# NIKSUN NetDetector/NetVCR 10440

## Security Target

ST Version: 1.0
June 22nd, 2018

**NIKSUN, Inc.**
457 N. Harrison St
Princeton, NJ 08540

Prepared By:

Booz | Allen | Hamilton
delivering results that endure

Cyber Assurance Testing Laboratory
1100 West Street
Laurel, MD 20707

# Table of Contents

# Table of Figures

# Table of Tables

# 1 Security Target Introduction

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

## 1.1 ST Reference

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

### 1.1.1 ST Identification

| | |
|---|---|
| **ST Title:** | NIKSUN NetDetector/NetVCR 10440 Security Target |
| **ST Version:** | 1.0 |
| **ST Publication Date:** | June 22nd, 2018 |
| **ST Author:** | Booz Allen Hamilton |

### 1.1.2 Document Organization

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

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

*Chapter 3* describes the conformance claims made by this ST.

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

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

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

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

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

### 1.1.3 Terminology

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

| Term | Definition |
|---|---|
| **Administrator** | The web GUI role that is considered a Security Administrator for the evaluation. TOE users with this role will configure and manage the TOE security functions and manages audit records. |
| **NetDetector/NetVCR** | The TOE is a monitoring system that captures, records, and analyzes traffic streams on the monitored network for network security and performance. The NetDetector/NetVCR appliance communicates with a NetOmni appliance that is in the Operational Environment. NetDetector/NetVCR provides information on the performance of the network, and NetOmni provides instructions to the NetDetector/NetVCR. |
| **NetOmni** | It provides an overview of critical operations of the monitored network. |
| **VAR Log** | Log file that contains TOE audit records which is stored in the /var/log directory. |
| **VCR User** | The CLI role that is considered to be a Security Administrator for the evaluation. It manages security functions and audit records. The VCR user assumes the Linux root role to perform some of its TOE management functions using the command 'su root'. |

**Table 1: Customer Specific Terminology**

| Term | Definition |
|---|---|
| **Security Administrator** | The claimed Protection Profile defines a Security Administrator role that is authorized to manage the TOE and its data. For the TOE, this is considered to be any user with the 'Administrator' role in the web GUI and the 'VCR User' in the CLI. |
| **Trusted Channel** | An encrypted connection between the TOE and a system in the Operational Environment. |
| **Trusted Path** | An encrypted connection between the TOE and the application a Security Administrator uses to manage it (web browser, terminal client, etc.). |
| **User** | In a CC context, any individual who has the ability to access the TOE functions or data. |

**Table 2: CC Specific Terminology**

### 1.1.4 Acronyms

The acronyms used throughout this ST are defined in Table 1-3. This table is to be used by the reader as a quick reference guide for acronym definitions.

| Acronym | Definition |
|---|---|
| **CA** | Certificate Authority |
| **CC** | Common Criteria |
| **CLI** | Command-Line Interface |
| **cPP** | collaborative Protection Profile |
| **CPU** | Central Processing Unit |
| **CRL** | Certificate Revocation List |
| **CSR** | Certificate Signing Request |
| **CVL** | Component Validation List |
| **DN** | Distinguished Name |
| **DNS** | Domain Name Server |

| | |
|---|---|
| **DRBG** | Deterministic Random Bit Generator |
| **GUI** | Graphical User Interface |
| **HTTP** | Hypertext Transfer Protocol |
| **HTTPS** | Hypertext Transfer Protocol Secure |
| **IDS** | Intrusion Detection System |
| **IP** | Internet Protocol |
| **IT** | Information Technology |
| **KAS** | Key Agreement Scheme |
| **KDF** | Key Derivation Function |
| **LDAP/AD** | Lightweight Directory Access Protocol / Active Directory |
| **NDcPP** | Network Device collaborative Protection Profile |
| **NIAP** | National Information Assurance Partnership |
| **NTP** | Network Time Protocol |
| **OS** | Operating System |
| **OSP** | Organizational Security Policy |
| **PP** | Protection Profile |
| **RAM** | Random Access Memory |
| **RBG** | Random Bit Generator |
| **RU** | Rack Unit |
| **SAN** | Subject Alternative Name |
| **SAR** | Security Assurance Requirement |
| **SCP** | Secure Copy Protocol |
| **SFP** | Security Function Policy |
| **SFR** | Security Functional Requirement |
| **SMTP** | Simple Mail Transfer Protocol |
| **SSH** | Secure Shell |
| **ST** | Security Target |
| **TCP** | Transmission Control Protocol |
| **TLS** | Transport Layer Security |
| **TOE** | Target of Evaluation |
| **TSF** | TOE Security Function |
| **UI** | User Interface |

**Table 3: Acronym Definition**

### 1.1.5 Reference

[1] Collaborative Protection Profile for Network Devices, version 2.0 + Errata 20180314 (NDcPP)

[2] Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated September 2012, version 3.1, Revision 4, CCMB-2012-009-001

[3] Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components, dated September 2012, version 3.1, Revision 4, CCMB-2012-009-002

[4] Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components, dated September 2012, version 3.1, Revision 4, CCMB-2012-009-003

[5] Common Methodology for Information Technology Security Evaluation – Evaluation Methodology, dated September 2012, version 3.1, Revision 4, CCMB-2012-009-004

[6] NIST Special Publication 800-38A Recommendation for Block Cipher Modes of Operation, December 2001

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

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

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

[10] FIPS PUB 197 Advanced Encryption Standard November 26 2001

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

[12] NIKSUN NetDetector Release 5.1.2.0 DISA HARDWARE INSTALLATION GUIDE

[13] NIKSUN NetDetector Release 5.1.2.0 DISA SOFTWARE UPGRADE & REINSTALLATION GUIDE

[14] NIKSUN Appliance NikOS Everest ADMINISTRATOR GUIDE 5.1.2.0

[15] NIKSUN Security Configuration & Tools NikOS Everest CONFIGURATION GUIDE 5.1.2.0

[16] NIKSUN Military Unique Deployment Guide Release 5.1.2.0 Version 1.0

[17] NIKSUN NetDetector/NetVCR Release 5.1.2.0 Supplemental Administrative Guidance for Common Criteria Version 1.0

## 1.2 TOE Reference

The TOE is the NIKSUN NetDetector/NetVCR 10440 appliance, running the NIKSUN NetDetector/NetVCR Everest software version 5.1.2.0.

## 1.3 TOE Overview

The TOE includes the NIKSUN NetDetector/NetVCR 10440 appliance, running the software NIKSUN NetDetector/NetVCR Everest version 5.1.2.0. The TOE allows Security Administrators to access the TOE through a local CLI, remote CLI via SSH, and a web GUI via TLS/HTTPS. The TOE was evaluated against the Security Functional Requirements defined in Section 6.3 only. Refer to Section 2.5 for a summary of the functional claims tested.

NetDetector/NetVCR's primary functionality is to provide security monitoring of network traffic using IDS methods and statistical anomaly detection in order to safeguard networks against cyber-attacks. The anomaly detection uses user-defined and threshold-based anomalies. Users of NetDetector/NetVCR are notified of security breaches as soon as they occur. NetDetector/NetVCR sends information about detected events to NetOmni for data aggregation and to provide real-time network-wide analysis.

The TOE was evaluated as a network device only and NetDetector/NetVCR's network monitoring and cyber-attack detection described above were not assessed during this evaluation.

The following figure depicts the TOE boundary:

