# Infoblox Trinzic Appliances with NIOS v8.5.2

## Security Target

**Version 1.0**

**15 June 2021**

**Prepared for:**


4750 Patrick Henry Drive
Santa Clara, CA 95054

---

**Prepared by:**


Accredited Testing & Evaluation Labs
6841 Benjamin Franklin Drive
Columbia, Maryland 21046

| Revision History | | |
|---|---|---|
| **Date** | **Author** | **Modifications** |
| 15 June 2021 | Leidos | Finalized document version |
| 4 May 2021 | Leidos | Updated guidance documentation to current dates |
| 23 February 2021 | Leidos | Added description of CM procedure for hardware |
| 12 January 2021 | Leidos | Updates based on vender review |
| 23 November 2020 | Leidos | Updates based on vender review |
| 8 October 2020 | Leidos | Updates based on EOR 2 |
| 2 October 2020 | Leidos | Updates based on EOR |
| 5 August 2020 | Leidos | Incorporated additional comments from review |
| 30 June 2020 | Leidos | Incorporate comments from initial review |
| 28 Feb 2020 | Leidos | Initial Draft |

# 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 v8.5.2 identified below in Section 1.1. The TOE appliances are a family of network appliances that provide basic core network services (DNS, DHCP, IPAM, FTP, TFTP, and HTTP); Network Insight Discovery Services; Threat Insight and Response Policy Zone capabilities.

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, Target of Evaluation, and Common Criteria Identification

**ST Title:** Infoblox Trinzic Appliances with NIOS v8.5.2 Security Target

**ST Version:** Version 1.0

**ST Date:** 15 June 2021

**TOE Identification:** Infoblox Trinzic Appliances with NIOS v8.5.2

*Table 1: TOE Appliance Models[1]*

| Series | Physical Appliance | Virtual Appliance[2] |
|---|---|---|
| Infoblox 805 Series | TE-815, TE-825, ND-805, TR-805 | ND-V805, IB-V815, IB-V825 |
| Infoblox 1405 Series | TE-1415, TE-1425, ND-1405, TR-1405 | ND-V1405, IB-V1415, IB-V1425 |

[1] The product documentation indicates that “TE” and “IB” are interchangeable however note that the virtual models always have a “V” in the identifier.

[2] The product documentation sometimes refers to the Infoblox Trinzic Virtual Appliances with NIOS as vNIOS. The term vNIOS is adopted for this Security Target.

| | | |
|---|---|---|
| Infoblox 2205 Series | TE-2215, TE-2225, ND-2205, TR-2205 | ND-V2205, IB-V2215, IB-V2225 |
| Infoblox 4005 Series | TE-4015, TE-4025, ND-4005, TR-4005 | IB-V4005, ND-V4005,<br>IB-V4015,<br>IB-V4025 |
| Infoblox 5005 Series | N/A | IB-V5005 |

The Virtual vNIOS appliances are supported on the following third-party platforms:

- VMware on ESX/ESXi Servers; versions 6.7, 6.5.x, 5.5.x
- KVM Hypervisor and KVM-based OpenStack deployments: RHEL for KVM-based OpenStack v7.4, and v7.5.
- Nutanix AHV v5.11

**TOE Developer:** Infoblox Inc.

**Evaluation Sponsor:** Infoblox Inc.

**CC Identification:** Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017.

## 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 5, April 2017
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components, Version 3.1 Revision 5, April 2017
  - 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 are used 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 1 of the CC defines the approved set of operations that may be applied to functional requirements: iteration; selection; assignment; and refinement.

- Iteration—allows a component to be used more than once with varying operations. In this 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).

- Selection—allows the specification of one or more elements from a list. Selections are indicated using bold italics and are enclosed by brackets (e.g., [***selection***]).
- Assignment—allows the specification of an identified parameter. Assignments are indicated using bold text and are enclosed by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in bold italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
- Refinement—allows the addition of details. Refinements are indicated using bold for additions and strike-through for deletions (e.g., "… ~~some~~ **all** objects).

Other sections of the ST—other sections of the ST use bolding and/or different fonts (such as `Courier`) to highlight text of special interest, such as captions, commands, or filenames specific to the TOE.

## 1.4 Abbreviations and Acronyms

*Table 2: Abbreviations and Acronyms*

| Abbreviation | Definition |
|---|---|
| AES | Advanced Encryption Standard |
| API | Application Programming Interface |
| CAVP | Cryptographic Algorithm Validation Program |
| CBC | Cipher Block Chaining |
| CC | Common Criteria |
| CLI | Command Line Interface |
| DHCP | Dynamic Host Configuration Protocol |
| DNS | Domain Name System |
| DNSSEC | Domain Name System Security Extensions |
| EAL | Evaluation Assurance Level |
| FIPS | Federal Information Processing Standard |
| FTP | File Transfer Protocol |
| GUI | Graphical User Interface |
| HA | High Availability |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |
| IPAM | Internet Protocol Address Management |
| IPSEC | Internet Protocol Security |
| LDAP | Lightweight Directory Access Protocol |
| NIOS | The software component of the TOE |
| NTP | Network Time Protocol |
| OS | Operating System |
| OSP | Organizational Security Policy |
| RADIUS | Remote Authentication Dial-In User Service |

| Abbreviation | Definition |
|---|---|
| REST | Representational State Transfer |
| RPZ | Response Policy Zone |
| SAML | Security Assertion Markup Language |
| SSH | Secure Shell |
| ST | Security Target |
| TACACS+ | Terminal Access Controller Access-Control System |
| TFTP | Trivial File Transfer Protocol |
| TLS | Transport Layer Security |
| TOE | Target of Evaluation |
| TSF | TOE Security Function |
| vNIOS | Infoblox Virtual Appliance with NIOS |
| VPN | Virtual Private Network |

# 2 TOE Description

The Target of Evaluation (TOE) is Infoblox Trinzic Appliances with NIOS v8.5.2, hereinafter referred to as Infoblox Trinzic Appliances, Infoblox Trinzic or the TOE. The TOE includes the NIOS v8.5.2 software, hardware and virtual appliances as identified in Table 1. The TOE appliances are a family of network appliances that provide basic core network services (DNS, DHCP, IPAM, FTP, TFTP, and HTTP); Network Insight Discovery Services; Threat Insight and Response Policy Zone capabilities.

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 Network Insight Discovery Services; Threat Insight; Response Policy Zone (RPZ) capabilities; and Secure Grid functionality.

Secure Grid is the capability of Infoblox appliances to work cooperatively in an enterprise deployment. One appliance is designated Grid Master (GM). The Grid Master appliance distributes configuration information to all other Grid devices (Grid members). Grid communication is secured with OpenVPN. HA (high availability) configurations are supported. HA provides hardware redundancy to minimize service outages. In this configuration, the two nodes that form an HA pair—identified as Node 1 and Node 2—are in an active/passive configuration. The active node receives, processes, and responds to all service requests. The passive node constantly keeps its database synchronized with that of the active node, so it can take over services if a failover occurs. A failover is the reversal of the active/passive roles of each node; that is, when a failover occurs, the previously active node becomes passive and the previously passive node becomes active.

The TOE NIOS operating system is a hardened Linux distribution optimized for security and network performance. Categories of appliances are differentiated by purpose. The Trinzic appliances ('TE', 'IB') provide the basic network device and core functions whereas the Network Insight Appliances provide the Network Insight/Discovery functions. The appliance models within a category are differentiated only 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 SSH/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:

- **Asset Discovery methods (Network Insight)**: Discovery methods used to detect assets and collect data about them include: SNMPv1/v2c device polling; SNMPv3 device polling; CLI device querying; ICMP Ping Sweep and Smart Subnet Ping Sweep; TCP Port Scanning; NetBIOS Queries; and vDiscovery. The protocols can be used to discover and catalogue device types: routers, enterprise switches, firewalls and security appliances, load balancers, enterprise printers, wireless access points, VoIP concentrators, application servers, VRF-based virtual networks, and end hosts. The TOE can be configured to send SNMP and email notifications

when it discovers the following events: Equipment, Software, or Process Failures; Threshold Crossing; Object State Change; and Process Started and Stopped. The subscription-based Advisor Service obtains released Common Vulnerabilities and Exposures (CVEs) and vendor product lifecycle announcements that assists administrators in monitoring and maintaining network and security infrastructure.

- **DNSSEC (DNS Security Extensions)**: implements the DNSSEC Protocol for authenticating the source of DNS data and ensuring its integrity. It protects DNS data from certain attacks, such as man-in the middle attacks and cache poisoning.
- **DNS Firewall** employs **DNS RPZs (Response Policy Zones**): protects external DNS from cyber DNS attacks and internal DNS from infrastructure attacks, data exfiltration, threats and malware. The technology allows reputable sources to dynamically communicate domain name reputation so administrators can implement policy controls for DNS lookup.
- **Threat Insight (also referred to as Threat Analytics in Grid Manager)**: protects mission-critical DNS infrastructure from data exfiltration through DNS tunneling. Infoblox Threat Insight employs streaming analytics to study DNS statistics and create algorithms to detect and mitigate DNS tunneling traffic by analyzing DNS queries and responses.
- **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. Various API's assist in the management of the Infoblox device and the network environments.

  The TOE provides the ability to automate the DNS and DHCP management features of the Grid instead of manually provisioning IP addresses and DNS name spaces for network devices and interfaces using the cloud API service.

  The Outbound API framework provides the ability to configure outbound Notifications. The API framework sends object information and conversations to other REST APIs when an event triggers in NIOS.
- **Reporting and Analytics solution**: automates the collection, analysis, and presentation of core network service data to assist administrators in planning and mitigating network outage risks. It provides predefined dashboards and reports that capture information about the activities and performance of core network services. It also provides the ability to create custom dashboards, reports, and alerts.
- **Trusted Updates:** The TOE uses digital signatures to verify updates prior to installation.
- **Self Protection:** The TOE implements self-tests during initial start-up to determine whether the TOE is operating correctly.
- **Secure Grid:** The TOE uses an SSL/TLS VPN to protect communication between itself and other TOE instances when deployed in a grid.
- **HA Configuration:** The TOE provides hardware redundancy for core network services.
- **Trusted Path/Channels:** The TOE provides secure communication channels with administrators; external syslog server; authentication servers; Backup/Restore servers and with the Advisor Service.

## 2.2 TOE Architecture

The TOE consists of physical and virtual appliances, each with NIOS v8.5.2 software. Each hardware appliance is labelled with a series number and is configured as one of the supported model numbers, which can be viewed from within the software. For example, an appliance may be labelled as an "IB-1405" which is the series number of the appliance and the configured model number could be either: IB-1415 and IB-1425.

The TOE does not expose system interfaces (for example, there is no shell access).

Two physical Infoblox NIOS appliances can be linked together to perform as a single virtual appliance in an HA configuration.

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.

Infoblox NIOS 8.5 is supported on the following platforms:

- Network Insight Appliances: ND-805, ND-1405, ND-2205, ND-4005
- Network Insight Virtual Appliances: IB-V4005
- Trinzic Appliances: TE-815, TE-825, TE-1415, TE-1425, TE-2215, TE-2225, TE-4015, TE-4025
- Trinzic Virtual Appliances: IB-V815, IB-V825, IB-V1415, IB-V1425, IB-V2215, IB-V2225, IB-V4015, IB-V4025.
- Trinzic Reporting Appliances: TR-805, TR-1405, TR-2205, TR-4005
- Trinzic Reporting Virtual Appliances: IB-V5005

Each appliance includes the following security relevant interfaces:
**Console Port** Secure CLI, device configuration and maintenance
**MGMT Ethernet Port** for appliance management
**LAN1 Ethernet Port** for connecting a NIOS appliance to the network. The passive node in an HA pair also uses this port to synchronize the database with the active node,
**LAN2 Ethernet Port** that connects a NIOS appliance to the network. The LAN2 port is disabled by default. You can enable the LAN2 port and define its use through the Grid Manager after the initial setup.

Some appliances have additional interfaces:
**Hard Disk Drive -** The Trinzic TE-1405 Series provide one (1) Infoblox hard disk storage device. The Trinzic Reporting TR-1405 appliance, and Network Insight ND-1405 appliance provide two (2) hard disks in a RAID 1 array. The 2205 Series and 4005 Series each have four hot-swappable Infoblox data storage devices configured in a RAID 10 (Redundant Array of Independent Disks) array.
**HA Port -** A 10/100/1000-Mbps gigabit Ethernet port through which the active node in an HA (high availability) pair connects to the network using a VIP (virtual IP) address. HA pair nodes also use their HA ports for VRRP (Virtual Router Redundancy Protocol) advertisements where supported.

In addition to these interfaces, the appliances also differ in terms of whether they are virtualized or hardware; and what their core function is. The appliances are designed for one of the following: specialized Network Insight or Reporting devices; or Trinzic Appliances providing all other functionality.

Otherwise the functionality is the same across all appliances within a series category. The appliances all run the same code and only differ by performance and capacity.

