
FIA_UAU.5			X				
FIA_UAU.7			X				
FMT_MOF.1				X			
FMT_MTD.1				X			
FMT_SMF.1				X			
FMT_SMR.1				X			
FPT_ITT.1					X		
FPT_STM.1					X		
FPT_TST_EXT.1					X		
FPT_TUD_EXT.1					X		
FTA_SSL.3						X	
FTA_SSL.4						X	
FTA_TAB.1						X	
FTP_ITC.1							X
FTP_TRP.1							X

Table 7-4: Security Functions vs. Requirements Mapping