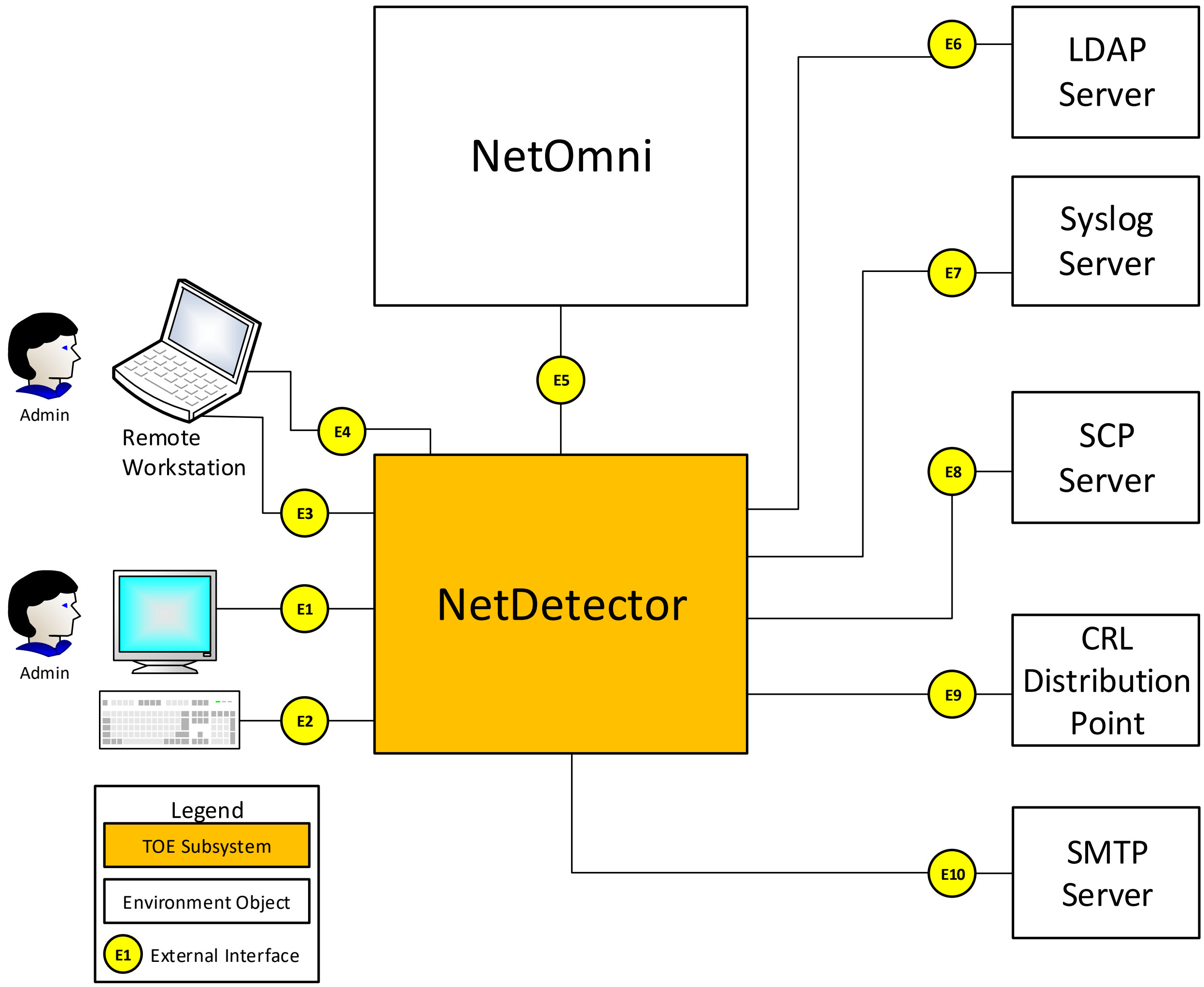

**Figure 1: TOE Boundary for NetDetector/NetVCR**

As illustrated in Figure 1, the NetDetector/NetVCR is a single appliance that has connections to a management workstation, support servers, and a NetOmni appliance. Administrators have the ability to manage the TOE both locally and remotely. NetDetector/NetVCR has connections to a monitor and keyboard for local CLI management (E1, E2). Remote administration is conducted by connecting to the remote CLI (E3) protected by SSH or the web GUI (E4) which is protected by TLS/HTTPS. The TOE also connects to several servers in its Operational Environment which support its normal functions. The TOE connects to an instance of a NetOmni appliance using HTTPS/TLS (E5). Remote authentication for TOE administrators is performed by a remote LDAP/AD Server whose connection to the TOE is secured by TLS (E6). The TOE transfers audit records and VAR logs to a remote Syslog Server via TLS (E7). The TOE receives software updates from the SCP Server (E8). The connection with the SCP Server is secured using SSH. A CRL Distribution Point is used to determine the validity of certificates provided by an entity in the Operational Environment when it attempts to connect to the TOE (E9). When a user utilizes the "Forgot Username/Password" feature on the NetDetector/NetVCR login screen, NetDetector/NetVCR will send an email to the SMTP Server over a protected TLS channel (E10). If the user forgot their username, the user can enter the email associated with the username and the TOE will send an email using SMTP to that email account. If the user has forgotten the password, the user can enter their username and the TOE will send an email using SMTP to the email address associated with that username. The email will contain a link that directs the user to the TOE's web GUI to be able to securely change their password.

## 1.4 TOE Type

The TOE is a network device as defined in the NDcPP which states: "This is a Collaborative Protection Profile (cPP) whose Target of Evaluation (TOE) is a network device… A network device in the context of this cPP is a device composed of both hardware and software that is connected to the network and has an infrastructure role within the network". The TOE consists of the NIKSUN NetDetector/NetVCR 10440 model, running the software NIKSUN NetDetector/NetVCR Everest version 5.1.2.0. Thus, the TOE is a network device composed of hardware and software. Within the infrastructure of the network, NetDetector/NetVCR performs security monitoring of network traffic using IDS methods in order to detect cyber-attacks. Because the device is connected to and has an infrastructure purpose within the network, this conformance claim is appropriate.

# 2 TOE Description

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

## 2.1 Evaluated Components of the TOE

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

| Component |
|---|
| NetDetector/NetVCR, running the NIKSUN NetDetector/NetVCR Everest 5.1.2.0 Software Version |

**Table 4: Evaluated Components of the TOE**

## 2.2 Components and Applications in the Operational Environment

The following table lists components and applications in the environment that the TOE relies upon in order to function properly:

| Component | Definition |
|---|---|
| **LDAP/AD Server** | A system that is capable of receiving authentication requests using LDAP over TLS and validating these requests against identity and credential data that is defined in an LDAP directory. In the evaluated configuration, the TOE connects to a server with OpenLDAP for its remote authentication store. |
| **Management Workstation** | Any general-purpose computer that is used by a Security Administrator to manage the TOE. The TOE can be managed remotely, in which case the management workstation requires an SSH client to access the CLI or a web browser (Microsoft Internet Explorer 9.0 or higher and Mozilla Firefox 3.6 or higher) to access the web GUI. |

| | |
|---|---|
| **Syslog Server** | The Syslog Server connects to the TOE and allows the TOE to send Syslog messages to it for remote storage. This is used to send copies of audit data to be stored in a remote location for data redundancy purposes. |
| **SCP Server** | A secure server used to ensure the secure copying of data through an SSH encrypted connection. In the evaluated configuration, the SCP Server is used to transfer software updates to the TOE's software image directory. |
| **CRL Distribution Point** | A server deployed within the Operational Environment which confirms the validity and revocation status of certificates. |
| **NetOmni** | NetDetector/NetVCR is a network security and performance monitoring system which sends captured packet data to NetOmni. The NetDetector/NetVCR also receives commands and is managed by NetOmni. The TOE communicates with NetOmni over an encrypted channel. |
| **SMTP Server** | A server that forwards an email that is sent from NetDetector/NetVCR when a user utilizes the "Forgot Username/Password" feature on the NetDetector/NetVCR log in screen. The email is protected from unauthorized disclosure using TLS. |

**Table 5: Evaluated Components of the Operational Environment**

## 2.3 Excluded from the TOE

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

### 2.3.1 Not Installed

This section contains components or software that were not installed for this evaluation:

- Public Dashboard license – displays public status of reports
- Net Updater license
- Software Updater license

### 2.3.2 Installed but Requires a Separate License

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

### 2.3.3 Installed but Not Part of the TSF

There are no components that are part of the purchased product but are not part of the TSF relevant functionality that is being evaluated as the TOE.

## 2.4 Physical Boundary

### 2.4.1 Hardware

NetDetector/NetVCR is a rack-mounted hardware device. The evaluated model's specific hardware and configuration is as follows:

| Property | NetDetector/NetVCR |
|---|---|
| **Model Number** | NIKSUN 10440 Platform |

| | |
|---|---|
| **Size** | 10 RUs+ |
| **Power** | AC |
| **Power Supplies** | 5 + 5 Redundant |
| **Processor** | Intel Xeon E5-2660 v2 |
| **Memory (RAM)** | 256GB (Max) |
| **Storage** | 160TB |
| **External Storage** | Yes |
| **RAID** | Yes |

**Table 6: NetDetector/NetVCR Series**

### 2.4.2 Software

- NIKSUN NetDetector/NetVCR Everest Release 5.1.2.0
- FreeBSD 11.1 Operating System with NIKOS FIPS Object Module 2.0.16 (derived from OpenSSL FIPS Object Module 2.0.7)

## 2.5 Logical Boundary

The TOE is comprised of several security features. Each of the security features identified above consists of several security functionalities, as identified below.

1. Security Audit
2. Cryptographic Support
3. Identification and Authentication
4. Security Management
5. Protection of the TSF
6. TOE Access
7. Trusted Path/Channels

### 2.5.1 Security Audit

Audit records are generated for various types of management activities and events. These records include the date and time stamp of the event, the event type, and the subject identity. Audit records are stored as syslog records on the TOE, and can be configured to also be sent to a Syslog Server via a TLS connection. When the storage space allocated to specific audit record types is exhausted, the TOE will overwrite the oldest relevant log file. Administrators are assumed to be trusted users and are not expected to delete or modify the audit records. Applications that run from the CLI keep their own VAR log files, such as Apache, LDAP client, etc. These log files are also sent via TLS to the Syslog Server.

### 2.5.2 Cryptographic Support

The TOE relies on its NIKOS FIPS Object Module 2.0.16 (derived from OpenSSL FIPS Object Module 2.0.7) cryptographic module to implement cryptographic methods and trusted channels. X.509v3 certificates are used to support authentication mechanisms. SSH is used to secure the remote CLI interface for remote management of the TOE. SSH is also used to secure the communication with the SCP Server when the TOE receives software image updates. The TOE uses TLS to secure the automatic transfer of syslog audit files and VAR logs to the Syslog Server, and for connection to the LDAP/AD Server for remote authentication. When a user utilizes the "Forgot Username/Password" feature on the NetDetector/NetVCR login screen, NetDetector/NetVCR will send an email to the SMTP Server over a protected TLS channel. TLS/HTTPS is used to secure the connection for remote management of the TOE via the web GUI as well as connections to NetOmni devices. The TOE will deny any connections for disallowed protocols and invalid X.509v3 certificates.

Cryptographic keys are generated using the CTR_DRBG provided through this module. The TOE destroys all plaintext secrets and private keys.

The following table contains the CAVP algorithm certificates:

| Algorithm | CAVP Cert. # |
|---|---|
| AES-128-CBC, AES-256-CBC | 5418 |
| RSA (FIPS 186-4 KeyGen, SigGen and SigVer) | 2902 |
| CTR_DRBG (AES) | 2113 |
| SHA-1, SHA-256, SHA-512 | 4349 |
| HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-512 | 3588 |
| CVL (FFC 800-56A except key derivation function) | 1871 |
| DSA | 1394 |

**Table 7 Cryptographic Algorithm Table**

### 2.5.3 Identification and Authentication

The TOE verifies the identity of users connecting to the TOE. All users must be identified and authenticated before being allowed to perform actions on the TOE. This is true of users accessing the TOE via the local console, or through protected paths using the remote CLI via SSH or the web GUI via TLS 1.2. Users can authenticate to the TOE using a username and password. In addition, when authenticating by the remote CLI, users can instead use SSH public-key authentication. LDAP can be configured to provide external authentication. Passwords can consist of upper case letters, lower case letters, numbers, and a set of selected special characters. Password information is never revealed during the authentication process including during login failures. Before a user authenticates to the device, a customizable warning banner is configured to be displayed. In addition, via the web GUI only, the user has the option to use a "Forgot Username/Password" feature prior to authenticating.

The TOE uses X.509v3 certificates to perform server side authentication of Syslog Server, SMTP Server and LDAP/AD Server and present its certificate to NetOmni for authentication. The TSF determines the validity of the certificates by confirming the validity of the certificate chain, and verifying that the certificate chain ends in a trusted Certificate Authority (CA). The TSF connects with a CRL distribution point through HTTP to confirm certificate validity and to access certificate revocation lists (CRL).

### 2.5.4 Security Management

The TOE has a role based authentication system where roles (permissions) are assigned to groups for the web GUI. Authorized actions for a particular user are dependent on which group they are assigned to. There are 4 initial groups: Administrator, Account Administrator, Advanced Users, and Users. Only users assigned to the Administrator group are capable of performing SFR related management functions via the web GUI and thus, are Security Administrators in the context of the evaluation. The VCR user is the Security Administrator user for the remote and local CLI, and is able to update the TOE's software and verify it via published hash.