*Table 3: TOE Hardware Models*

| Infoblox Model | CPU | CPU Speed | Memory | Storage |
|---|---|---|---|---|
| TR/ND-805 | IntelCorei36100TE (2.7Ghz Dual) | 2.7GHz | 32GB DDR4 | 1TB single fixed 7200rpm |
| TE-815 | IntelCorei3-6100TE (2.7Ghz Dual) | 1.10GHz | 16gb DDR4 | 1TB single fixed 7200rpm |
| TE-825 | Intel Core i3-6100TE | 3.6 GHz | 32GB | 1TB |
| TR-1405 | IntelXeonE31275v5 (3.6Ghz Quad) | 3.6GHz | 32GB DDR4 | 1.2TB RAID-1 FRU 2@10k |
| ND-1405 | IntelXeonE31275v5 (3.6Ghz Quad) IntelXeonE31275v6 (3.6Ghz Quad) | 3.6GHz | 32GB DDR4 | 1.2TB RAID-1 FRU 2@10k |
| TE-1415 | IntelXeonE31275v5 (3.6Ghz Quad) | 1.2GHz | 32GB DDR4 | 900GB single FRU 10k |
| TE-1425 | Intel Xeon E3-1275 | 3.6 GHz | 32GB | 900GB |
| TR/ND-2205 | IntelXeonE52620v4 (2.1Ghz 8) | performance governor | 64GB DDR4 | 2.4TB RAID-10 FRU 4@10k |
| TE-2215 | IntelXeonE52620v4 (2.1Ghz 8) | powersave governor | 64GB DDR4 | 1.8TB RAID-10 FRU 4@10k |
| TE-2225 | Intel Xeon E5-2620 | 2.1 GHz | 64GB | 1.8TB |
| TR/ND-4005 | IntelXeonE52680v4 (2.4Ghz 14) | performance governor | 128GB DDR4 | 3.6TB RAID-10 FRU 4@10k |
| TE-4015 | Intel Xeon E5-2680 | 2.4 GHz | 64GB | 1.8TB |
| TE-4025 | IntelXeonE52680v4 (2.4Ghz 14) | performance governor | 128GB DDR4 | 1.8TB RAID-10 FRU 4@10k |

The virtual appliances in the TOE run in a virtual machine and will be qualified on a device as specified in Table 4. The CPU, memory, # of drives, etc.. for the virtual appliances are the same as those for the hardware models specified in Table 3. The TOE includes virtual images for VMware, KVM Hypervisor (RHEL) and Nutanix.

Within appliance categories, the virtual appliance and hardware appliance are identical from the operating system up. They differ in hardware device drivers only.

The evaluated configuration consists of the following appliances:

- HA Pair consisting of two Trinzic appliances: a Grid Master Appliance and a Grid Member - 'TE' or 'IB' appliances.
- Two Network Insight Appliances- one probe and one consolidator – 'ND' or 'IB' appliances.
- Threat Insight (streaming analytics) - 'TE' or 'IB' appliance.
- One Reporting and Analytics Grid Member appliance – 'TR' or 'IB' appliance.

## 2.3 TOE Physical Boundaries

The TOE consists of the appliances and NIOS v8.5.2 software. See Table 1 for hardware and virtual appliance models in the TOE. See

Table 3 for hardware appliance model specifications. The resource requirements for the virtual appliances are specified in Table 4.

The TOE is deployed 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 and HA functionality.

*The TOE hardware appliances include the NIOS v8.5.2 software and the hardware listed in*

Table 3.

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, SAML, TACACS+ servers when the TOE is configured to use an external authentication source
- An OCSP Server when X509 certificates are used for 2-factor authentication
- NTP server when the TOE is configured to use an NTP server
- Backup Server
- Source(s) for Advisor Service
- SSHv2 client when accessing the CLI remotely across an Ethernet network
- The GUI can be accessed using the following browsers[3]: 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 Infoblox NIOS on VMware software runs on VMWare ESX/ESXi; KVM Hypervisor (RHEL); and Nutanix AHV platforms. The servers have DAS (Direct Attached Storage), or iSCSI (Internet Small Computer System Interface) or FC (Fibre Channel) SAN (Storage Area Network) attached. The TOE software package for virtual appliances is installed on one of the hosts and then configured as a virtual appliance. The host appliance and VM OS are part of the operational environment and not part of the TOE. The following

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

table lists the required memory, CPU, and disk allocation for each supported Infoblox virtual appliance model.

*Table 4: Resource Requirements for Virtual Appliances*

| NIOS Virtual Appliance | Primary Disk (GB) | # of CPU Cores | Memory Allocation (GB) |
|---|---|---|---|
| ND-V805 | 250 | 2 | 32 |
| IB-V815 | 250 | 2 | 16 |
| IB-V825 | 250 | 2 | 16 |
| ND-V1405 | 250 | 4 | 32 |
| IB-V1415 | 250 | 4 | 32 |
| IB-V1425 | 250 | 4 | 32 |
| ND-V2205 | 250 | 8 | 32 |
| IB-V2215 | 250 | 8 | 64 |
| IB-V2225 | 250 | 8 | 64 |
| IB-V4005 | 250 (+ 1500 GB reporting storage) | 14 | 128 |
| ND-V4005 | 250 | 14 | 128 |
| IB-V4015 | 250 | 14 | 128 |
| IB-V4025 | 250 | 14 | 128 |
| IB-V5005 | User defined | User defined | User defined |

## 2.4 TOE Logical Boundaries

This section summarizes the security functions provided by the TOE:

- Security Audit
- Cryptographic Support
- DNS Traffic Control
- Identification and authentication
- Asset Discovery
- 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 unauthorized deletion and prevents unauthorized 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.

The TOE implements the DNSSEC Protocol for authenticating the source of DNS data and ensuring its integrity. It protects DNS data from certain attacks, such as man-in the middle attacks and cache poisoning.

### 2.4.3 DNS Traffic Control

The TOE analyzes incoming DNS data and applies algorithms to detect security threats. Once security threats are detected, the TOE blacklists the domain, its traffic is blocked, and an SNMP trap is sent. The extensible service includes a whitelist that contains trusted domains on which the TOE allows DNS traffic that carry legitimate DNS tunneling traffic.

The TOE employs DNS RPZs (Response Policy Zones), for allowing reputable sources to dynamically communicate domain name reputation and allows administrators to implement policy controls for DNS lookups. An RPZ feed receives response policies from external sources and also allows administrators to define multiple response policies locally (local RPZs).

### 2.4.4 Identification and 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 Two-factor authentication, Active Directory (AD), LDAP, RADIUS, SAML, 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. The TOE implements a RADIUS client protocol to support authentication with external RADIUS servers.

### 2.4.5 Asset Discovery

The TOE can detect networks and assets and collect data about them utilizing collection methods: SNMP; CLI device querying; ICMP Ping Sweep and Smart Subnet Ping Sweep; TCP Port Scanning; NetBIOS Queries; and vDiscovery. The protocols can be used to discover and catalogue device types: routers, enterprise switches, firewalls and security appliances, load balancers, enterprise printers, wireless

access points, VoIP concentrators, application servers, VRF-based virtual networks, and end hosts. The TOE can be configured to send SNMP and email notifications when it detects particular events.

### 2.4.6 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' and 'Limited-Access Group role with Cloud API permission'. The superuser performs all security functions of the TOE including (but not limited to) managing audit configuration, password and authentication policies, and TOE updates. The Limited-Access Group user only has access to the Cloud API Service.

### 2.4.7 Protection of the TSF

Communications between the TOE instances (The Infoblox Grid) utilize a TLS secured VPN to protect against the disclosure and modification of data exchanged between the TOE appliances. HA configurations provide hardware redundancy and degraded fault tolerance to minimize service outages.

The TOE provides reliable time stamps and can optionally be set to receive clock updates from an NTP server. The TOE 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.

### 2.4.8 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.

### 2.4.9 Trusted Path/Channels

The TOE communicates with authorized remote administrators via a web based GUI that is protected using HTTPS/TLS. Administrators can also use a CLI over SSH.

The TOE uses TLS to protect all communications with external authentication servers, syslog servers, and backup/restore servers.

The TOE uses HTTPS to communicate with the Advisor Service. The Advisor assists TOE administrators in monitoring and maintaining network and security infrastructure based on Common Vulnerabilities and Exposures (CVEs) as well as vendor product lifecycle announcements.

## 2.5 TOE Documentation

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

- *Infoblox NIOS Administrator Guide Release 8.5,* 4/20/2021
- *Infoblox Installation Guide 4005 Series Appliances,* 2021
- *Infoblox Installation Guide 1405 Series Appliances,* 2021
- *Infoblox Installation Guide 2205 Series Appliances,* 2021

- *Infoblox Installation Guide 805 Series Appliances,* 2021
- *Infoblox Installation Guide NIOS™ for VMware,* 4/5/2021
- *Infoblox Installation Guide vNIOS for Nutanix™ AHV,* 2020
- *vNIOS™ Installation Guide for KVM Hypervisor and KVM-based OpenStack,* 4/6/2021
- *Infoblox WAPI 2.11 Documentation,* 2020

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 release notes 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.MALICIOUS_DNS**

DNS queries that originate from external sources may be to a malicious or unauthorized host designed to infiltrate, damage the system, or cause DNS cache poisoning.

**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.UNDETECTED_ASSETS**

Administrators might not be able to detect assets or networks in their security infrastructure allowing undetected vulnerabilities in the system such as equipment failures and obsolete devices or software potentially leading to an insecure system.

**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.DISCOVERY**

The TOE will provide asset discovery methods; advisory services; alerting and reporting capabilities to assist administrators in monitoring and maintaining network and security infrastructure.

**O.DISPLAY_BANNER**

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

**O.HIGH_AVAILABILITY**

The TOE will provide preservation of secure state, hardware and fault tolerance capabilities for core network services.

**O. MALICIOUS_DNS**

The TOE must be able to detect and react to potential data exfiltration tunnels using its threat analytics service and be able to validate a response to DNS query (a DNS record) before returning it to the client.

**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 use the management interfaces to configure the TOE and its network, 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 in 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

Class DTC: DNS Traffic Control

This class is defined specifically for the security functionality provided by the Infoblox TOE that is not defined in CC Part 2. This class of requirements covers the security functions provided by the TOE regarding analyzing the information forwarded to the TOE's interfaces and how the TSF should respond when certain behavior is detected.

**DTC_DTA _EXT**

Family Behavior

This family requires that the TOE provide the ability to analyze network traffic arriving on TOE interfaces to detect violations and to control the behavior of that traffic based on the analysis.

Management: DTC_DTA _EXT.1

Management of the parameters that control/enforce Traffic.

Audit: DTC_DTA _EXT.1

There are no auditable events foreseen

**DTC_DTA_EXT.1 – DNS Traffic Analysis**

Hierarchical to: No other components.

Dependencies: none

**DTC_DTA_EXT.1.1** The TSF shall perform [***Selection: statistical analytics, signature analysis, integrity analysis, streaming analytics***] to [**Assignment: type of traffic**] forwarded to the TOE's interfaces.

**Application Note:** *Statistical analysis involves identifying deviations from normal patterns of behavior. For example, it may involve mean frequencies and measures of variability to identify*

*abnormal or malicious usage. Signature analysis involves the use of patterns corresponding to known attacks or misuse. For example, patterns of System settings and user activity can be compared against a database of known attacks. Integrity analysis involves comparing System settings or user activity at some point in time with those of another point in time to detect differences. Streaming Analytics – Stream processing analyzes and performs actions on real-time data though the use of continuous queries and responses.*

**DTC_DTA_EXT.1.2** The TSF shall apply [***Selection: whitelist rules, blacklist rules***] to the traffic.

**DTC_DTA_EXT.1.3** The TSF shall allow rules to be applied to the traffic interfaces and shall enforce the defined rules: [**Assignment: rules applied to traffic**].

**DTC_RCT_EXT**

Family Behavior

This family requires that the TOE take action and send an alarm notification when a security threat is detected.

Management: DTC_RCT_EXT.1

No management activities foreseen.

Audit: DTC_RCT_EXT.1

There are no auditable events foreseen

**DTC_RCT_EXT.1 –Analyzer React**

**DTC_RCT_EXT.1.1** The TSF shall send an alarm to [**Assignment: alarm destination**] and take [**Assignment: action to take**] when a security threat is detected.

**FAU_STG_EXT.1**

Family Behavior

This family requires that the TOE provide the ability to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC, and to store audit data locally.

Management: FAU_STG_EXT.1

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

a) The TSF shall have the ability to configure the cryptographic functionality.

Audit: FAU_STG_EXT.1

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

a) No audit necessary.

**FAU_STG_EXT.1 – Protected audit event storage**

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FTP_ITC.1 Inter-TSF Trusted Channel

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

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

**FCS_DNSSEC_EXT**

Family Behavior

The component in this family defines the TOE's DNSSEC protocol implementation.

Management: FCS_DNSSEC_EXT.1

There are no management activities foreseen.

Audit: FCS_DNSSEC_EXT.1

There are no auditable events foreseen.

**FCS_DNSSEC_EXT.1 – DNSSEC protocol**

Hierarchical to: No other components.

Dependencies: FCS_COP.1 cryptographic operation

**FCS_DNSSEC_EXT.1.1** The TSF shall implement the DNSSEC protocol that complies with RFC 4033 and [***selection: 4034, 4035, 4956, 4986, 5155, 5702***].

**FCS_DNSSEC_EXT.1.2** The TSF shall be able to use DNSSEC to authenticate the source of DNS responses.

**FCS_DNSSEC_EXT.1.3** The TSF shall be able to use DNSSEC to ensure that DNS responses were not modified during transit.

**FCS_DNSSEC_EXT.1.4** If a DNS response cannot be authenticated, the TSF shall not return the DNS data.

***Application Note:*** *RFC 4034 should be selected when the TOE supports DNS Security Extensions that provide source authentication for the DNS. RFC 4035 should be selected when the TOE supports the new resource records and protocol modifications that add data origin authentication and data integrity to the DNS. RFC 4956 should be selected when the TOE supports DNSSEC Opt-In extensions for secure delegations to unsigned zones. RFC 4986 should be selected when the TOE supports automated methods for updating Trust Anchors. RFC 5155 should be selected when the TOE supports the NSEC3 resource record, which provides authenticated denial of existence. RFC 5702 should be selected when the TOE produces RSA/SHA-256 or RSA/SHA-512 DNSKEY and RRSIG resource records for use in the Domain Name System Security Extensions*.

**FCS_SSHC_EXT**

Family Behavior

The component in this family addresses the ability for a client to use SSH to protect data between the client and a server using the SSH protocol.

