# NIKSUN NetOmni, and NetDetector/NetVCR/LogWave running Everest Software v6.0.1.0 Security Target

Document Version: 1.8

## Revision History:

| Version | Date | Changes |
|---|---|---|
| Version 1.1 | 21-May-2024 | Initial Draft |
| Version 1.2 | 28-Oct-2024 | 2nd Draft |
| Version 1.3 | 26-Dec-2024 | 3rd Draft |
| Version 1.4 | 14-Apr-2025 | 4th Draft |
| Version 1.5 | 20-Apr-2025 | 5th Draft |
| Version 1.6 | 20-Apr-2025 | 6th Draft |
| Version 1.7 | 27-Apr-2025 | 7th Draft |
| Version 1.8 | 1-May-2025 | Final Draft |

# Contents

1 Introduction ........ 5
1.1 Security Target and TOE Reference ........ 5
1.2 TOE Overview ........ 5
1.3 TOE Description ........ 6
1.3.1 Physical Boundaries ........ 6
1.3.2 Security Functions Provided by the TOE ........ 7
1.3.3 TOE Documentation ........ 9
1.4 TOE Environment ........ 10
1.5 Product Functionality not Included in the Scope of the Evaluation ........ 10
2 Conformance Claims ........ 11
2.1 CC Conformance Claims ........ 11
2.2 Protection Profile Conformance ........ 11
2.2.1 Technical Decisions ........ 11
3 Security Problem Definition ........ 13
3.1 Threats ........ 13
3.2 Assumptions ........ 14
3.3 Organizational Security Policies ........ 16
4 Security Objectives ........ 17
4.1 Security Objectives for the Operational Environment ........ 17
5 Security Requirements ........ 18
5.1 Conventions ........ 19
5.2 Security Functional Requirements ........ 19
5.2.1 Security Audit (FAU) ........ 19
5.2.2 Cryptographic Support (FCS) ........ 22
5.2.3 Identification and Authentication (FIA) ........ 26
5.2.4 Security Management (FMT) ........ 28
5.2.5 Protection of the TSF (FPT) ........ 29
5.2.6 TOE Access (FTA) ........ 30
5.2.7 Trusted Path/Channels (FTP) ........ 30
5.3 TOE SFR Dependencies Rationale for SFRs ........ 32
5.4 Security Assurance Requirements ........ 32
5.5 Assurance Measures ........ 33
6 TOE Summary Specifications ........ 34

6.1 CAVP Algorithm Certificate Details ........ 48
6.2 Cryptographic Key Destruction ........ 49
7 Acronym Table ........ 50

# 1 Introduction

The Security Target (ST) serves as the basis for the Common Criteria (CC) evaluation and identifies the Target of Evaluation (TOE), the scope of the evaluation, and the assumptions made throughout. This document will also describe the intended operational environment of the TOE, and the functional and assurance requirements that the TOE meets.

## 1.1 Security Target and TOE Reference

This section provides the information needed to identify and control the TOE and the ST.

**Table 1 – TOE/ST Identification**

| Category | Identifier |
|---|---|
| ST Title | NIKSUN NetOmni, and NetDetector/NetVCR/LogWave running Everest software v6.0.1.0 Security Target |
| ST Version | 1.8 |
| ST Date | May 1, 2025 |
| ST Author | NIKSUN, Inc. |
| TOE Identifier | NIKSUN NetOmni, and NetDetector/NetVCR/LogWave running Everest software |
| TOE Hardware | B1000, C3010 |
| TOE Version | 6.0.1.0 |
| TOE Developer | NIKSUN, Inc. |
| Key Words | Network Device, Security Appliance |

## 1.2 TOE Overview

The TOE includes the NIKSUN NetOmni, and NetDetector/NetVCR/LogWave appliances, running the software Everest version 6.0.1.0. NIKSUN NetOmni, and NetDetector/NetVCR/LogWave independently represents a TOE. Each of the appliances are running the exact same Everest software and the functionality is distinguished based on the licenses that are activated on the appliance.

NetOmni's primary functionality is to provide an overview of critical operations of the monitored network. The overview includes monitoring business service disruptions, performance issues, and security incidents. NetOmni accomplishes this by providing performance monitoring, traffic analysis, and reporting systems for a network[1]. NetOmni communicates to one or more NetDetector/NetVCR/LogWave appliances to collect data from distributed point solutions. The data is aggregated from many sources based on user-defined criteria so that it can be viewed as one flow. NetOmni generates reports based on the data collected that covers network-wide services, applications, and performance. Finally, NetOmni provides real-time network-wide analysis, forensics, and event alerting.