The NDcPP's definition of "role" is synonymous with NIKSUN's definition of "permissions". NIKSUN's terminology fits into the Protection Profiles by using the term "user roles" in place of "user permissions". For the remainder of this document, "user permissions" is used in order to match the terminology used by Common Criteria.

### 2.5.5 Protection of the TSF

The TOE stores passwords in a variety of locations depending on their use and encryption. They cannot be viewed by any user regardless of the user's role. The VCR user passwords are stored in the OS hashed by SHA-512. Web GUI passwords are stored in the PostgreSQL Database hashed with SHA-256. Pre-shared keys, symmetric keys, and private keys cannot be accessed in plaintext form by any user. There is an underlying hardware clock that is used for accurate timekeeping and is set by the Security Administrator. Power-on self-tests are executed automatically when the cryptographic module is loaded into memory. It verifies its own integrity using an HMAC-SHA-256 digest computed at build time and also tests all algorithms for integrity. The TOE also performs self-tests on the CPU, RAM, and disk components. The TOE's DRBG also performs its own health tests.

The version of the TOE is verified via the CLI or web GUI. The TOE is updated by the VCR user via the CLI. Updated software images are downloaded to the SCP Server and are transferred to the TOE via the SCP using SSH. The administrator is also capable of copying the image to a CD and manually loading it to the TOE. The TOE conducts a hash verification on the system image using SHA-256 against the known hash to ensure the integrity of the update.

### 2.5.6 TOE Access

Before any user authenticates to the TOE, the TOE displays a configurable Security Administrator banner for the web GUI. The local and remote CLI interfaces display the default Linux security banner prior to authentication that is also configurable. The TOE can terminate local CLI, remote CLI, and web GUI sessions after a specified time period of inactivity. Administrator users have the capability to terminate their own sessions.

### 2.5.7 Trusted Path/Channels

The TOE connects and sends data to IT entities that reside in the Operational Environment via trusted channels. In the evaluated configuration, the TOE connects to Syslog Server via TLS to send audit data for remote storage. TLS is used for the TOE's connection with the SMTP Server to send secure email. TLS is also used for the TOE's connection with the LDAP/AD Server for its remote authentication store. TLS/HTTPS is used for the transfer of data to the NetOmni appliance. TLS/HTTPS and SSH are used for remote administration of the TOE via the web GUI and remote CLI respectively.

# 3 Conformance Claims

## 3.1 CC Version

This ST is compliant with Common Criteria for Information Technology Security Evaluation, Version 3.1 Revision 4 September 2012.

## 3.2 CC Part 2 Conformance Claims

This ST and Target of Evaluation (TOE) is Part 2 extended to include all applicable NIAP and International interpretations through June 22, 2018.

## 3.3 CC Part 3 Conformance Claims

This ST and Target of Evaluation (TOE) are conformant to Part 3 to include all applicable NIAP and International interpretations through June 22, 2018.

## 3.4 PP Claims

This ST claims exact conformance to the following Protection Profiles:

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

## 3.5 Package Claims

The TOE claims exact conformance to the NDcPP, which is conformant with CC Part 3.

The TOE claims following Selection SFRs that are defined in the appendices of the claimed cPP:

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

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

- FMT_MTD.1/CryptoKeys

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

## 3.6 Package Name Conformant or Package Name Augmented

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

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

- 0324: NIT Technical Decision for Correction of section numbers in SD Table 1
- 0321: Protection of NTP communications (Note: This applies due to the application note wording change. The TOE does not claim the use of NTP for the FPT_STM_EXT.1 SFR.)
- 0291: NIT technical decision for DH14 and FCS_CKM.1
- 0290: NIT technical decision for physical interruption of trusted path/channel
- 0289: NIT technical decision for FCS_TLSC_EXT.x.1 Test 5e
- 0281: NIT Technical Decision for Testing both thresholds for SSH rekey
- 0260: NIT Technical Decision for Typo in FCS_SSHS_EXT.1.4
- 0259: NIT Technical Decision for Support for X509 ssh rsa authentication IAW RFC 6187
- 0257: NIT Technical Decision for Updating FCS_DTLSC_EXT.x.2/FCS_TLSC_EXT.x.2 Tests 1-4
- 0228: NIT Technical Decision for CA certificates - basicConstraints validation

The following list of the NDcPP Technical Decisions do not apply to the TOE for the reasons defined next to the Technical Decision:

- 0323: NIT Technical Decision for DTLS server testing - Empty Certificate Authorities list – The TOE does not claim FCS_DTLSS_EXT.2 SFRs
- 0322: NIT Technical Decision for TLS server testing - Empty Certificate Authorities list – The TOE does not claim FCS_TLSS_EXT.2 SFRs
- 0262: NIT Technical Decision for TLS server testing - Empty Certificate Authorities list – The TOE does not claim FCS_TLSS_EXT.2 SFRs
- 0256: NIT Technical Decision for Handling of TLS connections with and without mutual authentication – The TOE does not claim FCS_DTLSC_EXT.2 nor FCS_TLSC_EXT.2 SFRs

## 3.7 Conformance Claim Rationale

The NDcPP states the following: "This is a Collaborative Protection Profile (cPP) whose Target of Evaluation (TOE) is a network device… A network device in the context of this cPP is a device composed of both hardware and software that is connected to the network and has an infrastructure within the network."

The TOE is a network device composed of hardware and software that is connected to the network and performs security monitoring of network traffic using IDS methods in order to detect cyber-attacks. Because the device is connected to and has an infrastructure within the network, this conformance claim is appropriate.

# 4 Security Problem Definition

## 4.1 Threats

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

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

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

**Table 8 TOE Threats**

## 4.2 Organizational Security Policies

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

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

**Table 9 TOE Organization Security Policies**

## 4.3 Assumptions

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

| Assumption | Assumption Definition |
|---|---|
| **A.PHYSICAL_PROTECTION** | The network device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security and/or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP will not include any requirements on physical tamper protection or other physical attack mitigations. The cPP will not expect the product to defend |

| | |
|---|---|
| | against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device. |
| **A.LIMITED_FUNCTIONALITY** | The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality). |
| **A.NO_THRU_TRAFFIC_PROTECTION** | A standard/generic network device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the network device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the network device, destined for another network entity, is not covered by the ND cPP. It is assumed that this protection will be covered by cPPs for particular types of network devices (e.g., firewall). |
| **A.TRUSTED_ADMINISTRATOR** | The Security Administrator(s) for the network device are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The network device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device. |
| **A.REGULAR_UPDATES** | The network device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities. |
| **A.ADMIN_CREDENTIALS_SECURE** | The Administrator's credentials (private key) used to access the network device are protected by the platform on which they reside. |
| **A.RESIDUAL_INFORMATION** | The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. |

**Table 10 TOE Assumptions**

## 4.4 Security Objectives

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

### 4.4.1 TOE Security Objectives

This section identifies the security objectives of the TOE. The NDcPP does not define any security objectives for the TOE.

### 4.4.2 Security Objectives for the Operational Environment

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

| Objective | Objective Definition |
|---|---|
| **OE.ADMIN_CREDENTIALS_SECURE** | The Administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside. |
| **OE.NO_GENERAL_PURPOSE** | There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. |
| **OE.NO_THRU_TRAFFIC_PROTECTION** | The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment. |
| **OE.PHYSICAL** | Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment. |
| **OE.TRUSTED_ADMIN** | Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. |
| **OE.UPDATES** | The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities. |
| **OE.RESIDUAL_INFORMATION** | The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. |

**Table 11 TOE Operational Environment Objectives**

## 4.5 Security Problem Definition Rationale

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

# 5 Extended Components Definition

## 5.1 Extended Security Functional Requirements

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

## 5.2 Extended Security Assurance Requirements

There are no extended Security Assurance Requirements in this ST.

# 6 Security Functional Requirements

## 6.1 Conventions

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

- **Assignment:** allows the specification of an identified parameter. Indicated with italicized text.
- **Refinement:** allows the addition of details. Indicated with bold text.
- **Selection:** allows the specification of one or more elements from a list. Indicated with underlined text.
- **Iteration:** allows a component to be used more than once with varying operations. Indicated with a sequential number in parentheses following the element number of the iterated SFR.

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

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

## 6.2 Security Functional Requirements Summary

The following table lists the SFRs claimed by the TOE:

| Class Name | Component Identification | Component Name |
|---|---|---|
| **Security Audit** | FAU_GEN.1 | Audit Data Generation |
| | FAU_GEN.2 | User Identity Association |
| | FAU_STG_EXT.1 | Protected Audit Event Storage |
| **Cryptographic Support (FCS)** | FCS_CKM.1 | Cryptographic Key Generation |
| | FCS_CKM.2 | Cryptographic Key Establishment |
| | FCS_CKM.4 | Cryptographic Key Destruction |
| | FCS_COP.1/DataEncryption | Cryptographic Operation (AES Data Encryption/Decryption) |
| | FCS_COP.1/SigGen | Cryptographic Operation (Signature Generation and Verification) |
| | FCS_COP.1/Hash | Cryptographic Operation (Hash Algorithm) |
| | FCS_COP.1/KeyedHash | Cryptographic Operation (Keyed Hash Algorithm) |
| | FCS_HTTPS_EXT.1 | HTTPS Protocol |
| | FCS_RBG_EXT.1 | Random Bit Generation |
| | FCS_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 |
| | FIA_AFL.1 | Authentication Failure Management |

| Class Name | Component Identification | Component Name |
|---|---|---|
| **Identification and Authentication (FIA)** | FIA_PMG_EXT.1 | Password Management |
| | FIA_UAU_EXT.2 | Password-Based Authentication Mechanism |
| | FIA_UAU.7 | Protected Authentication Feedback |
| | FIA_UIA_EXT.1 | User Identification and Authentication |
| | FIA_X509_EXT.1/Rev | X.509 Certificate Validation |
| | FIA_X509_EXT.2 | X.509 Certificate Authentication |
| | FIA_X509_EXT.3 | X.509 Certificate Requests |
| **Security Management (FMT)** | FMT_MOF.1/ManualUpdate | Management of Security Functions Behaviour |
| | FMT_MTD.1/CryptoKeys | Management of TSF Data |
| | FMT_MTD.1/CoreData | Management of TSF Data |
| | FMT_SMF.1 | Specification of Management Functions |
| | FMT_SMR.2 | Restrictions on Security Roles |
| **Protection of the TSF (FPT)** | FPT_APW_EXT.1 | Protection of Administrator Passwords |
| | FPT_SKP_EXT.1 | Protection of TSF Data (for Reading of All Pre-shared, Symmetric and Private Keys) |
| | FPT_STM_EXT.1 | Reliable Time Stamps |
| | FPT_TST_EXT.1 | TSF Testing |
| | FPT_TUD_EXT.1 | Trusted Update |
| **TOE Access (FTA)** | FTA_SSL_EXT.1 | TSF-Initiated Session Locking |
| | FTA_SSL.3 | TSF-Initiated Termination |
| | FTA_SSL.4 | User-Initiated Termination |
| | FTA_TAB.1 | Default TOE Access Banners |
| **Trusted Path/Channels (FTP)** | FTP_ITC.1 | Inter-TSF Trusted Channel |
| | FTP_TRP.1/Admin | Trusted Path |

**Table 12 Security Functional Requirements for the TOE**

## 6.3 Security Functional Requirements

### 6.3.1 Class FAU: Security Audit

#### 6.3.1.1 *FAU_GEN.1* *Audit Data Generation*

**FAU_GEN.1.1**

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

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) All administrative actions comprising:
   - Administrative login and logout (name of user account shall be logged if individual user accounts are required for administrators).
   - Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).
   - Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).