Management: FCS_SSHC_EXT.1

There are no management activities foreseen.

Audit: FCS_SSHC_EXT.1

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

Failure of SSH session establishment

**SSH Client Protocol (FCS_SSHC_EXT.1)**

Hierarchical to: No other components.

Dependencies: FCS_CKM.1 Cryptographic key generation

FCS_COP.1 Cryptographic operation.

**FCS_SSHC_EXT.1.1** The TSF shall implement the SSH protocol that complies with RFC(s) [***4251, 4252, 4253, 4254, 4344, 5647, 5656, 6187, 6668, 8332***].

***Application Note:*** *The following mapping is provided as a guide to ST authors to ensure the appropriate RFC selections are made:*

*RFC 4251 – Select if the TOE complies with the SSH protocol architecture.*

*RFC 4252 – Select if the TOE implements the SSH authentication protocol.*

*RFC 4253 – Select if the TOE implements the SSH transport layer protocol.*

*RFC 4254– Select if the TOE implements the SSH the SSH Connection Protocol.*

*RFC 4344 – Select if AES-128-CTR or AES-256-CTR modes are available.*

*RFC 5647 – Select if AEAD_AES_128_GCM or AEAD_AES_256_GCM are available.*

*RFC 5656 – Select if elliptical curve cryptography is available.*

*RFC 6187 – Select if X.509 certificates are available for public key algorithms.*

*RFC 6668 – Select if HMAC-SHA-2 algorithms are available.*

*RFC 8332 – Select if SHA-2 is available with ssh-rsa selection for public key algorithms.*

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

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

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

**FCS_SSHC_EXT.1.5** The TSF shall ensure that the SSH public-key based authentication implementation uses [**selection: *ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, x509v3-ssh-rsa, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521, x509v3-rsa2048-sha256***] as its public key algorithm(s) and rejects all other public key algorithms.

**FCS_SSHC_EXT.1.6** The TSF shall ensure that the SSH transport implementation uses [***selection: hmac-sha1, hmac-sha1-etm@openssh.com, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512, AEAD_AES_128_GCM, AEAD_AES_256_GCM, implicit***] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

**FCS_SSHC_EXT.1.7** The TSF shall ensure that [***selection: diffie-hellman-group14-sha1, diffie-hellman-group15-sha512, ecdh-sha2-nistp256***] and [***selection: diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, ecdh-sha2-nistp384, ecdh-sha2-nistp521,***

***no other methods***] are the only allowed key exchange methods used for the SSH protocol.

**FCS_SSHC_EXT.1.8** The TSF shall ensure that within SSH connections, each encryption key is used to protect no more than one gigabyte of data. After the threshold is reached, a rekey needs to be performed.

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

**FCS_SSHS_EXT**

Family Behavior

The component in this family addresses the ability for a server to offer SSH to protect data between a client and the server using the SSH protocol.

Management: FCS_SSHS_EXT.1

There are no management activities foreseen.

Audit: FCS_SSHS_EXT.1

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

Failure of SSH session establishment

**SSH Server Protocol (FCS_SSHS_EXT.1)**

Hierarchical to: No other components.

Dependencies:

FCS_CKM.1 Cryptographic key generation

FCS_COP.1 Cryptographic operation

**FCS_SSHS_EXT.1.1** The TSF shall implement the SSH protocol that complies with RFC(s) [**selection:** ***4251, 4252, 4253, 4254, 4344, 5647, 5656, 6187, 6668, 8332***].

***Application Note:*** *The following mapping is provided as a guide to ST authors to ensure the appropriate RFC selections are made:*

*RFC 4251 – Select if the TOE complies with the SSH protocol architecture.*

*RFC 4252 – Select if the TOE implements the SSH authentication protocol.*

*RFC 4253 – Select if the TOE implements the SSH transport layer protocol.*

*RFC 4254– Select if the TOE implements the SSH the SSH Connection Protocol.*

*RFC 4344 – Select if AES-128-CTR or AES-256-CTR modes are available.*

*RFC 5647 – Select if AEAD_AES_128_GCM or AEAD_AES_256_GCM are available.*

*RFC 5656 – Select if elliptical curve cryptography is available.*

*RFC 6187 – Select if X.509 certificates are available for public key algorithms.*

*RFC 6668 – Select if HMAC-SHA-2 algorithms are available.*

*RFC 8332 – Select if SHA-2 is available with ssh-rsa selection for public key algorithms.*

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

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

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

**FCS_SSHS_EXT.1.5** The TSF shall ensure that the SSH public-key based authentication implementation uses [**selection:** ***ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, x509v3-ssh-rsa, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521, x509v3-rsa2048-sha256***] as its public key algorithm(s) and rejects all other public key algorithms.

**FCS_SSHS_EXT.1.6** The TSF shall ensure that the SSH transport implementation uses [***selection: hmac-sha1, hmac-sha1-etm@openssh.com, hmac-sha1-etm@openssh.com, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512, AEAD_AES_128_GCM, AEAD_AES_256_GCM, implicit***] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

**FCS_SSHS_EXT.1.7** The TSF shall ensure that [***selection: diffie-hellman-group14-sha1, diffie-hellman-group15-sha512, ecdh-sha2-nistp256***] and [***selection: diffie-hellman group exchange sha256; diffie-hellman group exchange sha1; diffie-hellman group 14 sha1; diffie-hellman group 1 sha1, diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, ecdh-sha2-nistp384, ecdh-sha2-nistp521, no other methods***] are the only allowed key exchange methods used for the SSH protocol.

**FCS_SSHS_EXT.1.8** The TSF shall ensure that within SSH connections, each encryption key is used to protect no more than one gigabyte of data. After the threshold is reached, a rekey needs to be performed.

**FCS_TLSC_EXT**

Family Behavior

The component in this family addresses the ability for a client to use TLS to protect data between the client and a server using the TLS protocol.

Management: FCS_TLSC_EXT.1

There are no management activities foreseen.

Audit: FCS_TLSC_EXT.1

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

Failure of TLS session establishment.

**TLS Client Protocol (FCS_TLSC_EXT.1)**

Hierarchical to: No other components.

Dependencies: FCS_CKM.1 Cryptographic key generation

FCS_COP.1 Cryptographic operation

**FCS_TLSC_EXT.1.1** The TSF shall implement [**selection:** ***TLS 1.2 (RFC 5246)***, ***TLS 1.1 (RFC 4346), TLS 1.0 (2246)***] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [**selection:**

- ***TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268***
- ***TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268***
- ***TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268***
- ***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492***
- ***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492***
- ***TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246***
- ***TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246***
- ***TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246***
- ***TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288***
- ***TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288***
- ***TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288***
- ***TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289***

].

**FCS_TLSC_EXT.1.2** The TSF shall verify that the presented identifiers of the following types: [**selection:** ***identifiers defined in RFC 6125***, ***IPv4 address in CN or SAN***, ***IPv6 address in the CN or SAN***, ***IPv4 address in SAN***, ***IPv6 address in the SAN***] are matched to reference identifiers.

**FCS_TLSC_EXT.1.3** When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [**selection:**

- ***Not implement any administrator override mechanism***
- ***require administrator authorization to establish the connection if the TSF fails to [selection: match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented server certificate***

].

**FCS_TLSC_EXT.1.4** The TSF shall [**selection:** ***not present the Supported Elliptic Curves Extension***, ***present the Supported Elliptic Curves Extension with the following NIST curves: [selection: secp256r1, secp384r1, secp521r1] and no other curves***] in the Client Hello.

**FCS_TLSS_EXT**

Family Behavior

The component in this family addresses the ability for a server to use TLS to protect data between a client and the server using the TLS protocol.

Management: FCS_TLSS_EXT.1

There are no management activities foreseen.

Audit: FCS_TLSS_EXT.1

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

Failure of TLS session establishment

**TLS Server Protocol (FCS_TLSS_EXT.1)**

Hierarchical to: No other components.

Dependencies: FCS_CKM.1 Cryptographic key generation

FCS_COP.1 Cryptographic operation

**FCS_TLSS_EXT.1.1** The TSF shall implement [**selection:** ***TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346), TLS 1.0 (2246)***] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [**selection:**

- ***TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268***
- ***TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268***
- ***TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268***
- ***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492***
- ***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492***
- ***TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246***
- ***TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246***
- ***TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246***
- ***TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288***
- ***TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288***
- ***TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288***
- ***TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289***

].

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

**FCS_TLSS_EXT.1.3** The TSF shall [**selection:** ***perform RSA key establishment with key size [selection: 2048 bits, 3072 bits, 4096 bits]; generate EC Diffie-Hellman parameters over NIST***

***curves [selection: secp256r1, secp384r1, secp521r1] and no other curves; generate Diffie-Hellman parameters of size [selection: 2048 bits, 3072 bits]***].

**FIA_RADIUS_EXT**

Family Behavior

The component in this family defines the TOE's RADIUS client protocol implementation.

Management: FIA_RADIUS_EXT.1

There are no management activities foreseen.

Audit: FIA_RADIUS_EXT.1

There are no auditable events foreseen.

**FIA_RADIUS_EXT.1 – RADIUS Client Protocol**

Hierarchical to: No other components.

Dependencies: None

**FIA_ RADIUS _EXT.1.1** The TSF shall implement the RADIUS protocol as specified in RFC 2865 for communication with a RADIUS Server.
**FIA_ RADIUS _EXT.1.2** The TSF shall implement RADIUS encapsulated EAP, as specified in RFC 3579.
**FIA_ RADIUS _EXT.1.3** The RADIUS extension for EAP (RFC 2869) shall support the use of EAP-TLS for authentication as specified in RFC 5216.

**FIA_X509_EXT**

Family Behavior

This family defines the behavior, management, and use of X.509 certificates for functions to be performed by the TSF. Components in this family require validation of certificates according to a specified set of rules, use of certificates for authentication for protocols and integrity verification, and the generation of certificate requests.

Management: FIA_X509_EXT.1, FIA_X509_EXT.2

There are no management activities foreseen.

Audit: FIA_X509_EXT.1, FIA_X509_EXT.2

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

Unsuccessful attempt to validate a certificate.

**X.509 certificate validation (FIA_X509_EXT.1)**

Hierarchical to: No other components.

Dependencies: FIA_X509_EXT.2 X.509 Certificate Authentication

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

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.

- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [**selection:** ***the Online Certificate Status Protocol (OCSP) as specified in RFC 2560, a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3, Certificate Revocation List (CRL) as specified in RFC 5759 Section 5***].
- The TSF shall validate the extendedKeyUsage field according to the following rules: [**selection:**
  - ***Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.***
  - ***Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.***
  - ***Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.***
  - ***OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field***.]

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

**FIA_X509_EXT.2 – X.509 certificate authentication**

Hierarchical to: No other components.

Dependencies: FIA_X509_EXT.1 X.509 Certificate Validation

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

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

**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 Cryptographic operation

**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]**.

Class SAD: Asset Discovery

This class is defined specifically for the security functionality provided by the Infoblox TOE that is not defined in CC Part 2. This class of requirements covers the security functions provided by the TOE regarding system data collection, review and alerting notifications.

**SAD_ARP_EXT:**

Family Behavior

This family requires that the TOE be able to trigger an alert and send notifications when certain conditions are detected.

Management: SAD_ARP_EXT.1

There are no management activities foreseen.

Audit: SAD_ARP_EXT.1

There are no auditable events foreseen.

**SAD_ARP_EXT.1 – Alert Definition and Reaction**

Hierarchical to: No other components.

Dependencies: SAD_SDC_EXT.1.

**SAD_ARP_EXT.1.1** The TSF **S**hall be able to trigger an alert when [**assignment: set of conditions]** are met.

**SAD_ARP_EXT.1.2** The TSF shall be able to send a notification to [**assignment: alert destination**] when an alert is triggered.

**SAD_SDC_EXT:**

Family Behavior

This family requires that the TOE be able to collect certain types of data from the TOE and IT systems. It also defines the methods of obtaining that data.

Management: FPT_TST_EXT.1

There are no management activities foreseen.

Audit: FPT_TST_EXT.1

There are no auditable events foreseen.

**SAD_SDC_EXT.1 – System Data Collection**

Hierarchical to: No other components.

Dependencies: None

**SAD_SDC_EXT.1.1** The TSF shall be able to collect System and Event data from the TOE and from external IT systems using [***selection: SNMP, syslog, database schema, XML, CLI device querying, ICMP Ping Sweep and Smart Subnet Ping Sweep, TCP Port Scanning, NetBIOS Queries, [assignment: other method of collection]***].

**SAD_SDC_EXT.1.2** The TSF shall be able to collect the following types of System and Event data [***selection: Equipment Failure; Processing and Software Failure; Threshold Crossing; Object State Change; and Process Started and Stopped***].

| Discovery Type | Data Returned |
|---|---|
| Smart IPv4 Subnet Ping Sweep | IP address<br>MAC address |
| Complete Ping Sweep | IP address<br>MAC address |
| NetBios | IP address<br>MAC address<br>OS<br>NetBIOS name |

| | |
|---|---|
| TCP | IP address<br>MAC address<br>OS |
| Port Scanning/Profile Device | Open and Closed TCP ports<br>IP Address |
| SNMPv1/v2<br>SNMPv3 | Open and Closed TCP ports<br>IP Address<br>System Description<br>System Up Time<br>Routing Neighbors<br>Routing and Forwarding Tables<br>ARP tables<br>SNMP credentials |
| CLI (Device Command-Line by SSH) | Similar data set to SNMP (see above) |
| vDiscovery | IP address<br>MAC address<br>OS<br>Discovered name<br>Virtual entity type<br>Virtual entity name<br>Virtual cluster<br>Virtual datacenter<br>Virtual switch<br>Virtual host<br>Virtual host adapter |

**Application Note:** *The ST author should choose the discovery types supported by the TOE for the first selection and include only the corresponding rows in the Table.*

**SAD_SDC_EXT.1.3** The TSF shall be able to collect the additional audit record content in Column 2 Data in the table above for the selected event types.