NetDetector's primary functionality is to provide security monitoring of network traffic using IDS methods and statistical anomaly detection to safeguard networks against cyber-attacks. The anomaly

[1] Note: NetOmni's performance monitoring, traffic analysis, forensics, event alerting and reporting features are not evaluated as part of the CC evaluation.

detection uses user-defined and threshold-based anomalies[2]. Users of NetDetector are notified of security breaches as soon as they occur. NetVCR is a solution for full packet capture with stream-to-disk recording, real-time indexing, and application analytics for network/application performance. LogWave is an advanced log and event analytics engine that provides real-time analysis of security alerts generated by applications or services.

NetOmni, and NetDetector/NetVCR/LogWave appliances running Everest software individually represent the TOE. They are identical in terms of security and management features and independently meet all the mandatory security requirements of the Protection Profile. The TOE allows Security Administrators to access the TOE through a local CLI, remote CLI via SSH, and a web GUI via TLS/HTTPS.

## 1.3 TOE Description

This section provides an overview of the TOE architecture, including physical boundaries, security functions, and relevant TOE documentation and references.

### 1.3.1 Physical Boundaries

The physical boundaries of the TOE consist of the physical appliance including all the hardware and the software. The TOE appliance model numbers and corresponding processor are shown in table below.

**Table 2a - TOE Hardware Models**

| Appliance | Model # | Processor | CPU Microarchitecture | OS |
|---|---|---|---|---|
| NetOmni | B1000 | AMD EPYC 7252 | AMD Zen-2 | NIKOS-Max 12 |
| NetDetector/NetVCR/LogWave | C3010 | Intel Xeon Gold 6238 | Intel Cascade Lake | NIKOS-Max 12 |

The physical boundaries of the TOEs are illustrated in the figure below with red dotted lines. Table 3 provides a list of environmental components that are part of the evaluated configuration.

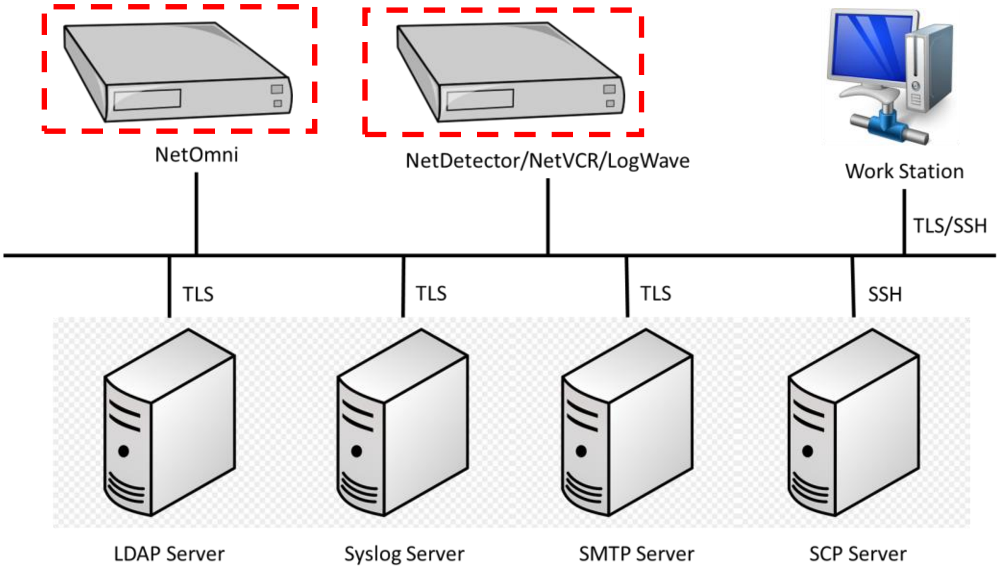

**Figure 1**

[2] Note: NetDetector/NetVCR/LogWave traffic monitoring, IDS, anomaly detection, application analytics, and event analytics features are not evaluated as part of the CC evaluation.

### 1.3.2 Security Functions Provided by the TOE

The TOE provides the security functions required by NDcPP v2.2e.

#### 1.3.2.1 Security Audit

The TOE provides extensive auditing capabilities. The TOE can audit events related to cryptographic functionality, identification and authentication, and administrative actions. The TOE generates an audit record for each auditable event. The TOE keeps local and remote audit records of security relevant events. The TOE internally maintains the date and time, which can be set manually. Each security relevant audit event includes the date, timestamp, event description, and subject identity. The TOE provides the administrator with a circular audit trail. The TOE can be configured to transmit its audit messages to an external syslog server over an encrypted channel using TLS.