- Resetting passwords (name of related user account shall be logged).
- [no other actions];

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

**FAU_GEN.1.2**

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

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

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

| Requirement | Auditable Event(s) | Additional Audit Record Contents |
|---|---|---|
| **FAU_GEN.1** | None. | None. |
| **FAU_GEN.2** | None. | None. |
| **FAU_STG_EXT.1** | None. | None. |
| **FCS_CKM.1** | None. | None. |
| **FCS_CKM.2** | None. | None. |
| **FCS_CKM.4** | None. | None. |
| **FCS_COP.1/DataEncryption** | None. | None. |
| **FCS_COP.1/SigGen** | None. | None. |
| **FCS_COP.1/Hash** | None. | None. |
| **FCS_COP.1/KeyedHash** | None. | None. |
| **FCS_HTTPS_EXT.1** | Failure to establish a HTTPS Session. | Reason for failure. |
| **FCS_RBG_EXT.1** | None. | None. |
| **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_AFL.1** | Unsuccessful login attempts limit is met or exceeded. | Origin of the attempt (e.g., IP address). |
| **FIA_PMG_EXT.1** | None. | None. |
| **FIA_UAU_EXT.2** | All use of identification and authentication mechanism. | Origin of the attempt (e.g., IP address). |
| **FIA_UAU.7** | None. | None. |
| **FIA_UIA_EXT.1** | All use of identification and authentication mechanism. | Origin of the attempt (e.g., IP address). |
| **FIA_X509_EXT.1/Rev** | Unsuccessful attempt to validate a certificate. | Reason for failure. |
| **FIA_X509_EXT.2** | None. | None. |
| **FIA_X509_EXT.3** | None. | None. |
| **FMT_MOF.1/ManualUpdate** | Any attempt to initiate a manual update | None. |
| **FMT_MTD.1/CryptoKeys** | Management of cryptographic keys. | None. |
| **FMT_MTD.1/CoreData** | All management activities of TSF data. | None. |

| | | |
|---|---|---|
| **FMT_SMF.1** | None. | None. |
| **FMT_SMR.2** | None. | None. |
| **FPT_APW_EXT.1** | None. | None. |
| **FPT_SKP_EXT.1** | None. | None. |
| **FPT_STM_EXT.1** | Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1) | For discontinuous changes to time:<br>The old and new values for the time.<br>Origin of the attempt to change time for success and failure (e.g., IP address). |
| **FPT_TST_EXT.1** | None. | None. |
| **FPT_TUD_EXT.1** | Initiation of update; result of the update attempt (success or failure) | None. |
| **FTA_SSL_EXT.1** | The termination of a local session by the session locking mechanism. | None. |
| **FTA_SSL.3** | The termination of a remote session by the session locking mechanism. | None. |
| **FTA_SSL.4** | The termination of an interactive session. | None. |
| **FTA_TAB.1** | None. | None. |
| **FTP_ITC.1** | Initiation of the trusted channel.<br>Termination of the trusted channel.<br>Failure of the trusted channel functions. | Identification of the initiator and target of failed trusted channels establishment attempt. |
| **FTP_TRP.1/Admin** | Initiation of the trusted path.<br>Termination of the trusted path.<br>Failure of the trusted path functions. | None. |

**Table 13 Auditable Events**

### 6.3.1.2 *FAU_GEN.2* *User Identity Association*

**FAU_GEN.2.1**

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

### 6.3.1.3 *FAU_STG_EXT.1* *Protected Audit Event Storage*

**FAU_STG_EXT.1.1**

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

**FAU_STG_EXT.1.2**

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

**FAU_STG_EXT.1.3**

The TSF shall [overwrite previous audit records according to the following rule: [*overwrite oldest log file of that audit record type*]] when the local storage space for audit data is full.

### 6.3.2 Class FCS: Cryptographic Support

#### 6.3.2.1 *FCS_CKM.1* *Cryptographic Key Generation*

**FCS_CKM.1.1**

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

- RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;
- FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1
- FFC schemes using Diffie-Hellman group 14 that meet the following: RFC 3526, Section 3

].

#### 6.3.2.2 *FCS_CKM.2* *Cryptographic Key Establishment*

**FCS_CKM.2.1**

The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [

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

].

#### 6.3.2.3 *FCS_CKM.4* *Cryptographic Key Destruction*

**FCS_CKM.4.1**

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

- For plaintext keys in volatile storage, the destruction shall be executed by a [single overwrite consisting of [zeroes]];
- For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [
  - logically addresses the storage location of the key and performs a [[*3*]-pass] overwrite consisting of [zeroes, ones]]

that meets the following: No Standard.

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

**FCS_COP.1.1/DataEncryption**

The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in [CBC] mode and cryptographic key sizes [128 bits, 256 bits] that meet the following: AES as specified in ISO 18033-3, [CBC as specified in ISO 10116].

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

**FCS_COP.1.1/SigGen**

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

[

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

that meet the following:

[

- For RSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3].

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

**FCS_COP.1.1/Hash**

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

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

**FCS_COP.1.1/KeyedHash**

The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm [HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-512] and cryptographic key sizes [*160 bits, 256 bits, 512 bits*] and message digest sizes [160, 256, 512] bits that meet the following: ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”.

### 6.3.2.8 *FCS_HTTPS_EXT.1 HTTPS Protocol*

**FCS_HTTPS_EXT.1.1**

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

**FCS_HTTPS_EXT.1.2**

The TSF shall implement HTTPS using TLS.

**FCS_HTTPS_EXT.1.3**

If a peer certificate is presented, the TSF shall [not establish the connection] if the peer certificate is deemed invalid.

### 6.3.2.9 *FCS_RBG_EXT.1* *Random Bit Generation*

**FCS_RBG_EXT.1.1**

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

**FCS_RBG_EXT.1.2**

The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [[5] software-based noise source] with a minimum of [256 bits] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and hashes that it will generate.

### 6.3.2.10 *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, 5656, 6668].

**FCS_SSHC_EXT.1.2**

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

**FCS_SSHC_EXT.1.3**

The TSF shall ensure that, as described in RFC 4253, packets greater than [*262144*] 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].

**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-sha2-256, hmac-sha2-512] and [no other MAC algorithms] 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] and [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 the same session keys are used for a threshold of no longer than one hour, and no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

**FCS_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.

#### 6.3.2.11 *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, 5656, 6668].

**FCS_SSHS_EXT.1.2**

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

**FCS_SSHS_EXT.1.3**

The TSF shall ensure that, as described in RFC 4253, packets greater than [*262144*] 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].

**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-sha2-256, hmac-sha2-512] and [no other MAC algorithms] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

**FCS_SSHS_EXT.1.7**

The TSF shall ensure that [diffie-hellman-group14-sha1] and [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 the same session keys are used for a threshold of no longer than one hour, and no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

#### 6.3.2.12 *FCS_TLSC_EXT.1* *TLS Client Protocol*

**FCS_TLSC_EXT.1.1**

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

[

- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268

].

**FCS_TLSC_EXT.1.2**

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

**FCS_TLSC_EXT.1.3**

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

**FCS_TLSC_EXT.1.4**

The TSF shall [not present the Supported Elliptic Curves Extension] in the Client Hello.

#### 6.3.2.13 *FCS_TLSS_EXT.1* *TLS Server Protocol*

**FCS_TLSS_EXT.1.1**

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

[

- TLS_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, TLS 1.0, and [TLS 1.1].

**FCS_TLSS_EXT.1.3**

The TSF shall [generate Diffie-Hellman parameters of size [2048 bits]].

### 6.3.3 Class FIA: Identification and Authentication

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

**FIA_AFL.1.1**

The TSF shall detect when an Administrator configurable positive integer within [*1 to 20*] unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely.

**FIA_AFL.1.2**

When the defined number of unsuccessful authentication attempts has been met, the TSF shall [prevent the offending remote Administrator from successfully authenticating until [*invoking an account unlocking command*] is taken by a local Administrator].

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

**FIA_PMG_EXT.1.1**

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

1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”];
2. Minimum password length shall be configurable to [*15 characters*] and [*100 characters*].

### 6.3.3.3 ***FIA_UAU_EXT.2*** ***Password-Based Authentication Mechanism***

**FIA_UAU_EXT.2.1**

The TSF shall provide a local password-based authentication mechanism, and [[*SSH public key-based, remote password-based authentication via LDAP*]] to perform local administrative user authentication.

### 6.3.3.4 ***FIA_UAU.7*** ***Protected Authentication Feedback***

**FIA_UAU.7.1**

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

### 6.3.3.5 ***FIA_UIA_EXT.1*** ***User Identification and Authentication***

**FIA_UIA_EXT.1.1**

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

- Display the warning banner in accordance with FTA_TAB.1;
- [[*Forgot Username/Password feature, root level log messages*]]

**FIA_UIA_EXT.1.2**

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

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

**FIA_X509_EXT.1.1/Rev**

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

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

**FIA_X509_EXT.1.2/Rev**

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

#### 6.3.3.7 *FIA_X509_EXT.2* *X.509 Certificate Authentication*

**FIA_X509_EXT.2.1**

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

**FIA_X509_EXT.2.2**

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

#### 6.3.3.8 *FIA_X509_EXT.3* *X.509 Certificate Requests*

**FIA_X509_EXT.3.1**

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

**FIA_X509_EXT.3.2**

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

### 6.3.4 Class FMT: Security Management

#### 6.3.4.1 *FMT_MOF.1/ManualUpdate* *Management of Security Functions Behaviour*

**FMT_MOF.1.1/ManualUpdate**

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

#### 6.3.4.2 *FMT_MTD.1/CryptoKeys* *Management of TSF Data*

**FMT_MTD.1.1/CryptoKeys**

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

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

**FMT_MTD.1.1/CoreData**

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

#### 6.3.4.4 *FMT_SMF.1* *Specification of Management Functions*

**FMT_SMF.1.1**

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

- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using [hash comparison] capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- [
    - Ability to configure the cryptographic functionality;
    - Ability to re-enable an Administrator account;
    - Ability to set the time which is used for time-stamps;
    - Ability to configure the reference identifier for the peer]

#### 6.3.4.5 *FMT_SMR.2* *Restrictions on Security Roles*

**FMT_SMR.2.1**

The TSF shall maintain the roles:

- Security Administrator.