**SAD_SDR_EXT:**

Family Behavior

This family requires that the TOE provide the ability to review the collected system data to only authorized user(s).

Management: FPT_TST_EXT.1

There are no management activities foreseen.

Audit: FPT_TST_EXT.1

There are no auditable events foreseen.

**SAD_SDR_EXT.1 – System Data Review**

Hierarchical to: No other components.

Dependencies: SAD_SDC_EXT.1.

**SAD_SDR_EXT.1.1** The TSF shall provide **[assignment: authorized user]** with the capability to read **[assignment: system data that the user can read]** from the system data records**.**

**SAD_SDR_EXT.1.2** The TSF shall provide the system data in a manner suitable for the user to interpret the information.

**SAD_SDR_EXT.2 – Restricted System Data Review**

Hierarchical to: No other components.

Dependencies: SAD_SDR_EXT.1.

**SAD_SDR_EXT.2.1** The TSF shall prohibit all users read access to the system data records, except those users that have been granted explicit read-access**.**

### 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.

*Table 5: TOE Security Functional Components*

| 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 |

| Requirement Class | Requirement Component |
|---|---|
| | FAU_STG_EXT.1: Protected audit event storage |
| | FAU_STG.4: Prevention of audit data loss |
| **FCS: Cryptographic support** | FCS_CKM.1: Cryptographic key generation |
| | FCS_CKM.4: Cryptographic key destruction |
| | FCS_COP.1: Cryptographic operation |
| | FCS_DNSSEC_EXT.1: DNSSEC Protocol |
| | FCS_SSHC_EXT.1: SSH client protocol |
| | FCS_SSHS_EXT.1: SSH server protocol |
| | FCS_TLSC_EXT.1: TLS client protocol |
| | FCS_TLSS_EXT.1: TLS server protocol |
| **DTC: DNS Traffic Control** | DTC_DTA_EXT.1: DNS Traffic Analysis |
| | DTC_RCT_EXT.1: Analyzer React |
| **FIA: Identification and authentication** | FIA_RADIUS_EXT.1: RADIUS Client protocol |
| | 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 |
| | FIA_X509_EXT.1: X.509 certificate validation |
| | FIA_X509_EXT.2: X.509 certificate authentication |
| **FMT: Security management** | FMT_MOF.1(1): Management of security functions behaviour |
| | FMT_MOF.1(2): 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_FLS.1: Failure with preservation of secure state |
| | 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 |
| **FRU: Resource utilisation** | FRU_FLT.1: Degraded Fault Tolerance |
| **FTA: TOE Access** | FTA_SSL.3: TSF-initiated termination |

| Requirement Class | Requirement Component |
|---|---|
| | 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 |
| **SAD: Asset Discovery** | SAD_ARP_EXT.1: Alert Definition and Reaction |
| | SAD_SDC_EXT.1: System Data Collection |
| | SAD_SDR_EXT.1: System Data Review |
| | SAD_SDR_EXT.2: Restricted System Data Review |

### 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 6**].

**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 6**].

*Table 6: 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_EXT.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 |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FCS_COP.1(5) | None | None |
| FCS_DNSSEC_EXT.1 | None | None |
| FCS_SSHC_EXT.1 | None | None |
| FCS_SSHS_EXT.1 | Failure to establish an SSH session. | Reason for failure. |
| FCS_TLSC_EXT.1 | Failure to establish a TLS session. | Reason for failure. |
| FCS_TLSS_EXT.1 | Failure to establish a TLS session. | Reason for failure. |
| FIA_RADIUS_EXT.1 | None | None |
| FIA_SOS.1 | None | None |
| FIA_X509_EXT.1 | Unsuccessful attempt to validate a certificate. | Reason for failure of certificate validation. |
| FIA_X509_EXT.2 | 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(1) | None | None |
| FMT_MOF.1(2) | 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<br>(success or failure) | None |
| FTA_SSL.3 | The termination of a remote session by the session locking mechanism. | None |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| 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_EXT.1 – Protected audit event storage**

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

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

**FAU_STG.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, AES operating CTR**] 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,**
- **CTR 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**].

**FCS_DNSSEC_EXT.1 – DNSSEC protocol**

**FCS_DNSSEC_EXT.1.1** The TSF shall implement the DNSSEC protocol that complies with RFC 4033 and [***4034, 4035, 4956, 4986, 5155, 5702***].

**FCS_DNSSEC_EXT.1.2** The TSF shall be able to use DNSSEC to authenticate the source of DNS responses.

**FCS_DNSSEC_EXT.1.3** The TSF shall be able to use DNSSEC to ensure that DNS responses were not modified during transit.

**FCS_DNSSEC_EXT.1.4** If a DNS response cannot be authenticated, the TSF shall not return the DNS data.

**FCS_SSHC_EXT.1 – SSH client protocol**

**FCS_SSHC_EXT.1.1** The TSF shall implement the SSH protocol that complies with RFC(s) [***4251, 4252, 4253, 4254, 4344, 5656***].

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

**FCS_SSHC_EXT.1.3** The TSF shall ensure that, as described in RFC 4253, packets greater than [**(256 * 1024)**] bytes in an SSH transport connection are dropped.

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

**FCS_SSHC_EXT.1.5** The TSF shall ensure that the SSH public-key based authentication implementation uses [***ssh-rsa***] as its public key algorithm(s) and rejects all other public key algorithms.

**FCS_SSHC_EXT.1.6** The TSF shall ensure that the SSH transport implementation uses [***hmac-sha1, hmac-sha1-etm@openssh.com***] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

**FCS_SSHC_EXT.1.7** The TSF shall ensure that [***diffie-hellman-group14-sha1, ecdh-sha2-nistp256***] and [***ecdh-sha2-nistp384, ecdh-sha2-nistp521***] are the only allowed key exchange methods used for the SSH protocol.

**FCS_SSHC_EXT.1.8** The TSF shall ensure that within SSH connections, each encryption key is used to protect no more than one gigabyte of data. After the threshold is reached, a rekey needs to be performed.

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

***Application Note:*** *The TOE acts as an SSH Client when communicating with the Backup/Restore External Server*

**FCS_SSHS_EXT.1 – SSH server protocol**

**FCS_SSHS_EXT.1.1** The TSF shall implement the SSH protocol that complies with RFC(s) [***4251, 4252, 4253, 4254, 4344, 5656***].

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

**FCS_SSHS_EXT.1.3** The TSF shall ensure that, as described in RFC 4253, packets greater than [**(256 * 1024)**] bytes in an SSH transport connection are dropped.

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

**FCS_SSHS_EXT.1.5** The TSF shall ensure that the SSH public-key based authentication implementation uses [***ssh-rsa***] as its public key algorithm(s) and rejects all other public key algorithms.

**FCS_SSHS_EXT.1.6** The TSF shall ensure that the SSH transport implementation uses [***hmac-sha1, hmac-sha1-etm@openssh.com***] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

**FCS_SSHS_EXT.1.7** The TSF shall ensure that [***diffie-hellman-group14-sha1, ecdh-sha2-nistp256***] and [***ecdh-sha2-nistp384, ecdh-sha2-nistp521***] are the only allowed key exchange methods used for the SSH protocol.

**FCS_SSHS_EXT.1.8** The TSF shall ensure that within SSH connections, each encryption key is used to protect no more than one gigabyte of data. After the threshold is reached, a rekey needs to be performed.

***Application Note:*** *The TOE acts as an SSH Server when Administrators connect to the CLI.*

**FCS_TLSC_EXT.1 – TLS client protocol**

**FCS_TLSC_EXT.1.1** The TSF shall implement [**selection: *TLS 1.2 (RFC 5246)*, *TLS 1.1 (RFC 4346), TLS 1.0 (2246)***] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [**selection:**

- ***TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268***
- ***TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268***
- ***TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268***

].

**FCS_TLSC_EXT.1.2** The TSF shall verify that the presented identifiers of the following types: [***identifiers defined in RFC 6125*, *IPv4 address in SAN*, *IPv6 address in the SAN***] are matched to reference identifiers.

**FCS_TLSC_EXT.1.3** When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [

- ***Not implement any administrator override mechanism***

].

**FCS_TLSC_EXT.1.4** The TSF shall [***present the Supported Elliptic Curves Extension with the following NIST curves: [secp256r1, secp384r1, secp521r1] and no other curves***] in the Client Hello.

**Application Note:** The TOE is TLS client in communications with the syslog server and external authentication servers. REST API also acts as a TLS Client.

**FCS_TLSS_EXT.1 – TLS server protocol**

**FCS_TLSS_EXT.1.1** The TSF shall implement [***TLS 1.2 (RFC 5246)*, *TLS 1.1 (RFC 4346), TLS 1.0 (2246)***] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

- ***TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268***
- ***TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268***
- ***TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268***

].

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

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

**Application Note:** The TOE is TLS server when serving up the WEB UI and PAPI/WAPI to the administrators.

### 5.2.3 Identification and Authentication (FIA)

**FIA_RADIUS_EXT.1 – RADIUS Client Protocol**

**FIA_ RADIUS _EXT.1.1** The TSF shall implement the RADIUS protocol as specified in RFC 2865 for communication with a RADIUS Server.

**FIA_ RADIUS _EXT.1.2** The TSF shall implement RADIUS encapsulated EAP, as specified in RFC 3579.

**FIA_ RADIUS _EXT.1.3** The RADIUS extension for EAP (RFC 2869) shall support the use of EAP-TLS for authentication as specified in RFC 5216.

**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 be 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_X509_EXT.1 – X.509 certificate validation**

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

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [***the Online Certificate Status Protocol (OCSP) as specified in RFC 2560***].
- The TSF shall validate the extendedKeyUsage field according to the following rules: [
  - ***Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.***
  - ***Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.***
  - ***OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.***]

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

**FIA_X509_EXT.2 – X.509 certificate authentication**

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

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

**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, X509 certificate, two-factor authentication, and support for remote authentication via external AD, LDAP, RADIUS, SAML, 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 Resource utilisation (FRU)

**FRU_FLT.1 – Degraded Fault Tolerance**

**FRU_FLT.1.1** The TSF shall ensure the operation of **[Core Network Services]** when the following failures occur: **[HA Node Hardware or Service Failure].**

### 5.2.5 Security Management (FMT)

**FMT_MOF.1(1) – Management of security functions behaviour**

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

**FMT_MOF.1(2) – Management of security functions behaviour**

**FMT_MOF.1.1(2)** The TSF shall restrict the ability to **[*modify the behaviour of*]** the functions **[Manage IPAM automation]** to **[superuser, Limited-Access Group role with Cloud API permission]**.

**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**
- **Manage IPAM automation**
- **Manage Outbound Notifications**

].

**FMT_SMR.1 – Security roles**

**FMT_SMR.1.1** The TSF shall maintain the roles: [**superuser, Limited-Access Group role with Cloud API permission**].

**FMT_SMR.1.2** The TSF shall be able to associate users with roles.

### 5.2.6 Protection of the TSF (FPT)

**FPT_FLS.1 – Failure with preservation of secure state**

**FPT_FLS.1.1** The TSF shall preserve a secure state when the following types of failures occur: **[HA Node Hardware or Service Failure].**

**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**.**

### 5.2.7 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 between 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.

### 5.2.8 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, LDAP, RADIUS, TACACS+ Servers, Sending Audit Events to Syslog, Backup/Restore Data with Backup Server, connections with Advisor]**.

**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]***].

### 5.2.9 DNS Traffic Control (DTC)

**DTC_DTA_EXT.1 – DNS Traffic Analysis**

**DTC_DTA_EXT.1.1** The TSF shall perform [***streaming analytics***] to [**DNS network traffic**] forwarded to the TOE's interfaces.

**DTC_DTA_EXT.1.2** The TSF shall apply [***whitelist rules, blacklist rules***] to the traffic.

**DTC_DTA_EXT.1.3** The TSF shall allow rules to be applied to the traffic interfaces and shall enforce the defined rules:

[**DNS RPZ Rules:**

- **Passthru Rule**
- **Block (No Such Domain) Rule**
- **Block (No Data) Rule**
- **Substitute (Domain Name) Rule**
- **Substitute (Record) Rule**].

**DTC_RCT_EXT.1 –Analyzer React**

**DTC_RCT_EXT.1.1** The TSF shall send an alarm to [**SNMP**] and take [**action to Blacklist the offending Domain**] when a security threat is detected.

### 5.2.10 Asset Discovery (SAD)

**SAD_ARP_EXT.1 – Alert Definition and Reaction**

**SAD_ARP_EXT.1.1** The TSF Shall be able to trigger an alert when [**Equipment Failure; Processing and Software Failure; Threshold Crossing; Object State Change; and Process Started and Stopped]** are met.

**SAD_ARP_EXT.1.2** The TSF shall be able to send a notification to [**SNMP, Email**] when an alert is triggered.

**SAD_SDC_EXT.1 – System Data Collection**

**SAD_SDC_EXT.1.1** The TSF shall be able to collect System and Event data from the TOE and from external IT systems using [***SNMP, CLI device querying, ICMP Ping Sweep and Smart Subnet Ping Sweep, TCP Port Scanning, NetBIOS Queries, [vDiscovery]***].

**SAD_SDC_EXT.1.2** The TSF shall be able to collect the following types of System and Event data [***Equipment Failure; Processing and Software Failure; Threshold Crossing; Object State Change; and Process Started and Stopped***].