#### 1.3.2.2 Cryptographic Support

The TOE relies on its NIKOS-Max FIPS Object Module and NIKOS-Max Java Object Module to implement cryptographic methods and trusted channels. The TOE uses TLS to secure the automatic transfer of syslog audit files and VAR logs to the Syslog Server. The TOE uses TLS to secure the connection to the LDAP/AD Server for remote authentication. When a user utilizes the "Forgot Username/Password" feature on the login screen, the TOE will send an email to the SMTPS Server over a protected TLS channel. TOE communicates with another NIKSUN appliance over TLS. 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 communications with the SCP Server when the TOE receives software image updates. TLS/HTTPS is used to secure the connection for remote management of the TOE via the web GUI as well as connections to other devices. The TOE will deny any connections for disallowed protocols and invalid X.509v3 certificates.

**Table 2b – Appliance cryptographic providers**

| Cryptographic Provider | Protocol | Usage |
|---|---|---|
| Bouncy Castle v1.0.2.5 | HTTPS (TLS 1.2) | NIKSUN appliance and SMTPS Servers |
| OpenSSL v3.0.15 | TLS 1.2<br>SSHv2 | Syslog, HTTPS Server, and LDAP<br>SSH Server, SCP Server |

#### 1.3.2.3 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 can be 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 authentication for the Syslog Server. 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).

#### 1.3.2.4 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 root 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 digital signature validation.

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 to match the terminology used by Common Criteria.

#### 1.3.2.5 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 and root 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. The TOE performs integrity checks during initial start-up (power-on) to ensure the firmware integrity. After successful integrity checks, the TOE then further performs all cryptographic algorithm self-tests for its OpenSSL and Bouncy Castle cryptographic providers. The TOE also performs self-tests on the CPU, RAM, and disk components. The TOE's DRBGs also perform their own health tests.

The version of the TOE is verified via the CLI or web GUI. The TOE is updated by the root 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..

#### 1.3.2.6 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 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. Administrative users have the capability to terminate their own sessions.

#### 1.3.2.7 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. The TLS connection to the Syslog server is over TLS channel. TLS is used to connect to an SMTP email server for secure credentials reset. TLS is also used for the TOE's connection with the LDAP/AD Server for its remote authentication store. TLS is used for the transfer of data between the

NIKSUN appliances. SSH is used for the connection to the SCP Server when the TOE receives software image updates.

TLS/HTTPS and SSH are used for remote administration of the TOE via the web GUI and remote CLI respectively.

### 1.3.3 TOE Documentation

The following documents are essential to understanding and controlling the TOE in the evaluated configuration:

- NIKSUN NetOmni, NetDetector/NetVCR/LogWave running Everest software v6.0.1.0 Common Criteria Guide [AGD]

## 1.4 TOE Environment

The following environmental components are required to operate the TOE in the evaluated configuration:

**Table 3 –Environmental Components**

| Components | Description | Version Requirements |
|---|---|---|
| NIKSUN appliance | Another instance of the TOE | Everest software 6.0.1.0 |
| LDAP Server | Remote authentication | OpenLDAP 2.5.x or later |
| SCP Server | Firmware updates via an SCP server | Any SSH-2 compliant |
| SMTP Server | Email server | Any modern SMTP; POP3 or IMAP compliant |
| Syslog Server | External storage for audit logs | Any syslog protocol (per RFC 5424) compliant |
| Workstation | Local or remote management | - |

## 1.5 Product Functionality not Included in the Scope of the Evaluation

The hardware and software of the TOE environment, identified above in Section 1.4, are not included in the CC evaluation scope.

Only the security functions specified in Section 5.2 (Security Functional Requirements) and Section 6 (TOE Summary Specifications) are included in the CC evaluation scope. Other product features and functions not included in the scope of this ST are deemed unevaluated and non-interfering. These features and functions include the following:

- Performance monitoring, service disruptions and forensics
- Traffic and network monitoring and analysis
- Event analytics, alerting, and reporting
- Security incidents, IDS, and anomaly detection

Furthermore, when operating the TOE in a manner compliant with this ST, the following features may not be used:

- NTP-based updates to TOE time
- IPv6

# 2 Conformance Claims

This section identifies the TOE conformance claims, conformance rationale, and relevant Technical Decisions (TDs).

## 2.1 CC Conformance Claims

The TOE is conformant to the following:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017 (Extended)
- Common Criteria for Information Technology Security Evaluations Part 3, Version 3.1, Revision5, April 2017 (Conformant)

## 2.2 Protection Profile Conformance

This ST claims exact conformance to the Collaborated Protection Profiles for Network Devices, Version 2.2e, March 27, 2020.