**FMT_SMR.2.2**

The TSF shall be able to associate users with roles.

**FMT_SMR.2.3**

The TSF shall ensure that the conditions

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

are satisfied.

### 6.3.5 Class FPT: Protection of the TSF

#### 6.3.5.1 *FPT_APW_EXT.1 Protection of Administrator Passwords*

**FPT_APW_EXT.1.1**

The TSF shall store passwords in non-plaintext form.

**FPT_APW_EXT.1.2**

The TSF shall prevent the reading of plaintext passwords.

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

**FPT_SKP_EXT.1.1**

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

#### 6.3.5.3 *FPT_STM_EXT.1 Reliable Time Stamps*

**FPT_STM_EXT.1.1**

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

**FPT_STM_EXT.1.2**

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

#### 6.3.5.4 *FPT_TST_EXT.1 TSF Testing*

**FPT_TST_EXT.1.1**

The TSF shall run a suite of the following self-tests [during initial start-up (on power on)] to demonstrate the correct operation of the TSF: [*Cryptographic Module Integrity Pairwise Consistency and Known Answer Tests; CPU General Purpose, Floating Point Unit, Extension, and Prime Number Calculation Instruction Tests and Cooling System tests; and RAM and Disk Subsystem Read/Write Validation Tests*].

#### 6.3.5.5 *FPT_TUD_EXT.1 Trusted Update*

**FPT_TUD_EXT.1.1**

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

**FPT_TUD_EXT.1.2**

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

**FPT_TUD_EXT.1.3**

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

## 6.3.6 Class FTA: TOE Access

### 6.3.6.1 *FTA_SSL_EXT.1* *TSF-initiated Session Locking*

**FTA_SSL_EXT.1.1**

The TSF shall, for local interactive sessions, [terminate the session] after a Security Administrator-specified time period of inactivity.

### 6.3.6.2 *FTA_SSL.3* *TSF-initiated Termination*

**FTA_SSL.3.1**

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

### 6.3.6.3 *FTA_SSL.4* *User-initiated Termination*

**FTA_SSL.4.1**

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

### 6.3.6.4 *FTA_TAB.1* *Default TOE Access Banner*

**FTA_TAB.1.1**

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

## 6.3.7 Class FTP: Trusted Path/Channels

### 6.3.7.1 *FTP_ITC.1* *Inter-TSF Trusted Channel*

**FTP_ITC.1.1**

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

**FTP_ITC.1.2**

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

**FTP_ITC.1.3**

The TSF shall initiate communication via the trusted channel for [*audit transfer, authentication requests, software image updates, policy updates, network event data (metadata), Forgot Username/Password email*].

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

**FTP_TRP.1.1/Admin**

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

**FTP_TRP.1.2/Admin**

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

**FTP_TRP.1.3/Admin**

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

## 6.4 Statement of Security Functional Requirements Consistency

The Security Functional Requirements included in the ST represent all required SFRs specified in the cPP against which exact conformance is claimed and a subset of the optional SFRs. All hierarchical relationships, dependencies, and unfulfilled dependency rationales in the ST are considered to be identical to those that are defined in the claimed PP.

# 7 Security Assurance Requirements

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

## 7.1 Class ADV: Development

### 7.1.1 Basic Functional Specification (ADV_FSP.1)

#### 7.1.1.1 *Developer action elements:*

**ADV_FSP.1.1D**

The developer shall provide a functional specification.

**ADV_FSP.1.2D**

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

#### 7.1.1.2 *Content and presentation elements:*

**ADV_FSP.1.1C**

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

**ADV_FSP.1.2C**

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

**ADV_FSP.1.3C**

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

**ADV_FSP.1.4C**

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

#### 7.1.1.3 *Evaluator action elements:*

**ADV_ FSP.1.1E**

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

**ADV_ FSP.1.2E**

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

## 7.2 Class AGD: Guidance Documentation

### 7.2.1 Operational User Guidance (AGD_OPE.1)

#### 7.2.1.1 *Developer action elements:*

**AGD_OPE.1.1D**

The developer shall provide operational user guidance.

#### 7.2.1.2 *Content and presentation elements:*

**AGD_OPE.1.1C**

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

**AGD_OPE.1.2C**

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

**AGD_OPE.1.3C**

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

**AGD_OPE.1.4C**

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

**AGD_OPE.1.5C**

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

**AGD_OPE.1.6C**

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

**AGD_OPE.1.7C**

The operational user guidance shall be clear and reasonable.

#### 7.2.1.3 *Evaluator action elements:*

**AGD_OPE.1.1E**

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

### 7.2.2 Preparative Procedures (AGD_PRE.1)

#### 7.2.2.1 *Developer action elements:*

**AGD_PRE.1.1D**

The developer shall provide the TOE including its preparative procedures.

#### 7.2.2.2 *Content and presentation elements:*

**AGD_ PRE.1.1C**

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

**AGD_ PRE.1.2C**

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

#### 7.2.2.3 *Evaluator action elements:*

**AGD_ PRE.1.1E**

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

**AGD_ PRE.1.2E**

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

## 7.3 Class ALC: Life Cycle Supports

### 7.3.1 Labeling of the TOE (ALC_CMC.1)

#### 7.3.1.1 *Developer action elements:*

**ALC_CMC.1.1D**

The developer shall provide the TOE and a reference for the TOE.

#### 7.3.1.2 *Content and presentation elements:*

**ALC_CMC.1.1C**

The TOE shall be labeled with its unique reference.

#### 7.3.1.3 *Evaluator action elements:*

**ALC_CMC.1.1E**

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

### 7.3.2 TOE CM Coverage (ALC_CMS.1)

#### 7.3.2.1 *Developer action elements:*

**ALC_CMS.1.1D**

The developer shall provide a configuration list for the TOE.

#### 7.3.2.2 *Content and presentation elements:*

**ALC_CMS.1.1C**

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

**ALC_CMS.1.2C**

The configuration list shall uniquely identify the configuration items.

#### 7.3.2.3 *Evaluator action elements:*

**ALC_CMS.1.1E**

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

## 7.4 Class ATE: Tests

### 7.4.1 Independent Testing - Conformance (ATE_IND.1)

#### 7.4.1.1 *Developer action elements:*

**ATE_IND.1.1D**

The developer shall provide the TOE for testing.

#### 7.4.1.2 *Content and presentation elements:*

**ATE_IND.1.1C**

The TOE shall be suitable for testing.

#### 7.4.1.3 *Evaluator action elements:*

**ATE_IND.1.1E**

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

**ATE_IND.1.2E**

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

## 7.5 Class AVA: Vulnerability Assessment

### 7.5.1 Vulnerability Survey (AVA_VAN.1)

#### 7.5.1.1 *Developer action elements:*

**AVA_VAN.1.1D**

The developer shall provide the TOE for testing.

#### 7.5.1.2 *Content and presentation elements:*

**AVA_VAN.1.1C**

The TOE shall be suitable for testing.

#### 7.5.1.3 *Evaluator action elements:*

**AVA_VAN.1.1E**

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

**AVA_VAN.1.2E**

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

**AVA_VAN.1.3E**

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

# 8 TOE Summary Specification

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

## 8.1 Security Audit

### 8.1.1 FAU_GEN.1

The TOE contains mechanisms which generate audit data based upon successful and unsuccessful management actions by all authorized users of the TOE. The startup and shutdown of the TOE's audit functionality is synonymous with the startup and shutdown of the TOE, which is recorded in the TOE's audit records. In the evaluated configuration, the audit functions of the TOE cannot be turned on or off except for TOE reboot. Each audit record contains identifying information including the date and time the event occurred, the type of event, the subject identity of the event, and the outcome of the event. The audit records are generated and stored in the form of syslog records which are sent securely to the Syslog Server protected by TLS for the activities logs. VAR logs that are generated from applications are also transferred via TLS to the Syslog Server for remote storage.

Users with the appropriate permissions can view audit log files, however, only Administrator users can delete audit log files. If the Security Administrator deletes a log file, an audit record of that action is also recorded.

All actions performed on the TOE are logged. These include the following auditable events defined in Table 13 and FAU_GEN.1:

- Failure to establish an HTTPS session
- Failure to establish an SSH session.
- Failure to establish a TLS Session
- Unsuccessful login attempts limit is met or exceeded.
- All use of identification and authentication mechanism
- Unsuccessful attempt to validate a certificate.
- Any attempt to initiate a manual update
- Management of cryptographic keys.
- All management activities of TSF data.
- Discontinuous changes to time
- Initiation of update
- Result of the update attempt (success or failure)
- The termination of a local session by the session locking mechanism
- The termination of a remote session by the session locking mechanism
- The termination of an interactive session
- Initiation of the trusted channel
- Termination of the trusted channel
- Failure of the trusted channel functions
- Initiation of the trusted path

- Termination of the trusted path
- Failure of the trusted path functions
- Start-up and shut-down of the audit functions
- Administrative login and logout
- Changes to TSF data related to configuration changes
- Generating/import of, changing, or deleting of cryptographic keys
- Resetting passwords

Audit records are created when the administrator performs each of the management functions listed above via the web GUI and the CLI (local and remote) or when an external operational environment component is connecting with the TOE. Each audit record provides a timestamp, subject identity, defines the type of event, and identifies if the event was successful or failed. The TOE also records additional information as specified by the right column of Table 13 and under bullet c) of the FAU_GEN.1.1 SFR. For example, the audit record will contain the fingerprint and/or filename to identify the key for the administrative task of generating a cryptographic key and the audit record will contain the filename to identify the key for the administrative tasks of importing, changing, or deleting a cryptographic key.