Table 7: Discovery Methods and Data Returned

| Discovery Type | Data Returned |
|---|---|
| Smart IPv4 Subnet Ping Sweep | IP address<br>MAC address |
| Complete Ping Sweep | IP address<br>MAC address |
| NetBios | IP address<br>MAC address<br>OS<br>NetBIOS name |
| TCP | IP address<br>MAC address<br>OS |
| Port Scanning/Profile Device | Open and Closed TCP ports<br>IP Address |
| SNMPv1/v2<br>SNMPv3 | Open and Closed TCP<br>ports<br>IP Address<br>System Description<br>System Up Time |

| | |
|---|---|
| | Routing Neighbors<br>Routing and Forwarding Tables<br>ARP tables<br>SNMP credentials |
| CLI (Device Command-Line by SSH) | Similar data set to SNMP (see above) |
| vDiscovery | IP address<br>MAC address<br>OS<br>Discovered name<br>Virtual entity type<br>Virtual entity name<br>Virtual cluster<br>Virtual datacenter<br>Virtual switch<br>Virtual host<br>Virtual host adapter |

**SAD_SDC_EXT.1.3** The TSF shall be able to collect the additional audit record content in Column 2 Data in the table above for the selected event types.

**SAD_SDR_EXT.1 – System Data Review**

**SAD_SDR_EXT.1.1** The TSF shall provide **[superusers]** with the capability to read **[all system data]** from the system data records**.**

**SAD_SDR_EXT.1.2** The TSF shall provide the system data in a manner suitable for the user to interpret the information.

**SAD_SDR_EXT.2 – Restricted System Data Review**

**SAD_SDR_EXT.2.1** The TSF shall prohibit all users read access to the system data records, except those users that have been granted explicit read-access**.**

## 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.

*Table 8: 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 |

| Requirement Class | Requirement Component |
|---|---|
| **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 |
| | 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 |

### 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
- DNS Traffic Control
- Identification and authentication
- Asset Discovery
- Security Management
- Protection of the TSF
- TOE Access
- Trusted Path/Channels

## 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.

*Table 9: Auditable Events*

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FCS_SSHC_EXT.1 | Failure to establish an SSH session. | Reason for failure. |
| FCS_SSHS_EXT.1 | Failure to establish an SSH session. | Reason for failure. |
| FCS_TLSC_EXT.1 | Failure to establish a TLS session. | Reason for failure. |
| FCS_TLSS_EXT.1 | Failure to establish a TLS session. | Reason for failure. |
| FIA_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_X509_EXT.1 | Unsuccessful attempt to validate a certificate. | Reason for failure of certificate validation. |
| FMT_SMF.1 | Use of the management functions. | Identity of the administrator performing these functions. |
| FPT_FLS.1 | Hardware and Service Failures | 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). |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| 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 |
| 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. |

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 9. 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 Audit log file is 100 MB and the maximum size of the local syslog file is 300 MB. When either log file reaches its maximum size, the NIOS appliance automatically writes the file into a new file by adding a .0 extension to the first file and incrementing subsequent file extensions by 1. Files are compressed during the rotation process, adding a .gz extension following the numerical increment (file.#.gz). The first file starts with .0 and subsequent file extensions are incremented by one until it reaches nine. For example, the current log file moves to file.0.gz, the previous file.0.gz moves to file.1.gz, and so on through file.9.gz. A maximum of 10 log files (0-9) are kept. When the eleventh file is started, the last oldest log file (file.9.gz) is deleted, and subsequent files are renumbered accordingly. The rotation function essentially results in newer audit records overwriting the oldest audit records.

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_EXT.1: The TOE transmits audit data to a remote syslog server over TLS in real time.
- FAU_STG.1: 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.
- 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; for DNSSEC; 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 a TLS VPN.
- **Remote Administration.** Remote administrators configure the TOE via a web based GUI that is protected using HTTPS or via CLI protected using SSH.
- **Perl Application Programming Interface.** The TOE provides a Perl API to assist integration of the Infoblox device into network environments. The API is protected using HTTPS. The Perl API provides interfaces to DHCP, DNS, Grid and IPAM services.
- **REST Application Programming Interface.** The TOE provides a REST API for Grid management. It uses HTTP methods for operations and supports input and output in JSON and XML; and TLS as the transport mechanism.
- **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/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. 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/LDAP authentication servers. TLS provides protection of the communications pathways from disclosure and from undetected modification. Connections with the Advisor Server use HTTPS.

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
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA

The TLS_DHE_RSA_WITH_AES_256_CBC_SHA ciphersuite is used for Secure Grid Communications.

The TOE uses OpenSSL cryptography to implement all cryptographic functions.

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

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA

When the TOE is acting as a TLS client, the TOE verifies the presented identifiers of the following types:

- Identifiers defined in RFC 6125
    - Common Name: Specify the domain name of the NIOS appliance. The FQDN (fully qualified domain name) of the appliance can be used.
    - Organization: The company name.
    - Organizational Unit: The department name
- Subject Alternative Name: The following entries can be included as SAN extension: DNS, Email, IP Address (IPv4, IPv6), and URI.

The presented identifers are matched to reference identifiers. When establishing a trusted channel, by default the TOE will not establish a trusted channel if the server certificate is invalid and does not implement any override mechanisms. The TOE presents the Supported Elliptic Curves Extension with the following NIST curves: secp256r1, secp384r1, and secp521r1 in the Client Hello. The TOE is a TLS client in communications with the syslog server and external authentication servers. REST API also acts as a TLS Client.

The TOE implements the SSHv2 protocol for remote administrators accessing the CLI and for communicating with the external Backup/Restore Server, supporting the following algorithms.

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

The TOE's SSH implementation complies with RFC(s) ***4251, 4252, 4253, 4254, 4344, 5656*** and supports password-based, authentication methods. As described in RFC 4253, packets greater than (256 * 1024) bytes in an SSH transport connection are dropped. The TOE ensures that within SSH connections, each encryption key is used to protect no more than one gigabyte of data. After the threshold is reached, a rekey is performed.

The CAVP certificate numbers are listed in the Table below.

*Table 10: OpenSSL FIPS Object Module Certificates*

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

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.

*Table 11: Secret Keys, Private Keys, and CSPs*

| 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. |
| Integrity Test Private Key RSA/SHA256 – 4096-bit | Stored in memory | Overwritten with zeroes when no longer in use. |
| X509 CA Certificate:<br>ECDSA:<br>P-256 (256 bits),<br>P-384 (384 bits),<br>P-521 (521 bits)<br><br>RSA:<br>2048 bits,<br>3072 bits,<br>4096 bits | Encrypted database file | Overwritten when no longer in use; three times with a random pattern, and once with zeroes. |
| X.509 HTTPS Certificate Private Key: RSA 2049 bits, RSA 4096 bits | Stored in memory | Overwritten with zeroes when no longer in use. |
| SSHv2 Private Keys: RSA 2048 bits; ECDH 256, 384, 521 bits, DH 2048 | Stored in memory | Overwritten with zeroes when no longer in use. |
| 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. |
| DNSSEC KSK, ZSK Private Keys: RSA 2048, 3072, 4096 | 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 TOE implements the **DNSSEC protocol** that complies with RFC 4033, 4034, 4035, 4956, 4986, 5155, and 5702. DNSSEC (DNS Security Extensions) provides mechanisms for authenticating the source of DNS data and ensuring its integrity. It protects DNS data from certain attacks, such as man-in the middle attacks and cache poisoning. A man-in-the middle attack occurs when an attacker intercepts responses to queries and inserts false records. Cache poisoning can occur when a client accepts maliciously created data.

The TOE uses DNSSEC to authenticate the source of DNS responses and to ensure the integrity of DNS responses. If a DNS response cannot be authenticated, the TOE returns a SERVFAIL response and does not return the DNS data.

Name servers in a Grid can be configured to support DNSSEC. Grid Masters can be configured as the primary server for a signed zone and the Grid members as secondary servers.

The TOE uses public key cryptography in DNSSEC to authenticate the source of DNS responses and to ensure that DNS responses were not modified during transit ensuring its integrity. The primary name server (the TOE) of a zone generates two public/private key pairs. It signs each data set in the zone by running it through a one-way hash, and then encrypting the hash value with the private key. The public key is stored in an RR (Resource Record) type as defined in RFC 4034, and which is called the DNSKEY RR. Resolvers use the DNSKEY record to decrypt the hash value. If the hash values match, then the resolver is assured of the authenticity of the message.

The Grid Master can be enabled to sign zones and manage the DNSSEC keys. When NIOS digitally signs a zone, it generates two key pairs, a zone-signing key (ZSK) pair and a key-signing key (KSK) pair. NIOS then uses the private key of the ZSK pair to sign each RRset in a zone. An RRset is a group of resource records that are of the same owner, class, and type. It stores the public key of the ZSK pair in a DNSKEY record. The name server then uses the private key of the KSK pair to sign all DNSKEY records, including its own, and stores the corresponding public key in another DNSKEY record. As a result, a zone typically has two DNSKEY records; a DNSKEY record that holds the public key of the ZSK pair, and another DNSKEY record for the public key of the KSK pair. If a zone does not have a chain of trust from a parent zone, security aware resolvers can configure the KSK as a trust anchor; that is, the starting point from which it can build a chain of trust from that zone to its child zones.

The protocol extensions are implemented as defined in the RFCs with the first four fields specifying the domain name of the zone that owns the key, the resource record Time To Live (TTL), class, and RR type. The succeeding fields are: Flag fields, Protocol (always 3 for DNSSEC), and Algorithm (RSA with SHA1/SHA-256 or SHA-512 hashing algorithms).

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.

- FCS_DNSSEC_EXT.1: The TOE implements the DNSSEC protocol that complies with RFC 4033, 4034, 4035, 4956, 4986, 5155, 5702. The TOE is capable of using DNSSEC to authenticate the source of DNS responses and ensuring that DNS responses were not modified during transit. If a DNS response cannot be authenticated, the TOE returns a SERVFAIL response and does not return the DNS data.
- FCS_SSHC_EXT.1: The TOE implements the SSH client protocol for communications with the Backup Server.
- FCS_SSHS_EXT.1: The TOE implements SSHv2 and acts as an SSH Server for communications with administrators accessing the CLI.
- FCS_TLSC_EXT.1: The TOE implements TLS versions 1.0, 1.1 and 1.2 for the TLS Client connections
- FCS_TLSS_EXT.1: The TOE implements TLS versions 1.0, 1.1 and 1.2 for the TLS Server connections.

## 6.3 DNS Traffic Control

The Infoblox infrastructure security features protect external DNS from cyber DNS attacks and internal DNS from infrastructure attacks, data exfiltration, threats and malware.

**Infoblox DNS Firewall employs DNS RPZs** (Response Policy Zones), a technology developed by ISC (Internet System Consortium) for allowing reputable sources to dynamically communicate domain name reputation so you can implement policy controls for DNS lookups.

On an Infoblox appliance, RPZs and RPZ rules are configured and defined to block DNS resolution for malicious or unauthorized hostnames, or redirect clients to a walled garden by substituting responses. Administrators can assign action to RPZ rules. For example, abc.com can have an action of pass thru or substitute (domain) with the domain xyz.com. A Grid member can act as a lead secondary that receives RPZ updates from external reputation sources and redistributes the updates to other Grid members. Infoblox DNS Firewall supports both IPv4 and IPv6 networks.

An Infoblox Grid performs RPZ actions for queries that originate from external sources. The name server recursive cache on an RPZ enabled Grid member uses the address of the client from which the query originates to identify if the query is generated from an external source or an internal Grid. If the query originates from a Grid Master or a Grid member that has RPZ license installed, RPZ actions are automatically bypassed for those queries. For RPZ, Infoblox uses the ACL infoblox-deny-rpz, which contains a list of addresses for bypassing RPZ actions. The infoblox-deny-rpz list excludes Grid members that do not have an RPZ license. Note that RPZ action is performed only once for a single recursion.

The Infoblox DNS server receives RPZ updates, which include blacklisted hostnames and responses, from a reputation data server through a DNS zone transfer. The appliance then blocks or redirects queries and responses based on the imported RPZ rules. When DNSSEC is enabled on the Infoblox DNS server, the TOE does not redirect DNS clients that request DNSSEC data. The reporting server can then generate the DNS Top RPZ Hits report that details the top DNS clients that have received redirected responses through RPZs.

There are three types of RPZs:

**Local RPZ** – A local RPZ is a zone that allows administrators to define multiple response policies locally. A local RPZ is associated with at least one primary name server. Responses sent are based on the defined rules.

Each RPZ can have various rules associated with it. The response of a query is modified if it matches any of the RPZ rules. The responses are first matched with the RPZ rules, and if there is a match, the rule defined at the RPZ level override is used. When creating a new RPZ zone, the zone is associated with a threat severity level. Override Policies determine possible behavior such as Log Only, Block, Pass thru, Substitute or None. Each RPZ is configured with a Threat Severity Level: Critical, Major, Warning, or Informational. Each of these levels is represented by a number in the syslog (8 being Critical and 4 being Informational). A local RPZ can be disabled.

The RPZ syslog messages provide information about threat severity level of an RPZ zone associated with the matched RPZ rule. Threat details, are provided in the syslog messages. Multiple local RPZs and rules can be defined for a local RPZ. Override depends on the order of the zones. The zones on top will override the zones below. The order of the RPZs can be changed.

The rules are classified as follows:

- Passthru Rule
- Block (No Such Domain) Rule
- Block (No Data) Rule
- Substitute (Domain Name) Rule
- Substitute (Record) Rule.

Passthru rules can be defined when the actual responses of the queries do not need to be modified. The response received for a query is not modified, if there is a matching passthru rule and the actual response is forwarded to the user or client.

Rules can be defined to block certain domain names, IP addresses or networks, or client IP addresses or networks. With this option to block a domain name, the query name is matched with the RPZ rule. If the query name matches the RPZ rule, the DNS client receives a DNS response that indicates the domain does not exist.