### 2.2.1 Technical Decisions

All NIAP Technical Decisions (TDs) issued to date and applicable to NDcPP v2.2e have been considered. Table 4 identifies all applicable TDs.

**Table 4 – Relevant Technical Decisions**

| Technical Decision | Applicable (Y/N) | Exclusion Rationale (if applicable) |
|---|---|---|
| TD0800: Updated NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance | No | FCS_IPSEC_EXT.1 is not claimed by the TOE |
| TD0792: NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR | Yes | |
| TD0790: NIT Technical Decision: Clarification Required for testing IPv6 | No | IPv6 is not claimed by the TOE |
| TD0738: NIT Technical Decision for Link to Allowed-With List | Yes | |
| TD0670: NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing | Yes | |
| TD0639: NIT Technical Decision for Clarification for NTP MAC Keys | No | FCS_NTP_EXT.1 is not claimed by the TOE. |
| TD0638: NIT Technical Decision for Key Pair Generation for Authentication | No | The TOE is not a Distributed TOE |
| TD0636: NIT Technical Decision for Clarification of Public Key User Authentication for SSH | Yes | |
| TD0635: NIT Technical Decision for TLS Server and Key Agreement Parameters | Yes | |

| Technical Decision | Applicable (Y/N) | Exclusion Rationale (if applicable) |
|---|---|---|
| TD0632: NIT Technical Decision for Consistency with Time Data for vNDs | No | The TOE is not a virtual TOE. |
| TD0631: NIT Technical Decision for Clarification of public key authentication for SSH Server | Yes | |
| TD0592: NIT Technical Decision for Local Storage of Audit Records | Yes | |
| TD0591: NIT Technical Decision for Virtual TOEs and hypervisors | No | The TOE is not a virtual TOE. |
| TD0581: NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3 | Yes | |
| TD0580: NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e | Yes | |
| TD0572: NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers | Yes | |
| TD0571: NiT Technical Decision for Guidance on how to handle FIA_AFL.1 | No | The TOE can distinguish between local and remote connections. |
| TD0570: NiT Technical Decision for Clarification about FIA_AFL.1 | Yes | |
| TD0569: NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7 | Yes | |
| TD0564: NiT Technical Decision for Vulnerability Analysis Search Criteria | Yes | |
| TD0563: NiT Technical Decision for Clarification of audit date information | Yes | |
| TD0556: NIT Technical Decision for RFC 5077 question | Yes | |
| TD0555: NIT Technical Decision for RFC Reference incorrect in TLSS Test | Yes | |
| TD0547: NIT Technical Decision for Clarification on developer disclosure of AVA_VAN | Yes | |
| TD0546: NIT Technical Decision for DTLS - clarification of Application Note 63 | No | DTLS is not claimed by the TOE. |
| TD0537: NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3 | No | |
| TD0536: NIT Technical Decision for Update Verification Inconsistency | Yes | |
| TD0528: NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4 | No | FCS_NTP_EXT.1 is not claimed by the TOE. |
| TD0527: Updated to Certificate Revocation Testing (FIA_X509_EXT.1) | Yes | |

# 3 Security Problem Definition

The security problem definition has been taken directly from the claimed PP and is reproduced here for the convenience of the reader. The security pro blem is described in terms of the threats that the TOE is expected to address, assumptions about the operational environment, and any Organizational Security Policies (OSPs) that the TOE is expected to enforce.

## 3.1 Threats

The threats included in Table 5 are drawn directly from the PP specified in Section 2.2.

**Table 5 – Threats**

| ID | Threat |
|---|---|
| T.UNAUTHORIZED_ADMINISTRATOR_ACCESS | Threat agents may attempt to gain Administrator access to the Network Device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between Network Devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides. |
| T.WEAK_CRYPTOGRAPHY | Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort. |
| T.UNTRUSTED_COMMUNICATION_CHANNELS | Threat agents may attempt to target Network Devices that do not use standardized secure tunneling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the Network Device itself. |
| T.WEAK_AUTHENTICATION_ENDPOINTS | Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints, e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker |

| ID | Threat |
|---|---|
| | 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. |

## 3.2 Assumptions

The assumptions included in Table 6 are drawn directly from PP and any relevant.

**Table 6 – Assumptions**

| ID | Assumption |
|---|---|
| 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 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 does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does 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. For vNDs, this assumption applies to the physical platform on which the VM runs. |
| 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 and PP-Modules 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 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.<br>For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification). |

| ID | Assumption |
|---|---|
| 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. |

## 3.3 Organizational Security Policies

The OSPs included in Table 7 are drawn directly from the PP.