### 8.1.2 FAU_GEN.2

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

Username=admin,

System name= Process crond (Cron Daemon),

IP address= 10.115.0.108.

### 8.1.3 FAU_STG_EXT.1

The TOE keeps audit records for all auditable events related to the web GUI and CLI management actions. These audit records are stored locally in the /var/log directory. When the current log file reaches its allowed maximum size, it is closed and renamed sequentially (e.g. log1, log2, etc.) and a new log file is opened as the current log. Once the local log file reaches 6, the oldest file is deleted to provide space for the new files. In addition, many applications run from the CLI keep their own VAR log files, such as Apache, LDAP client, etc. Both sets of logs are automatically transferred remotely to a Syslog Server over a TLS channel in real-time. The TOE automatically manages all audit file types in the same manner whereas if the storage space allocated for an individual file type is filled the TOE will overwrite the oldest log file of that type.

The maximum size of each log file, the manner of rotation (deletes the oldest log file), and the maximum number of log files for each log file type are all configurable via the CLI by the VCR user. These configurations are managed through log4j (java), newsyslog (NIKOS), logrotate (Linux) and syslog-ng (Syslog).

The VCR user is the only user that has access to the CLI and as a Security Administrator is expected to operate as a trusted administrator. Thus, all audit records stored on the TOE are protected by the TOE’s authentication mechanisms for the VCR user and the VCR user is not expected to modify or delete the audit records for malicious purposes.

## 8.2 Cryptographic Support

### 8.2.1 FCS_CKM.1

The TOE implements a FIPS PUB 186-4 conformant key generation mechanism for RSA key generation schemes for generating X.509 certificates. Specifically, the TOE complies with the FIPS 186-4 (Digital Signature Standard (DSS) Appendix B.3). This is used to generate the RSA key pairs with a modulus of at least 2048 bits which has an equivalent key strength of 112 bits.

The TOE implements a FIPS PUB 186-4 conformant key generation mechanism for DSA to perform FFC key generation of 2048-bit keys in support of TLS. Specifically, the TOE complies with the FIPS 186-4 (Digital Signature Standard (DSS) Appendix B.1).

The TOE implements Diffie Hellman group 14 (RFC 3526, Section 3) key establishment methods for SSH.

The TOE's key generation functions have the following CAVP certificates:

RSA: #2902

DSA: #1394

### 8.2.2 FCS_CKM.2

The TOE implements a NIST SP 800-56A conformant key establishment mechanism for Diffie-Hellman key establishment schemes in support of TLS as both a sender and a recipient. Specifically, the TOE complies with the NIST SP 800-56A Key Agreement Scheme (KAS) without a Key Derivation Function (KDF) which is defined in section 5.6 of the SP. This requirement is met by the Component Validation List (CVL) certificate #1871.

The TOE uses Diffie-Hellman-group14-SHA1 in accordance with RFC 3526, Section 3 in support of SSH. The TOE implements and uses the prime and generator specified in RFC 3526 Section 3 when generating parameters for the key exchange.

### 8.2.3 FCS_CKM.4

The Diffie-Hellman Shared Secret, Diffie Hellman private exponent, and SSH session key are generated by the TOE and stored in volatile memory (RAM). These keys are destroyed by a single direct overwrite consisting of zeroes. These keys are zeroized immediately after they are no longer needed (i.e. connection terminated or re-key) and when the TOE is shut down as well as when power is lost.

The SSH private key and SSL server key are stored on the local filesystem and RAM. When stored in RAM, these keys will be zeroized in the same manner as the other keys stored in volatile memory. In the evaluated configuration, these keys are generated by the TOE. The SSH private key is stored in /root/.ssh/id_rsa and the SSL server key is stored in /usr/local/etc/apache24/server.key. The VCR user has the ability to delete SSH private key and SSL server key using the rm –P <keyfile>  command via the Console or remote CLI. The VCR user would perform this action before generating a new key. When this command is used, the TOE overwrites the location where the keys are stored with three overwrite passes with the byte pattern of 0xff (i.e. ones), followed by 0x00 (i.e. zeroes), and followed by 0xff (i.e. ones) again.

### 8.2.4 FCS_COP.1/DataEncryption

The TOE performs encryption and decryption using the AES algorithm in CBC mode with key sizes of 128 and 256 bits. This algorithm has CAVP AES certificate #5418. The AES algorithm meets ISO 18033-3 and CBC meets ISO 10116.

### 8.2.5 FCS_COP.1/SigGen

The TOE performs cryptographic digital signature verification and generation in accordance with FIPS PUB 186-4: RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater.

The algorithm has CAVP RSA certificate #2902.

### 8.2.6 FCS_COP.1/Hash

The TOE provides cryptographic hashing services using SHA-1, SHA-256 and SHA-512 with message digest sizes of 160, 256 and 512 bits respectively, as specified in FIPS PUB 180-4. The TSF also uses SHA-1, SHA-256 and SHA-512 for HMAC for message authentication, health tests, TLS certificate authentication and SSH. SHA-256 and SHA-512 are used in RSA for Signature Generation and Signature Verification. The SHA algorithm meets ISO/IEC 10118-3:2004 and has CAVP SHS certificate #4349.

### 8.2.7 FCS_COP.1/KeyedHash

The TOE provides keyed-hashing message authentication services using HMAC-SHA-1, HMAC-SHA-256, and HMAC-SHA-512 with key sizes (less than, greater than or equal to block size) and digest sizes of 160, 256, and 512 bits as specified in FIPS PUB 198-1 and FIPS PUB 180-4. The algorithm meets ISO/IEC 9797-2:2011 and has CAVP HMAC certificate #3588.

### 8.2.8 FCS_HTTPS_EXT.1

The TOE invokes HTTPS in compliance with RFC 2818 to provide a secure interactive management interface via the web GUI. If the certificate from the TOE is invalid, the browser will inform the user and let them decide whether to proceed with the connection. HTTPS is also used to facilitate a secure exchange of information and status reports between the TOE and NetOmni. NetOmni will initiate the connection and it will only be established if the peer certificate provided by the TOE to NetOmni is valid. A 2048-bit RSA peer certificate is used by the HTTPS/TLS protocol.

HTTPS uses TLSv1.2 (as specified by FCS_TLSS_EXT.1) to securely establish the AES encrypted session which uses the TLS_DHE_RSA_WITH_AES_256_CBC_SHA ciphersuite. Port 443 is used for initial HTTPS connections in accordance with RFC 2818.

### 8.2.9 FCS_RBG_EXT.1

The TOE performs random bit generation services in accordance with ISO/IEC 18031:2011 by CTR_DRBG (AES) (CAVP cert #2113). The TOE is seeded with at least 256 bits of entropy from /dev/random which is seeded with unconditioned random data from 5 software entropy sources: RANDOM_KEYBOARD, RANDOM_INTERRUPT, RANDOM_MOUSE, RANDOM_NET_ETHER and RANDOM_SWI.

### 8.2.10 FCS_SSHC_EXT.1 and FCS_SSHS_EXT.1

The TOE acts as an SSH server for remote CLI management. The TOE acts as an SSH Client when SCP is used to transfer the software image updates to the TOE.

Regardless of whether the TOE is acting as a client or a server, the SSH functionality is compliant with RFCs 4251, 4252, 4253, 4254, 5656, and 6668. The TOE supports password based and public key based authentication using an RSA key of 2048 bits in length as described in RFC 4252, using ssh-rsa as its public key authentication algorithm. Encryption is provided by aes128-cbc and aes256-cbc, with data integrity MAC algorithms hmac-sha2-256 and hmac-sha2-512, and diffie-hellman-group14-sha1 as its key exchange algorithm. The SSH connection will drop any connection when a packet greater than 262144 bytes is detected, in accordance with RFC 4253. The SSH connection will rekey before 1 hour has elapsed or 1 GB of data has been transmitted using that key, whichever occurs first.

In addition, whether acting as a server or a client, public key based authentication is achieved for SSH and SCP by generating the private/public key pair of size 2048 bits with the command

```
ssh-keygen –t rsa;
```

which creates the keys in the directory ~/.ssh under id_rsa and id_rsa.pub for the private and public keys respectively. Copying the public key to remote servers and appending ~/.ssh/authorized_keys will allow for public key authentication.

When acting as a client, the TOE uses a local database located in ~/.ssh/authorized_keys to associate SSH host names with their corresponding public key to validate their identity. This is in conformance with RFC 4251 section 4.1.

### 8.2.11 FCS_TLSC_EXT.1

The TOE uses the TLSv1.2 protocol and TLS_DHE_RSA_WITH_AES_256_CBC_SHA ciphersuite to secure the channel for the following purposes with these servers in the operational environment:

- sending of audit data to the Syslog Server,
- performing authentication requests with the LDAP/AD Server, and
- sending “Forgot Username/Password” emails to an SMTP Server.

Configuring these channels requires the Security Administrator to define the reference identifier of the operational environment servers to which the TOE will connect. The TOE supports the DNS hostname only and does not support IP addresses. Wildcards cannot be defined as part of the DNS hostname on the TOE but the TOE will accept certificates with wildcards specified. As part of the TLS session establishment, the TOE will validate the 2048-bit certificate received from the operational environment server and will only establish the connection if the certificate is valid. The TOE will also verify the identity of the Syslog Server, LDAP/AD Server and SMTP Server in accordance with RFC 6125 by checking that the presented identifier from the certificate, which includes the Common Name and DNS Name (Subject Alternative Name), matches the reference identifier (i.e. DNS hostname) defined on the TOE. The TOE does not support certificate pinning or Elliptic Curves.

### 8.2.12 FCS_TLSS_EXT.1

The TOE and the NetOmni perform HTTPS/TLS server side authentication using X.509v3 certificates. When NetOmni attempts to connect to the TOE, the TOE will present its X.509v3 server certificate to NetOmni. NetOmni will then validate the certificate and confirm the TOE's identity before establishing the connection. Remote user administration via the web GUI is also protected using HTTPS/TLS. The TOE uses the TLSv1.2 protocol and TLS_DHE_RSA_WITH_AES_256_CBC_SHA ciphersuite to secure the NetOmni channel and web GUI path to the TOE. The TSF denies all connections from clients requesting connections dependent on the following SSL 2.0, SSL 3.0, TLS 1.0, and TLS 1.1 protocols and any other ciphersuite. The TSF uses keys that are established using 2048-bit Diffie-Hellman.