With this option to block an IP address or network, the DNS resource A and AAAA records are matched with the RPZ rule. If the records match an RPZ rule, the DNS client receives a DNS response that indicates the domain does not exist. With the option to block a specific client IP address or network, the IP address or network of a client querying the DNS server is matched with the RPZ rule. If the IP address or the network of the client matches the RPZ rule, the DNS client receives a DNS response that indicates the domain does not exist.

When Managing Block (No Data) Rules, rules are defined to block certain domain names, IP addresses or networks, or client IP addresses or networks. With this option to block a domain name, the query name is matched with the RPZ rule. If the query name matches the RPZ rule, the DNS client receives a DNS response that indicates there is no data for the requested record type.

To block an IP address or network using this option, the A and AAAA records are matched with the RPZ rules. If the records match an RPZ rule, the DNS client receives a DNS response that indicates there is no data for the requested record type.

With this option to block a specific client IP address or network, the IP address or network of a client querying the DNS server is matched with the RPZ rule. If the IP address or the network of the client

matches the RPZ rule, the DNS client receives a DNS response that indicates there is no data for the requested record type.

An administrator can define an alternate IP address or domain name to redirect an IP address or domain name that is malicious or unauthorized. When the response to the client query matches an RPZ rule, the actual domain name or IP address is substituted with the alternative domain name or IP address. The client will receive the substituted value instead of the actual response.

An administrator can define a substitute record for a domain name that is considered malicious. Substitutes can be defined for records or IP addresses in a zone.

**RPZ Feed** – An RPZ feed receives response policies from external sources. DNS clients receive responses based on the imported rules from a reputable source, such as a commercial RPZ provider. Unlike Local RPZ, administrators cannot define the rules for an RPZ feed. An RPZ feed uses rules defined by external servers. The rules for RPZs from feeds are the same as those described above: Passthru, block etc…The RPZ feed tab displays a dialog box that provides various options to export the rules of the configured external servers in .CSV format.

The Infoblox Threat Intelligence Feed source is a threat feed subscription for RPZ updates that offer protection against malicious hostnames. You can configure the Threat Intelligence Feed and receive reputation RPZ updates on a regular basis. An RPZ feed receives response policies from the Infoblox in-house threat intelligence team, which produces reputation RPZ data and transfers the data to Grid name servers through zone transfers with or without a Secret Key Transaction Authentication for DNS (TSIG) key. To ensure proper authentication and integrity of the RPZ feed zone transfers, using a TSIG key is recommended.

The Infoblox Threat Insight (also referred to as Threat Analytics in Grid Manager), protects mission-critical DNS infrastructure from data exfiltration through DNS tunneling. Infoblox Threat Insight employs streaming analytics to study DNS statistics and create algorithms to detect and mitigate DNS tunneling traffic by analyzing DNS queries and responses.

**Infoblox Threat Insight** identifies data exfiltration tunnels that bypass typical firewall systems. Some popular tunneling tools are OyzmanDNS, SplitBrain, Iodine, DNS2TCP, TCP-Over-DNS, and others. These types of DNS threats are identified as having high activities by using the TXT records in DNS queries. Infoblox Threat Insight also identifies tunnels that are used for C&C (Command-and-Control). These threats typically do not exhibit high activities or payloads. In general, NXDOMAIN responses fall into this category of threats.

The threat analytics service analyzes incoming DNS data and applies the algorithms to detect security threats that have the same or similar behavior as the known data. Once security threats are detected, NIOS blacklists the domains and transfers them to the designated mitigation RPZ (Response Policy Zone); traffic from the offending domains is blocked; and no DNS lookups are allowed for these domains from NIOS members on which RPZ are assigned to them. The appliance also sends an SNMP trap each time it detects a new blacklisted domain.

Infoblox Threat Insight includes a whitelist that contains trusted domains on which NIOS allows DNS traffic. These are known good domains that carry legitimate DNS tunneling traffic such as Avast, Sophos, McAfee, Boingo, Barracuda, and others. The whitelist is extensible so new whitelisted domains can be added and rolled out accordingly. Threat Insight ensures all whitelist entries are accurate and curated,

and contain only valid entries. Custom whitelisted domains can be added and blacklisted domains can be moved to the whitelist.

Infoblox Threat Insight comes with a module set and a whitelist set. To receive subsequent module set and whitelist set updates, you can configure the appliance to automatically download and apply the updates or they can be manually uploaded when the appliance displays a banner message notifying about available updates.

The DNS Traffic Analysis function is designed to satisfy the following security functional requirements:

- DTC_DTA_EXT.1: The TOE performs analysis of DNS network traffic forwarded to the TOE's interfaces, and enforces Whitelist/Blacklist rules.
- DTC_RCT_EXT.1: The TOE blacklists offending Domains and sends SNMP notification when a new blacklisted domain is detected.

## 6.4 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, SAML 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).

For access to administrative interfaces using HTTPS, the TOE also supports certificate and two factor authentication using X.509 digital certificates assigned to specific administrators. If an administrator is configured for certificate or two-factor authentication, the TOE must receive a "good" status from an OCSP responder in addition to successfully authenticating the user via password-based logon (if two-factor is configured for the user) according to policy defined for that user in order for the user to gain access. If the TOE receives a certificate status of "revoked" or "unknown" from the OCSP responder or if password-based authentication fails, the authentication fails and the user is not given access. When the TOE cannot establish a connection to determine the validity of a certificate, the TOE does not accept the certificate and authentication fails.

The NIOS appliance supports authentication using the following RADIUS servers: FreeRADIUS, Microsoft, Cisco, and Funk. When NIOS authenticates administrators against RADIUS servers, NIOS acts similarly to a network access server (NAS), which is a RADIUS client that sends authentication and accounting requests to a RADIUS server. RADIUS provides authentication, accounting, and authorization functions to support authentication of administrators.

The TOE supports PAP (Password Authentication Protocol) and CHAP (Challenge Handshake Authentication Protocol) authentication protocols. The TOE uses a shared secret to encrypt and decrypt messages with the RADIUS server. This shared secret is a value that is known only to the NIOS appliance and the RADIUS server. The TOE supports RADIUS accounting which allows an admin to track an administrator's activities during a session. Multiple RADIUS servers are supported.

The TOE's RADIUS Client protocol implementation complies with RFCs 2865, 3579, and 5216. The TOE ensures that EAP is the authentication protocol to be used between the TOE and the RADIUS server; that TLS is the means of mutual authentication to be carried out over EAP; and that other authentication

frameworks are disallowed. The TOE's implementation of RADIUS encapsulated EAP Message Authenticators conform to RFC 3579.

The TOE validates certificates in accordance with the following rules:

• RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.

• The certification path terminates with a trusted CA certificate designated as a trust anchor.

• The certification path is validated by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.

• The TOE validates the revocation status of the certificate using the Online Certificate Status Protocol (OCSP) as specified in RFC 2560.

• The TOE validates the extendedKeyUsage field according to the following rules:

- Server certificates presented for TLS have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
- Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
- OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

Note that the TOE does not use certificates for trusted updates or executable code integrity verification and therefore the code signing purpose need not be checked.

The TOE enforces the following password policy for administrative passwords:

- Passwords can be composed of any combination of upper and lower case letters, numbers, and special characters (that include: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")");
- Minimum password length can be configured by the Authorized Administrator, and support passwords of 15 characters or greater;
- Passwords should 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_RADIUS_EXT.1: The TOE supports RADIUS user authentication by implementing the RADIUS Client protocol complying with standards.
- 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 X.509 certificates, two-factor (password with X.509 certificate), Active Directory, LDAP, RADIUS, SAML, TACACS+ authentication.
- FIA_UAU.7: The TOE provides only obscured feedback to the user while the authentication is in progress.
- FIA_X509_EXT.1: The TOE supports X.509 certificate validation for certificate and two-factor authentication
- FIA_X509_EXT.2: The TOE supports X509 certificate authentication and will not accept a certificate if the TOE cannot establish a connection to determine the validity of a certificate.

## 6.5 Asset Discovery

**Supported Discovery Methods**

The TOE provides the following Discovery methods used to detect assets and collect data about them.

- SNMPv1/v2c device polling
- SNMPv3 device polling
- CLI device querying
- ICMP Ping Sweep and Smart Subnet Ping Sweep
- TCP Port Scanning
- NetBIOS Queries
- vDiscovery

These methods actively scan predefined networks and probe IP addresses. The appliance listens for responses from the IP addresses as proof of activity. The IP discovery scans through the specified network ranges and probes IP addresses (except for the network, broadcast, and multicast address types) in each network, including the /31 and /32 subnets.

The TOE supports discovery of devices and networks through SNMPv1/v2c and through SNMPv3 protocols. Discovery acquires information from standard SNMP MIB object IDs (OIDs) to correctly identify and catalogue devices. The administrator enters or imports lists of SNMP credentials with which the appliances query devices on the network to perform discovery. SNMPv1/v2c discovery requires standard read community strings to be stored on the Grid Master.

Accounts using SNMPv3 use a standard suite of authentication and security protocols. When SNMPv3 is used to collect data from devices supporting the protocol, the administrator can define specific user credentials with combinations of authentication and protocol support, and the unique keys for each protocol. Multiple entries for the same username string are supported which enables checking of similar SNMPv3 credentials that use different authentication and security protocols.

While SNMP is required for all device discovery, CLI data collection to collect information for specific device types is optional. Network Insight enables the use of dynamically created and closed SSH command-line sessions to log in, query, and configure ports using each device's command-line syntax.

Discovery uses different variations of Ping traces to perform higher-performance, brute-force device discovery. ICMP is the last resort when devices do not support SNMP management protocols or an SNMP credential is lacking. The ICMP Smart Ping Sweep option enables brute-force subnet Ping sweeps on IPv4 networks. Subnet ping sweeps are used as a last resort in the discovery process. A subnet ping sweep is performed if the TOE is otherwise unable to identify any network devices in a given subnet. Subnet ping sweeps are performed no more than once per day, and will end the ping sweep on a given subnet once Network Insight discovers a network device and is able to collect data from it. The administrator can configure the timeout value (Ping Sweep Timeout) and the number of attempts (Ping Sweep Attempts).

Complete Ping Sweep differs from the Smart Subnet ping sweep in the following ways:

- The discovery ping sweep runs only against the specified range.
- The sweep runs regardless of the range size.
- The sweep runs regardless of the number of discovered devices within the specified range.

TCP scanning probes each active host on a list of TCP ports using TCP SYN packets. This method detects all active hosts that generate SYN ACK responses to at least one TCP SYN. The discovery can determine the OS on a host by analyzing how the host reacts to the requests on opened and closed ports. It then uses the TCP fingerprints to guess the OS. To obtain a TCP fingerprint, IP discovery provides two scanning techniques, SYN and CONNECT. When you use the SYN technique, the discovery sends a TCP SYN packet to establish a connection on a TCP port. If the port is open, the host replies with a SYN ACK response. The discovery does not close the port connection. The CONNECT technique is a three-way TCP handshake. The discovery starts with the same process as the SYN technique by sending the TCP SYN packet. A response containing a RST flag indicates that the port is closed. If the host replies with a SYN ACK response, discovery sends a RST packet to close the connection. If there is no reply, the port is considered filtered. TCP scanning is a deliberate and accurate discovery method, enabling detection of all active hosts on a network provided that there are no firewalls blocking TCP packet exchanges.

TCP Port Scanning probes the list of TCP ports enabled in the Advanced tab, to determine whether they are open. Some settings for port scanning behavior are configurable, including the choice of a TCP scanning technique. If Profile Device is enabled, the TOE attempts to identify the network device based on the response characteristics of its TCP stack, and uses this information to determine the device type. In the absence of SNMP access, the Profile Device function is usually the only way to identify devices that do not support SNMP. If Profile Device is disabled, devices accessible via SNMP are still correctly identified; all other devices are assigned a device type of Unknown. Profile Device is disabled by default for discovery polling. The Profile Device option uses the editable list of TCP protocol ports from the Grid Discovery Properties –> Polling –>Advanced tab as its profile, and polls each of the ports enabled in that list, using the configured timeout value and the number of polling attempts for each port.

The NetBIOS method queries IP addresses for an existing NetBIOS service. This method detects active hosts by sending NetBIOS queries and listening for NetBIOS replies. It is a fast discovery that focuses on Microsoft hosts or non-Microsoft hosts that run NetBIOS services. NetBIOS discovery returns the following information for each detected host:

- IP address: The IP address of the host.
- MAC address: Listed only if the discovered host is running Microsoft, otherwise blank.
- OS: This value is set to Microsoft for an active host that has a MAC address in the NetBIOS reply.
- NetBIOS name: This value is set to the name returned in the NetBIOS reply.

The vDiscovery method communicates with the vSphere servers to collect discovery data on virtual machine instances. vDiscovery returns the following information for AWS/Azure/OpenStack/VMWare servers:

- IP address
- MAC address
- OS
- Discovered name
- Virtual entity type
- Virtual entity name
- Virtual cluster
- Virtual datacenter
- Virtual switch
- Virtual host
- Virtual host adapter

See *Table 7: Discovery Methods and Data Returned* for a complete list of system data that can be collected.

The TOE collects System and Event data from the TOE and from external IT systems using SNMP for the following types of System and Event categories: Equipment Failure; Processing and Software Failure; Threshold Crossing; Object State Change; and Process Started and Stopped assignment. Network Insight appliances use SNMP and other protocols to discover and catalogue a diverse assortment of device types, including the following: routers, enterprise switches, firewalls and security appliances, load balancers, enterprise printers, wireless access points, VoIP concentrators, application servers, VRF-based virtual networks, and end hosts.

**SNMP Monitoring and Alert Notification**

The TOE can be configured to send SNMP and email notifications when it discovers the following types of events: Equipment, Software, or Process Failures; Threshold Crossing; Object State Change; and Process Started and Stopped.