## 8.3 Identification and Authentication

### 8.3.1 FIA_AFL.1

In the evaluated configuration, the TOE will lock a remote administrative account when an administrator configured number of successive invalid login attempts have been made within an administrator configured time period. This applies to both the GUI and remote CLI interfaces, and the default values for the failed attempts and time period settings are 3 successive unsuccessful attempts within 15 minutes. In the evaluated configuration, the number of failed attempts can be set to a positive integer between 1 and 20 attempts. These settings can be configured by the VCR user via the console (local CLI) or remote CLI by modifying the following files:

- GUI: /usr/local/niksun/apps/etc/apps.conf

```
#no of failed attempts before disabling the account default=3
failed_attempts=3
#time to disable accounts for the above count failed_attempts (in mins)
#default=15
time_failed_attempts=15
```

- CLI: /etc/pam.d/sshd

```
auth required pam_exec.so return_prog_exit_status --
/usr/local/niksun/apps/bin/niksun_te 3 900
```

  Note: The 3 is number of failed attempts, and the 900 is number of seconds

The TOE maintains a counter per username for the number of failed authentication attempts and tracks the time when each failed authentication attempt occurs. If a valid password is provided before the failed attempt value is met, then authentication is granted and the counter resets to zero. When a failed authentication attempt is older than the set time period and the counter has not met the failed attempt value, the counter will be reduced by one failed attempt. If the limit of failed authentication attempts is reached within the defined time period, the account associated with the username will be locked. Once an account is locked, repeated attempts to authenticate with that account will result in displaying the following error message:

- GUI: Account Disabled!
- Remote CLI: `PAM: maximum number of tries exceeded for <account> from <IP address>`

Once an account is locked, the VCR user via the Console (local CLI) must unlock the account using the following commands before another authentication attempt will be checked for that account:

- GUI: `/usr/local/niksun/apps/bin/manage_pass enable <account>`
- CLI: `pw unlock <account>`

The root account via the Console (local CLI) is not subject to authentication failures and thus, authentication failures by remote administrators cannot lead to a situation that prevents all administration of the TOE.

### 8.3.2 **FIA_PMG_EXT.1**

In the evaluated configuration, the TOE enforces passwords to have minimum length between 15 and 100 characters. The accepted characters include upper and lower case letters, numbers, and the special characters "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")". The users are divided into two separate groups by how they access the TOE, whether through an available web GUI or the CLI. For the web GUI, the Administrators have the ability to set the minimum password length for GUI passwords in order to match the evaluated configuration. For the CLI the VCR user has the ability to set the minimum character limit for CLI passwords in the /etc/pam.d/passwd file.

### 8.3.3 **FIA_UAU_EXT.2**

Users can authenticate to the TOE locally or remotely. Local users can gain access to the TOE via the console (local CLI) by authenticating to the TOE's local authentication mechanism with their username/password combination. Remote users can gain access to the TOE by either the remote CLI or the web GUI. The remote CLI is protected by SSH and allows users to authenticate against the TOE's local authentication mechanisms with either their username/password combination or SSH public key.

SSH public key authentication can be achieved for SSH and SCP sessions with the following steps:

- `ssh-keygen –t rsa` – This command generates a private/ public key pair in `~/.ssh` in files `id_rsa` and `id_rsa.pub` for private and public keys respectively.
- Copy the public key to a remote serve and append it to `~/.ssh/authorized_keys`.

The web GUI is protected by HTTPS/TLS and allows users to authenticate with their username/password combination against the TOE's local authentication mechanism or a remote LDAP/AD Server. When validating user's credentials stored in the remote LDAP/AD Server, the TOE will send a verification request to the LDAP/AD Server with the user's entered username and password and the LDAP/AD Server will respond with pass or failed authentication. If authentication passes, the validated username is used to determine the user's assigned role in the TOE's local user store. In the evaluated configuration, the TOE connects to a server with OpenLDAP using LDAPS protected by TLS.

### 8.3.4 **FIA_UAU.7**

While authenticating to the TOE with an incorrect login (specifically an invalid username and/or an invalid password) on any interface the TOE does not indicate whether the username or password was incorrect. Also, there is no feedback while a user is entering their password via the console (local CLI), using the monitor and keyboard.

### 8.3.5 FIA_UIA_EXT.1

Users can connect to the TOE via the CLI or the web GUI. The CLI can be accessed remotely through an SSH client or locally on the console. Prior to authentication, the TOE displays the standard Linux pre-authentication banner on the CLI; which can be configured by the VCR user through the CLI. Additionally, the console will display root level log messages prior to authentication. The display of the warning banner and root level log messages (console only) are the only pre-authentication services provided by the TOE via the CLI.

The web GUI can be accessed through a web browser and is protected by HTTPS/TLS. The pre-authentication services that the TOE allows via the web GUI are displaying the warning banner and a “Forgot Username/Password” feature. In the evaluated configuration, an administrator will configure the secure mode for the web GUI’s “Forgot Username/Password” feature which will enable TLS for the Mail Transfer Agent (MTA). When a user utilizes the “Forgot Username/Password” feature on the web GUI login screen, NetDetector’s MTA will send an email to the SMTP Server over a protected TLS channel. If the user forgot their username, the user can enter the email associated with the username and the TOE will send an email using SMTP to that email account. If the user has forgotten the password, the user can enter their username and the TOE will send an email using SMTP to the email address associated with that username. The email will contain a link that directs the user to the TOE’s web GUI to be able to securely change their password.

### 8.3.6 FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, and FIA_X509_EXT.3

The TOE checks the validity of a server’s certificate for the following connections: when it connects to a Syslog Server for sending audit data over TLS, when it connects with an LDAP/AD Server for authenticating users over TLS, and when it connects to an SMTP Server to send emails over TLS.

The TSF determines the validity of certificates by ensuring that the certificate and the certificate path is valid in accordance with RFC 5280. In addition, the certificate path must terminate in a trusted CA certificate, the basicConstraints extension is present, and the CA flag is set to TRUE for all CA certificates. The TOE will only consider a certificate as a CA certificate if both the basicConstraint extension is present and the CA flag is set to TRUE. The TSF also validates the revocation status of the certificate by using a CRL in accordance with RFC 5280 Section 6.3. Finally, the TOE ensures the extendedKeyUsage field includes the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) for server certificates used in TLS.

The TOE uses X.509v3 certificates to support authentication for TLS connections in accordance with RFC 5280. The TOE requests CRLs to check the revocation status of certificates provided in a certificate chain. CRLs are requested using HTTP from a CRL distribution point which resides in the operating environment. It is expected that the CRL distribution point has the same physical controls and security provided to the TOE. When the TOE cannot establish a connection to the CRL distribution point to determine the validity of a certificate, the TSF rejects the certificate.

The TOE and the NetOmni perform HTTPS/TLS server side authentication using X.509v3 certificates. When NetOmni attempts to connect to the TOE, the TOE will present its X.509v3 server certificate to NetOmni. The TOE can create a Certificate Signing Request (CSR) as specified by RFC 2986. The request includes the following information: Public Key, Distinguished Name, Country, Organization Name, and Organizational Unit. Once created, the CSR can be manually transferred to the CA for signature and then manually transferred back. When the TSF receives the CA Certificate Response, it will validate the chain of certificates from the Root CA upon receiving CA Certificate Response.

## 8.4 Security Management

### 8.4.1 FMT_MOF.1/ManualUpdate

The TOE restricts the ability to perform manual updates to the VCR user via the CLI. There are no other methods for updating the TOE.

### 8.4.2 FMT_MTD.1/CryptoKeys

The ability to modify, delete, and generate/import cryptographic keys is limited to the VCR user through the CLI.

### 8.4.3 FMT_MTD.1/CoreData

The TOE has several privileges which it can grant to user groups for the web GUI interface and only one user function for the CLI.

The CLI user function, called the VCR user, provides the majority of support for and modification of the TOE security functions, and acts as one of the main Security Administrators. It also assumes the role of root in order to perform some security functions.

The web GUI has users, which can belong to at most one group each. Groups map to permissions that determine the actions authorized for the members of a group. The Admin user (default user account) belongs to the Administrators Group and serves as the Security Administrator. The Admin user cannot change its name and the Administrators Group cannot have its permissions changed.

The only administrative actions allowed before authentication is the use of the “forgot password” function for the web GUI, the ability to view root level log messages via the console (local CLI) and the display of the security banner for the web GUI and the CLI.

### 8.4.4 FMT_SMF.1

The TOE has two types of users, those that access the TOE via the CLI, and those by the web GUI. The CLI allows management of the TOE remotely and locally, while the web GUI allows only remote management. The role of administrator for the CLI is fulfilled by the VCR, while for the web GUI it is fulfilled by the Administrators Group, with the user named Admin being the original administrator. The Administrator users are capable of performing the following management functions on the TOE as defined elsewhere in this document:

- Ability to administer the TOE locally (Console) and remotely (GUI and remote CLI);
- Ability to configure the GUI access banner via the GUI and the CLI access banner via the Console or remote CLI;
- Ability to configure the session inactivity time before session termination via the Console or remote CLI;

- Ability to update the TOE, and to verify the updates using a published hash prior to installing those updates via the Console or remote CLI;
- Ability to configure the authentication failure parameters for FIA_AFL.1 and re-enable a locked Administrator account via the Console or remote CLI;
- Ability to configure cryptographic functionality including the SSH and TLS protocols and their connections, and generate SSH public keys and CSRs via the Console or remote CLI;
- Ability to set the time which is used for time-stamps via the Console and remote CLI;
- Ability to configure the reference identifier for the peer via the GUI, Console, and remote CLI.

### 8.4.5 FMT_SMR.2

There are two types of user accounts, those that access the TOE through the CLI, and those that access through the web GUI. The TOE maintains the role of Security Administrator which is fulfilled by the VCR user for the CLI and the Administrator users for the web GUI. For the CLI interface, the VCR user is the only user in the evaluated configuration. The CLI can be accessed locally through a keyboard and terminal or remotely through an SSH session. The VCR user sometimes assumes the role of root for some management activities for the TOE. All web GUI security management functions available to authorized users of the TOE are mediated by a role-based access control system.