Examples of each type of event that can trigger a trap/email are:

- Equipment Failure: Primary drive is full; A power supply failure has occurred
- Software and Process Failure: An SSH daemon failure has occurred; An Apache software failure has occurred
- Threshold Crossing: System Memory Usage exceeds the critical threshold value.; CPU / Hard Drive Usage over threshold value
- Object State Change: Service Shutdown/ state changes; Network Interfaces Monitoring (i.e. port link up or down); HA State change; Nodes connected to Grid
- Process Started and Stopped: Httpd Start/Stop and Zone transfer Failed

**Infoblox Advisory Service**

The TOE provides a subscription-based Advisory Service that gathers information regarding released Common Vulnerabilities and Exposures (CVEs) and vendor product lifecycle announcements. The CVEs and other information obtained from Advisory is used by Infoblox Reporting and Analytics to assist administrators in monitoring and maintaining network and security infrastructure.

**Infoblox Reporting and Analytics**

The TOE provides a Reporting function: the Infoblox Reporting and Analytics solution that automates the collection, analysis, and presentation of core network service data to assist administrators in planning and mitigating network outage risks. It provides predefined dashboards and reports that capture information about the activities and performance of core network services. It also provides an enhanced reporting interface so you can create custom dashboards, reports, and alerts. Data for reporting is gathered from all DNS, DDNS, IPAM, DHCP, Discovery, Advisory Service, and system traffic or events from all members with data transmission enabled within the Grid.

The following tabs are available in Reporting: Settings (configure email notifications); Dashboards (information about the reporting data in the connected grid); Reports (saved searches of reporting data); Alerts (Set alerts to trigger actions when certain events occur); Search (Create a search interactively from scratch and save it as a dashboard panel, alert and report); and Administration (Setup and Permissions).

Report Categories include Audit Log, DNS, DHCP, DDI, Security Network User, FireEye Alerts, DNS Top RPZ Hits, Threat Protection Event, Cloud, System Utilization, and Device (Discovery).

Scheduled Alerts - schedule an alert to notify when a scheduled report returns results that meet a specific condition. The appliance sends an alert when it encounters the trigger condition. You can schedule a report to run on a scheduled interval and trigger an action (e.g. send an email to receive report results) each time it runs.

The Audit Log Events dashboard provides information about the administrator-initiated events such as login events, logout events, service restarts, appliance reboots, write operations such as the addition, modification, and deletion of objects, etc. The default dashboard displays the audit log events for all admin users and for all Grid members in table format. You can use the displayed fields as filters to get specific information you want displayed in the dashboard. Only superusers can view and modify this dashboard.

The Audit Log WAPI Events dashboard provides statistics about the WAPI login session information. It displays the URI, InData and response time for WAPI calls, such as PUT, POST, and DELETE.

Permissions to access, view, edit, and clone searches, dashboards, reports, and alerts that are available in the Reporting tab are restricted to superusers.

The Asset Discovery function is designed to satisfy the following security functional requirements:

- SAD_ARP_EXT.1– The TOE is able to trigger alerts and send SNMP and Email notifications when certain conditions are met.
- SAD_SDC_EXT.1 – The TOE collects System and Event data from the TOE and from external IT systems using SNMP, CLI device querying, ICMP Ping Sweep and Smart Subnet Ping Sweep, TCP Port Scanning, NetBIOS Queries, and vDiscovery.
- SAD_SDR_EXT.1/SAD_SDR_EXT.2 – The TOE provides a Reporting Function restricted to authorized administrators to read the system data from the system data records. The system data is provided in a manner suitable for the user to interpret the information.

## 6.6 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 by themselves 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. There is one exception to this. A system defined role can be assigned to a Limited-Access group in conjunction with the Cloud API permission. The Limited-Access group provides the user with the ability to use the Cloud API to manage the IPAM Automation features.

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 or with Limited-Access Group role with Cloud API permission. Superuser can perform all security management functions. Limited-Access group with the Cloud API permission is restricted to using the Cloud API to manage the IPAM Automation features.

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
- Manage IPAM Automation
- Manage Outbound Notifications.

Management and modification of the behavior of the functions is restricted to the superuser admin. The Limited-Access Group roles with Cloud API permission can only access the Cloud API Service to Manage IPAM Automation for network devices.

**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.
- **Manage IPAM Automation for network devices —**Gives the administrator the ability to automate the DNS and DHCP management features of the Grid instead of manually provisioning IP addresses and DNS name spaces for network devices and interfaces using the Cloud API service. The cloud API service provides the ability to automate management of IP addresses and DNS

records so the cloud environment can take full advantage of IPAM, DNS, and DHCP capabilities in NIOS without the need for manual intervention.

In order to use the cloud API, users must have cloud API permissions assigned in Limited-Access group. Only Superuser and users with the Limited-Access Admin Group roles with the cloud API permission can execute cloud API commands.

The supported cloud API object types, methods, and functions are limited to those identified in the following table. The supported types and operations for cloud API requests are subsets of all types and operations supported on the Grid Master.

*Table 12: Cloud API*

| Supported Object Type | Cloud API Object | Allowed Operations in cloud API Requests |
|---|---|---|
| Network View | networkview | Read, Create, Modify, Delete |
| IPv4 Network Container | networkcontainer | Read, Create, Modify, Delete<br>Function: next_available_network |
| IPv6 Network Container | ipv6networkcontainer | Read, Create, Modify, Delete<br>Function: next_available_network |
| IPv4 Network | network | Read, Create, Modify, Delete<br>Function: next_available_ip |
| IPv6 Network | ipv6network | Read, Create, Modify, Delete<br>Function: next_available_ip |
| IPv4 DHCP Range | range | Read, Create, Modify, Delete<br>Function: next_available_ip |
| IPv6 DHCP Range | ipv6range | Read, Create, Modify, Delete<br>Function: next_available_ip |
| IPv4 Fixed Address (Reservation) | fixedaddress | Read, Create, Modify, Delete<br>Function: next_available_ip<br>You can also create and delete through Grid Manager. All required |

| | | |
|---|---|---|
| | | Cloud EAs are automatically populated in the GUI. |
| IPv6 Fixed Address (Reservation) | ipv6fixedaddress | Read, Create, Modify, Delete<br>Function: next_available_ip<br>You can also create and delete through Grid Manager. All required Cloud EAs are automatically populated in the GUI. |
| DNS View | view | Read, Modify |
| DNS Zone | zone_auth | Read, Create, Modify, Delete |
| Host Record | record:host | Read, Create, Modify, Delete<br>You can also create and delete through Grid Manager. All required Cloud EAs are automatically populated in the GUI. |
| | record:host_ipv4addr | Read, Create, Modify, Delete<br>Function: next_available_ip<br>You can also create and delete through Grid Manager. All required Cloud EAs are automatically populated in the GUI. |
| | record:host_ipv6addr | Read, Create, Modify, Delete<br>Function: next_available_ip<br>You can also create and delete through Grid Manager. All required Cloud EAs are automatically populated in the GUI. |
| Resource Record | record:a | Read, Create, Modify, Delete<br>Function: next_available_ip |
| | record:aaaa | Read, Create, Modify, Delete |

| | | |
|---|---|---|
| | | Function: next_available_ip |
| | record:cname | Read, Create, Modify, Delete |
| | record:ptr | Read, Create, Modify, Delete<br>Function: next_available_ip |
| | record:mx | Read, Create, Modify, Delete |
| | record:naptr | Read, Create, Modify, Delete |
| | record:srv | Read, Create, Modify, Delete |
| | record:txt | Read, Create, Modify, Delete |
| Grid Member | member | Read only<br>Function: restartservices |
| Grid | grid | Read only<br>Function: restartservices |
| Extensible Attribute | extensibleattributedef | Read only |

A Cloud Platform Appliance is a Grid member designed and dedicated to accept and process WAPI (RESTful API) requests related to cloud objects, in addition to serving DNS and DHCP protocols.

The Supported cloud platforms are: VMware vRA/vRO, OpenStack, Kubernetes, and Docker.

- **Manage Outbound Notifications**—The Infoblox outbound API framework gives the administrator the ability to configure outbound Session Management and Action templates, outbound Endpoints (REST API, DXL, or Syslog), and Notification Rules that when combined are used to exchange both IPAM data (such as networks, network containers, hosts, leases) and DNS threat data with external interfaces. The RESTful API and DXL fabric are used to obtain core network service information from the Infoblox Grid to assist administrators with profiling the source or destination of network devices. The RESTful API and WAPI in DXL endpoint are used to change configurations in the Infoblox Grid to help mitigate security threats. In addition to querying inbound data and changing system configurations and query interfaces, the RESTful API and DXL messages are used to send outbound notifications so administrators can prioritize their security needs by detecting new hosts or networks or managing network access control.

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

- FMT_MOF.1(1): The TOE restricts the ability to modify the behavior of the management functions to the superuser.
- FMT_MOF.1(2): The TOE restricts the ability to manage IPAM automation to users with Limited-Access Group role with Cloud API permission.
- 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 roles: 'superuser' and 'Limited-Access Group roles with Cloud API permission' and is able to associate users with these roles.

## 6.7 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.

HA configurations provide hardware redundancy for core network services. The two nodes that form an HA pair—identified as Node 1 and Node 2—are in an active/passive configuration. The active node receives, processes, and responds to all service requests. The passive node constantly keeps its database synchronized with that of the active node, so it can take over services if a hardware or service failure occurs (preservation of secure state). When a hardware or service failure occurs on the active node, the active node becomes passive and the previously passive node becomes active. HA pairs can be configured in IPv4, IPv6, or in dual mode. An IPv4 HA pair uses IPv4 as the communication protocol between the two nodes and an IPv6 HA pair uses IPv6 as the communication protocol between the two nodes. But in a dual mode HA pair, you can select either IPv4 or IPv6 as the communication protocol between the two nodes.

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 requirements:

- FPT_FLS.1/FRU_FLT.1: TOE HA configurations provide hardware redundancy and fault tolerance with preservation of a secure state for core network services when there are HA node hardware or Service failures.
- 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.

## 6.8 TOE Access

The TOE terminates local and remote interactive sessions after an administrator configurable time interval of between 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 unauthorized use of the TOE.

## 6.9 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/or API that is protected using HTTPS/TLS.
- **Syslog Server**—All communication with the syslog server is protected with TLS
- **Active Directory, LDAP Servers**—All communication with these external authentication servers is protected with TLS.
- **RADIUS, and TACACS+ Servers**—All communication with these external authentication servers is protected with IPsec.
- **Backup/Restore Servers** – The TOE provides the ability to back up stored data to/restore backed up data with a remote backup server over SFTP. All communication with the Backup and Restore Servers are protected with SSH.
- **Advisor**—All communication with the TOE's Advisor Subscription Service is protected using HTTPS.

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

- FTP_ITC.1: The TOE utilizes TLS to protect all data transmitted between the TOE and trusted external third-party IT entities from unauthorized 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/APIs (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

This section shows that all threats and OSPs are completely covered by security objectives for the TOE. In addition, each objective counters or addresses at least one threat or OSP.

*Table 13: Threats and OSPs to TOE Security Objectives Correspondence*

| | T.PASSWORD_CRACKING | T.MALICIOUS_DNS | T.SECURITY_FUNCTIONALITY_FAILURE | T. UNAUTHORIZED_ADMINISTRATOR_ACCESS | T.UNDETECTED_ACTIVITY | T.UNDETECTED_ASSETS | T.UNTRUSTED_COMMUNICATION_CHANNELS | T.UPDATE_COMPROMISE | T.WEAK_AUTHENTICATION_ENDPOINTS | T. WEAK_CRYPTOGRAPHY | P.ACCESS_BANNER |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **O.AUDIT_GENERATION** | | | | | X | | | | | | |
| **O.DISCOVERY** | | | | | | X | | | | | |
| **O.DISPLAY_BANNER** | | | | X | | | | | | | X |
| **O.HIGH_AVAILABILITY** | | | X | | | | | | | | |
| **O.MALICIOUS_DNS** | | X | | | | | | | | | |
| **O.PROTECTED_COMMUNIATIONS** | | | | | | | X | | X | | |
| **O.STRONG_CRYPTOGRAPHY** | | | | | | | X | X | X | X | |

| | T.PASSWORD_CRACKING | T.MALICIOUS_DNS | T.SECURITY_FUNCTIONALITY_FAILURE | T. UNAUTHORIZED_ADMINISTRATOR_ACCESS | T.UNDETECTED_ACTIVITY | T.UNDETECTED_ASSETS | T.UNTRUSTED_COMMUNICATION_CHANNELS | T.UPDATE_COMPROMISE | T.WEAK_AUTHENTICATION_ENDPOINTS | T. WEAK_CRYPTOGRAPHY | P.ACCESS_BANNER |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **O.TOE_ADMINISTRATION** | X | | | X | | | | | | | |
| **O.TSF_SELF_TEST** | | | X | | | | | | | | |
| **O.VERIFIABLE_UPDATES** | | | | | | | | X | | | |

**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 use the management interfaces to configure the TOE and its network, mechanisms that control a user's logical access to the TOE and mechanisms to ensure strong passwords.

**T.MALICIOUS_DNS**

*DNS queries that originate from external sources may be to a malicious or unauthorized host designed to infiltrate and damage the system.*

This threat is countered by ensuring that:

O. MALICIOUS_DNS: The TOE must be able to detect and react to potential data exfiltration tunnels using its threat analytics service and be able to validate a response to DNS query (a DNS record) before returning it to the client.

**T.SECURITY_FUNCTIONALITY_FAILURE**

*A component of the TOE may fail during start-up or during operations causing a compromise or failure in the security functionality of the TOE, leaving the TOE unavailable and/or 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.