Each user has the following security attributes associated with them:

CLI users (standard Unix attributes):

- Username
- Password (SHA512)
- Full name (optional)
- SSH public key for remote CLI login
- User groups (note -- users can be in more than 1 group, including TSF management)
- User home directory
- User default shell
- Last successful authentication time
- Number of failed authentications
- Status (locked or unlocked)

Web GUI users:

- Username
- Password (SHA256)
- Full name
- Description (optional)
- Email
- Phone (optional)
- User group (note -- users can only be in 1 group)
- User rights (note -- ACL defined by group)
- Password policy (note -- defined by group)

- Status (enabled or disabled)
- Last password change date
- Password expiration date
- Password expiration notification policy (how many days before expiration for reminders)
- Password age
- Old passwords (SHA256 list)
- First authentication failure time
- Last authentication failure time
- Number of failed authentications

The TOE will store all user data if local authentication is used, and the LDAP/AD Server will store only the authentication data in the event of LDAP enterprise authentication being used. The roles are always stored locally, and when LDAP is used the LDAP validated username is used to query these attributes. The username and password are for authenticating to the TOE.

All security management functions for the web GUI are managed by roles (permissions) being assigned to certain groups. Authorized actions for a particular user are dependent on which group they are assigned to. A user can only be assigned to one group. There are 4 initial groups: Administrator, Account Administrator, Advanced Users, and User. Groups have permissions assigned to them, which determines what actions members of a group can take. The Admin user (default user account) is a member of the Administrators Group. The Admin User cannot change their name, and the permissions of the Administrator Group cannot be changed. The Admin user can create other Administrator users, and change the permissions of other groups, however the Account Administrator group cannot create new users in the Admins Group. The web GUI can only be accessed remotely.

## 8.5 Protection of the TSF

### 8.5.1 FPT_APW_EXT.1

The TOE stores passwords in several different locations and secures them through different means depending on their use. The TOE stores passwords for the VCR user and web GUI users. The VCR user’s password is stored in the OS’s password file which has configurable hashing and in the evaluated configuration will use SHA-512. Web GUI user passwords are stored in an internal PostgreSQL Database and hashed using SHA-256. The VCR user is able to view the locations of these passwords but can only see their hash or encrypted values.

### 8.5.2 FPT_SKP_EXT.1

The TOE does not contain any interface that was specifically designed to view any of its pre-shared keys, symmetric keys, and private keys. The Diffie-Hellman Shared Secret, Diffie Hellman private exponent, and SSH session key are stored in volatile memory (RAM) and are not accessible by any user. Because this key data is stored in memory, core dumps are disabled to prevent this data from being disclosed if an error were to occur on the underlying operating system.

The SSH private key and SSL server key are stored on the local filesystem and RAM. When these keys are stored in RAM, they have the same protections as the other keys stored in volatile memory. After these keys are created by the VCR user via the Console or remote CLI, they will be assigned permissions to prevent unauthorized access which is enforced by the TOE’s access controls. The VCR user also has the ability to delete SSH private key and SSL server key using the rm –P <keyfile> command via the Console or remote CLI.

### 8.5.3 FPT_STM_EXT.1

The TOE has an underlying hardware clock that is used for time keeping. The Admin can use the web GUI to set the time zone, date options, and the time format used for audit records and TOE functionality (seconds, milliseconds, microseconds, or nanoseconds); however, the web GUI cannot be used to specifically change the time and/or date. To specifically change the time and/or date, the VCR user can configure all aspects of the clock using the local CLI or remote CLI via SSH. The TOE uses the clock for several purposes in the time format selected including for:

- Audit records
- Inactivity timeout for local CLI sessions
- Inactivity timeout for remote CLI sessions
- Inactivity timeout for remote web GUI sessions
- X.509 certificate validation
- Time period for failed authentication attempts

### 8.5.4 FPT_TST_EXT.1

Self-tests are conducted by the TOE during initial start-up (on power-on) to ensure the integrity of the FIPS cryptographic module, CPU functions, RAM memory, and the DRBG.

When the cryptographic module is loaded into memory, it conducts power-on self-tests to ensure its integrity and proper function. The FIPS cryptographic module verifies its own integrity using an HMAC-SHA-256 digest computed at build time. If the integrity test passes, then the cryptographic module conducts the power-on self-tests. These consist of algorithm-specific Pairwise Consistency and Known Answer Tests. All of the cryptographic algorithms used by the TOE are tested: AES, RSA, CTR_DRBG, HMAC (also covers SHA), ECC CDH (CVL), and DSA. The FIPS_mode_set() function performs all power-on self-tests listed above with no operator intervention required, returning a “1” if all power-on self-tests succeed, and a “0” otherwise. If any component of the power-on self-test fails an internal flag is set to prevent subsequent invocation of any cryptographic function calls. The module will only enter the FIPS Approved mode if the module is reloaded and the call to FIPS_mode_set() succeeds.

The CPU tests ensure the health of its general purpose instructions, floating point unit instructions, CPU extension instructions, prime number calculations, and proper functioning of the cooling system. To verify general purpose instructions (x86), the tests exercise and verify the following functions: integer mathematics, data transfer instructions, bitwise logical instructions, shift and rotate instructions, logical instructions, control transfer instructions, and string instructions. Floating point unit (FPU) instructions (x87) are tested by verifying the following floating point unit functions: floating point math, transcendentals, and load constants. CPU extension instructions for the x86 CPU instruction set are tested to ensure their proper function for a variety of applications such as MMX, SSE, SSE2, SSE3, SSE4.2, SSE4a, and AES. Prime number tests are conducted to exercise and verify correct operations of the CPU through the use of a prime number generation algorithm. Finally, the maximum heat test is conducted by running a BurnInTest on startup to generate the highest CPU temperature possible to ensure the cooling system is functional under extreme CPU load.

To verify the RAM memory is working, an 8-bit Cell Adjacency Test has the operating system write and validate each memory address by cycling through the following patterns: Sequence (0, 1, 2, … 255), Binary 1 (10101010), Binary 2 (01010101), Zeros (00000000), Ones (11111111). In addition, the "walking 0" and "walking 1" tests are performed where an isolated 0 or 1 gets shifted through the 8 positions of a cell.

To test the Disk subsystem, a FIO is used to generate load onto the physical disks. The test conducts sequential reads, sequential writes, random reads, and random writes. The tests are run over a period of time and results are collected. Those results are analyzed by the testing system to catch errors and faulty disks so the proper and optimal performance of the TOE can be maintained.

The DRBG has health tests performed on it, which are further described in the proprietary entropy document that was reviewed by NIAP.

### 8.5.5 FPT_TUD_EXT.1

The TOE's software version that is currently executing is the same as the last installed software version. TOE users can find the TOE's current software version via the web GUI or the CLI. For the web GUI, the user can check the current software version in the 'About' tab. In the CLI, the command "appliance_env" returns the current executed software versions. The "applhistory" command returns the software image version history.

The TOE software is updated by the VCR user via the CLI. When an updated software image becomes available, an administrator with a support account at supportnet.niksun.com will receive an email or the administrator can go to supportnet.niksun.com to view available software image patches. To update the TOE, the software image is downloaded to the SCP Server. The VCR user can either transfer the image onto a CD and then transfer it to the TOE, or use SCP to securely transfer the image directly to the TOE from the SCP Server.

When an update is performed, the TOE conducts a hash verification on the system image using SHA-256. If the two hashes match the install continues and the TOE will restart running the latest version. Otherwise, if the values do not match, the VCR user will receive an error message and the install process is halted.

## 8.6 TOE Access

### 8.6.1 FTA_SSL_EXT.1

The TOE is designed to terminate a local session via the keyboard and monitor after a specific time period of inactivity. The CLI timeout value is configured by the VCR user at the OS level in /etc/profile TMOUT value with a default of 900 (15 minutes), a minimum value of 5 minutes and a maximum value no greater than the 'System timeout'.

The 'System timeout' can be reset by the VCR user in the /usr/local/niksun/apps/etc/apps.conf file under session_timeout= 86400 (24 hours) with a maximum of 24 hours, a minimum of 5 minutes, and a default of 1 hour.

### 8.6.2 FTA_SSL.3

The TOE is designed to terminate a remote session using the web GUI or CLI after a given amount of time passes on the system clock. The CLI timeout can be configured by the VCR user in the same manner as defined in FTA_SSL_EXT.1.

Web GUI users can set a time limit for themselves by navigating through Preferences to web session timeout to in seconds, with a minimum of 300 seconds (5 minutes), a default of 1 hour, and a maximum no greater than the ‘System timeout’.

The ‘System timeout’ can be reset by the VCR in the same manner as defined in FTA_SSL_EXT.1.

### 8.6.3 FTA_SSL.4

The VCR user accessing the TOE remotely or locally through the CLI can terminate their session by entering the exit command. Users accessing the TOE remotely through the web GUI will terminate their sessions by clicking the logout button.

### 8.6.4 FTA_TAB.1

There are three possible ways to log in to the TOE: local CLI, remote CLI, and a remote web GUI. When logging in locally or remotely through the CLI, the standard Linux pre-authentication banner is displayed. It can be configured by the VCR user via the CLI. When connecting remotely via the web GUI, a pre-authentication banner is displayed that can be configured by an Administrator through the web GUI.

## 8.7 Trusted Path/Channels

### 8.7.1 FTP_ITC.1

The TOE connects with a Syslog Server, LDAP/AD Server, SCP Server, SMTP Server and a NetOmni appliance to send and receive data. All devices reside in the operational environment and communicate with the TOE via trusted channels. The channels are logically distinct from each other and do not interfere with the operation of the other channels of communication for other devices. When initiating the connection to the Syslog Server to transfer audit records the records are secured with TLS. SCP using an SSH client is used to transfer the software image updates to the TOE. TLS is used to secure the connection imitated by the TOE and an external LDAP/AD. When a user performs the “Forgot Username/Password” feature, the TOE will send an email protected by TLS to an SMTP Server. HTTPS/TLS is used by NetOmni, operating as the client, to connect to the TOE, operating as the server, to receive policy updates and transfer network metadata for NetDetector/NetVCR’s primary purpose.

When acting as a client the TOE initiates communication and when acting as a server the TOE receives connection requests using the protocols discussed in the relevant SFRs to protect the data traversing the channel from disclosure and/or modification. HTTPS conforms to FCS_HTTPS_EXT.1. The TLS conforms to FCS_TLSC_EXT.1 and FCS_TLSS_EXT.1.

### 8.7.2 FTP_TRP.1/Admin

When Administrator users connect to the TOE remotely, they are required by the TOE not only to authenticate but also to use trusted paths. When using the web GUI, HTTPS/TLS is used to secure the channel, and is conformant to SFRs FCS_HTTPS_EXT.1 and FCS_TLSS_EXT.1. When connecting by the remote CLI, users must use SSH, which is conformant to FCS_SSHS_EXT.1. These protocols are used to protect the data traversing the channel from disclosure and/or modification.