- O.HIGH_AVAILABILITY: The TOE will provide hardware and fault tolerance capabilities for core network services to ensure the services are available in the event of a hardware or software failure.

**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 use the management interfaces to configure the TOE and its network, 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 threat is countered 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.UNDETECTED_ASSETS**

*Administrators might not be able to detect assets or networks in their security infrastructure allowing undetected vulnerabilities in the system such as equipment failures and obsolete devices or software potentially leading to an insecure system.*

This threat is countered by ensuring that:

O.DISCOVERY**:** The TOE will provide asset discovery methods; advisory services; alerting and reporting capabilities to assist administrators in monitoring and maintaining network and security infrastructure.

**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.

### 7.1.2 Security Objectives Rationale for the Operational Environment

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

*Table 14: Assumptions to Operational Environment Security Objectives Correspondence*

| | A.ADMIN_CREDENTIALS_SECURE | A.LIMITED_FUNCTIONALITY | A.NO_THRU_TRAFFIC_PROTECTION | A. TRUSTED_ADMINISTRATOR | A.PHYSICAL_PROTECTION | A. .REGULAR_UPDATES |
|---|---|---|---|---|---|---|
| **OE.ADMIN_CREDENTIALS_SECURE** | X | | | | | |
| **OE. NO_GENERAL_PURPOSE** | | X | | | | |
| **OE. NO_THRU_TRAFFIC_PROTECTION** | | | X | | | |
| **OE.PHYSICAL** | | | | | X | |
| **OE. TRUSTED_ADMIN** | | | | X | | |
| **OE.UPDATES** | | | | | | X |

**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 15 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.

*Table 15: Objectives to Requirements Correspondence*

| | O.AUDIT_GENERATION | O.DISCOVERY | O.DISPLAY_BANNER | O.HIGH_AVAILABILITY | O. MALICIOUS_DNS | O.PROTECTED_ COMMUNICATIONS | O.STRONG_CRYPTOGRAPHY | O.TOE_ADMINISTRATION | O.TSF_SELF_TEST | O.VERIFIABLE_UPDATES |
|---|---|---|---|---|---|---|---|---|---|---|
| **DTC_DTA_EXT.1** | | | | | X | | | | | |
| **DTC_RCT_EXT.1** | | | | | X | | | | | |
| **FAU_GEN.1** | X | | | | | | | | | |
| **FAU_GEN.2** | X | | | | | | | | | |
| **FAU_STG_EXT.1** | 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 | | | |
| **FCS_DNSSEC_EXT.1** | | | | | X | | | | | |
| **FCS_SSHC_EXT.1** | | | | | | X | X | | | |
| **FCS_SSHS_EXT.1** | | | | | | X | X | | | |
| **FCS_TLSC_EXT.1** | | | | | | X | X | | | |
| **FCS_TLSS_EXT.1** | | | | | | X | X | | | |
| **FIA_RADIUS_EXT.1** | | | | | | | | X | | |
| **FIA_SOS.1** | | | | | | | | X | | |
| **FIA_UAU.2** | | | | | | | | X | | |
| **FIA_UAU.5** | | | | | | | | X | | |
| **FIA_UAU.7** | | | | | | | | X | | |

| | O.AUDIT_GENERATION | O.DISCOVERY | O.DISPLAY_BANNER | O.HIGH_AVAILABILITY | O. MALICIOUS_DNS | O.PROTECTED_ COMMUNICATIONS | O.STRONG_CRYPTOGRAPHY | O.TOE_ADMINISTRATION | O.TSF_SELF_TEST | O.VERIFIABLE_UPDATES |
|---|---|---|---|---|---|---|---|---|---|---|
| **FIA_UID.2** | | | | | | | | X | | |
| **FIA_X509_EXT.1** | | | | | | | | X | | |
| **FIA_X509_EXT.2** | | | | | | | | X | | |
| **SAD_ARP_EXT.1** | | X | | | | | | | | |
| **SAD_SDC_EXT.1** | | X | | | | | | | | |
| **SAD_SDR_EXT.1** | | X | | | | | | | | |
| **SAD_SDR_EXT.2** | | X | | | | | | | | |
| **FMT_MOF.1(1)** | | | | | | | | X | | |
| **FMT_MOF.1(2)** | | | | | | | | X | | |
| **FMT_MTD.1** | | | | | | | | X | | |
| **FMT_SMF.1** | | | | | | | | X | | |
| **FMT_SMR.1** | | | | | | | | X | | |
| **FPT_FLS.1** | | | | X | | | | | | |
| **FPT_ITT.1** | | | | | | X | | | | |
| **FPT_STM.1** | X | | | | | | | | | |
| **FPT_TST_EXT.1** | | | | | | | | | X | |
| **FPT_TUD_EXT.1** | | | | | | | | | | X |
| **FRU_FLT.1** | | | | X | | | | | | |
| **FTA_SSL.3** | | | | | | | | X | | |
| **FTA_SSL.4** | | | | | | | | X | | |
| **FTA_TAB.1** | | | X | | | | | | | |
| **FTP_ITC.1** | | | | | | X | | | | |
| **FTP_TRP.1** | | | | | | X | | | | |

**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_EXT.1: The TOE is required to provide the ability to transmit audit data to a remote syslog server over TLS.
- FAU_STG.1: The TOE is required to protect the stored audit records in the audit trail from unauthorized deletion and prevents unauthorized 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.DISCOVERY**

*The TOE will provide asset discovery methods; advisory services; alerting and reporting capabilities to assist administrators in monitoring and maintaining network and security infrastructure.*

This TOE Security Objective is satisfied by ensuring that:

- SAD_ARP_EXT.1: The TOE is able to trigger alerts when certain events occur.
- SAD_SDC_EXT.1: The TOE is able to discover assets and networks and provides advisory services.
- SAD_SDR_EXT.1: The TOE provides the system data to administrators in a readable format.
- SAD_SDR_EXT.2: the TOE restricts read access to the system data to only superusers.

**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 unauthorized use of the TOE.

**O.HIGH_AVAILABILITY**

*The TOE will provide preservation of secure state, hardware and fault tolerance capabilities for core network services.*

This TOE Security Objective is satisfied by ensuring that:

- FPT_FLS.1/FRU_FLT.1: When there are HA node hardware or Service failures, the TOE's hardware redundancy and fault tolerance services preserve a secure state and continue to provide core network services.

**O. MALICIOUS_DNS**

*The TOE must be able to detect and react to potential data exfiltration tunnels using its threat analytics service and be able to validate a response to DNS query (a DNS record) before returning it to the client.*

This TOE Security Objective is satisfied by ensuring that:

- DTC_DTA_EXT.1: The TOE applies its streaming analytics, whitelist and blacklist rules and enforce RPZs (local and threat feed) on the DNS traffic.
- DTC_RCT_EXT.1: The TOE blacklists offending Domains and sends an SNMP trap each time it detects a new blacklisted domain.
- FCS_DNSSEC_EXT.1 – The TOE's DNSSEC function is able to authenticate and verify the integrity of DNS data. If a DNS response (DNS data) cannot be authenticated, the TOE does not return the DNS data.

**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.
- FCS_SSHC_EXT.1: The TOE is required to implement the SSH Client Protocol.
- FCS_SSHS_EXT.1: The TOE is required to implement the SSH Server Protocol.
- FCS_TLSC_EXT.1: The TOE is required to implement the TLS Client Protocol.
- FCS_TLSS_EXT.1: The TOE is required to implement the TLS Server Protocol.

- 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 unauthorized 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 to meet the standards: AES as specified in ISO 18033-3, CBC as specified in ISO 10116, with the specified bit sizes and mode 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.
- FCS_SSHC_EXT.1: The TOE is required to implement the SSH Client Protocol.
- FCS_SSHS_EXT.1: The TOE is required to implement the SSH Server Protocol.
- FCS_TLSC_EXT.1: The TOE is required to implement the TLS Client Protocol
- FCS_TLSS_EXT.1: The TOE is required to implement the TLS Server Protocol

**O.TOE_ADMINISTRATION**

*The TOE will provide mechanisms to ensure that only administrators are able to log in and use the management interfaces to configure the TOE and its network, 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_RADIUS_EXT.1: The TOE supports administrator authentication using external RADIUS authentication servers. The TOE implements a RADIUS client protocol according to standards to ensure the proper authentication of users.
- 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 X509 Certificates, Two-factor authentication, Active Directory, LDAP, SAML, RADIUS, TACACS+ authentication.
- FIA_UAU.7: The TOE is required to provide only obscured feedback to the user while the authentication is in progress.
- FIA_X509_EXT.1: The TOE is required to provide X.509 certificate validation for administrators.
- FIA_X509_EXT.2: The TOE is required to provide X.509 certificate authentication for administrators.
- FMT_MOF.1(1): The TOE is required to restrict the ability to modify the behavior of the management functions (except Manage IPAM automation) to the superuser role.
- FMT_MOF.1(2): The TOE is required to restrict the ability to modify the behavior of the IPAM automation management function to the superuser role and Limited-Access Group role with Cloud API permission.
- 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 operate and configure the TOE and its environment.
- FMT_SMR.1: The TOE is required to maintain the security roles: 'superuser', 'Limited-Access Group role with Cloud API permission' and is able to associate users with these roles.
- 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.

*Table 16: Requirement Dependencies*

| 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_EXT.1 | FAU_GEN.1, and FTP_ITC.1 | FAU_GEN.1, and FTP_ITC.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 |

| ST Requirement | CC Dependencies | ST Dependencies |
|---|---|---|
| FCS_COP.1(5) | (FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4 | See DRBG Note Below. |
| FCS_DNSSEC_EXT.1 | FCS_COP.1(3) | FCS_COP.1(3) |
| FCS_SSHC_EXT.1 | FCS_CKM.1, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_COP.1(5) | FCS_CKM.1, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_COP.1(5) |
| FCS_SSHS_EXT.1 | FCS_CKM.1, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_COP.1(5) | FCS_CKM.1, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_COP.1(5) |
| FCS_TLSC_EXT.1 | FCS_CKM.1, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_COP.1(5) | FCS_CKM.1, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_COP.1(5) |
| FCS_TLSS_EXT.1 | FCS_CKM.1, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_COP.1(5) | FCS_CKM.1, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_COP.1(5) |
| FIA_RADIUS_EXT.1 | None | None |
| 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 |
| FIA_X509_EXT.1 | FIA_X509_EXT.2 | FIA_X509_EXT.2 |
| FIA_X509_EXT.2 | FIA_X509_EXT.1 | FIA_X509_EXT.1 |
| SAD_ARP_EXT.1 | SAD_SDC_EXT.1 | SAD_SDC_EXT.1 |
| SAD_SDC_EXT.1 | None | None |
| SAD_SDR_EXT.1 | SAD_SDC_EXT.1 | SAD_SDC_EXT.1 |
| SAD_SDR_EXT.2 | SAD_SDR_EXT.1 | SAD_SDR_EXT.1 |
| FMT_MOF.1(1) | FMT_SMR.1 and FMT_SMF.1 | FMT_SMR.1 and FMT_SMF.1 |
| FMT_MOF.1(2) | 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 |
| FMT_SMF.1 | None | None |
| FMT_SMR.1 | FIA_UID.1 | FIA_UID.2 |
| FPT_FLS.1 | None | None |
| 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) |
| FRU_FLT.1 | FPT_FLS.1 | FPT_FLS.1 |

| ST Requirement | CC Dependencies | ST Dependencies |
|---|---|---|
| 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.

## 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 17 demonstrates the relationship between security requirements and security functions.

*Table 17: Security Functions vs. Requirements Mapping*

| | Security audit | Cryptographic support | Identification and authentication | Asset Discovery | 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 | | | | | | |
| **FCS_DNSSEC_EXT.1** | | X | | | | | | |
| **FCS_TLSC_EXT.1** | | X | | | | | | |
| **FCS_TLSS_EXT.1** | | X | | | | | | |
| **FIA_RADIUS_EXT.1** | | | X | | | | | |
| **FIA_SOS.1** | | | X | | | | | |
| **FIA_UAU.2** | | | X | | | | | |
| **FIA_UID.2** | | | X | | | | | |
| **FIA_UAU.5** | | | X | | | | | |
| **FIA_UAU.7** | | | X | | | | | |

| | Security audit | Cryptographic support | Identification and authentication | Asset Discovery | Security management | Protection of the TSF | TOE Access | Trusted Path/Channels |
|---|---|---|---|---|---|---|---|---|
| **FIA_X509_EXT.1** | | | X | | | | | |
| **FIA_X509_EXT.2** | | | X | | | | | |
| **SAD_ARP_EXT.1** | | | | X | | | | |
| **SAD_SDC_EXT.1** | | | | X | | | | |
| **SAD_SDR_EXT.1** | | | | X | | | | |
| **SAD_SDR_EXT.2** | | | | X | | | | |
| **FMT_MOF.1(1)** | | | | | X | | | |
| **FMT_MOF.1(2)** | | | | | X | | | |
| **FMT_MTD.1** | | | | | X | | | |
| **FMT_SMF.1** | | | | | X | | | |
| **FMT_SMR.1** | | | | | X | | | |
| **FPT_FLS.1** | | | | | | X | | |
| **FPT_ITT.1** | | | | | | X | | |
| **FPT_STM.1** | | | | | | X | | |
| **FPT_TST_EXT.1** | | | | | | X | | |
| **FPT_TUD_EXT.1** | | | | | | X | | |
| **FRU_FLT.1** | | | | | | X | | |
| **FTA_SSL.3** | | | | | | | X | |
| **FTA_SSL.4** | | | | | | | X | |
| **FTA_TAB.1** | | | | | | | X | |
| **FTP_ITC.1** | | | | | | | | X |
| **FTP_TRP.1** | | | | | | | | X |