**Table 7 – OSPs**

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

# 4 Security Objectives

The security objectives have been taken directly from the claimed PP and are reproduced here for the convenience of the reader.

## 4.1 Security Objectives for the Operational Environment

Security objectives for the operational environment assist the TOE in correctly providing its security functionality. These objectives, which are found in the table below, track with the assumptions about the TOE operational environment.

Table 8 – Security Objectives for the Operational Environment

| ID | Objectives for 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.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. Note: For vNDs the TOE includes only the contents of the its own VM, and does not include other VMs or the VS. |
| 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.TRUSTED_ADMN | Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality.<br>For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted. |
| 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.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.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. For vNDs, this applies when the physical platform on which the VM runs is removed from its operational environment. |

# 5 Security Requirements

This section identifies the Security Functional Requirements (SFRs) for the TOE. The SFRs included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, September 2017, and all international interpretations.

**Table 9 – SFRs**

| Requirement | Description |
|---|---|
| FAU_GEN.1 | Audit Data Generation |
| FAU_GEN.2 | User Identity Association |
| FAU_STG_EXT.1 | Protected Audit Event Storage |
| 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_SSHC_EXT.1 | SSH Client Protocol |
| FCS_SSHS_EXT.1 | SSH Server Protocol |
| FCS_TLSC_EXT.1 | TLS Client Protocol without Mutual Authentication |
| FCS_TLSS_EXT.1 | TLS Server Protocol |
| FCS_RBG_EXT.1 | Random Bit Generation |
| FIA_AFL.1 | Authentication Failure Management |
| FIA_PMG_EXT.1 | Password Management |
| FIA_UIA_EXT.1 | User Identification and Authentication |
| FIA_UAU_EXT.2 | Password-based Authentication Mechanism |
| FIA_UAU.7 | Protected Authentication Feedback |
| 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 |
| FMT_MOF.1/ManualUpdate | Management of Security Functions Behaviour |
| FMT_MTD.1/CoreData | Management of TSF Data |
| FMT_MTD.1/CryptoKeys | Management of TSF Data |
| FMT_SMF.1 | Specification of Management Functions |
| FMT_SMR.2 | Restrictions on security roles |
| FPT_SKP_EXT.1 | Protection of TSF Data (for reading of all pre-shared, symmetric and private keys) |
| FPT_APW_EXT.1 | Protection of Administrator Passwords |
| FPT_TST_EXT.1 | TSF Testing |
| FPT_STM_EXT.1 | Reliable Time Stamps |
| FPT_TUD_EXT.1 | Trusted Update |
| FTA_SSL.3 | TSF-initiated Termination |
| FTA_SSL.4 | User-initiated Termination |
| FTA_SSL_EXT.1 | TSF-initiated Session Locking |
| FTA_TAB.1 | Default TOE Access Banner |
| FTP_ITC.1 | Inter-TSF Trusted Channel |
| FTP_TRP.1/Admin | Trusted Path |

## 5.1 Conventions

The CC allows the following types of operations to be performed on the functional requirements: assignments, selections, refinements, and iterations. The following font conventions are used within this document to identify operations defined by CC:

- Assignment: Indicated with *italicized* text;
- Refinement: Indicated with **bold** text;
- Selection: Indicated with underlined text;
- Iteration: Indicated by appending the iteration identifier after a slash, e.g., /SigGen.
- Where operations were completed in the PP, the formatting used in the PP has been retained.
- Extended SFRs are identified by the addition of "EXT" after the requirement name.

## 5.2 Security Functional Requirements

This section includes the security functional requirements for this ST.

### 5.2.1 Security Audit (FAU)

#### 5.2.1.1 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) 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 10a***.

**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 10a.***

**Table 10a – Security Functional Requirements and Auditable Events**

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FAU_GEN.1 | None | None |
| FAU_GEN.2 | None | None |
| FAU_STG_EXT.1 | None | None |
| FCS_CMK.1 | None | None |
| FCS_CKM.2 | None | None |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| 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_RBG_EXT.1 | None | None |
| FCS_HTTPS_EXT.1 | Failure to establish a HTTPS Session | Reason for failure |
| 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_UIA_EXT.1 | All use of identification and authentication mechanism | Origin of the attempt (e.g., IP address) |
| 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_X509_EXT.1/Rev | • Unsuccessful attempt to validate a certificate<br>• Any addition, replacement or removal of trust anchors in the TOE's trust store | • Reason for failure of certificate validation<br>• Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store |
| 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/CoreData | None | None |
| FMT_MTD.1/CryptoKeys | None | None |
| FMT_SMF.1 | All management activities of TSF data | None |
| FMT_SMR.2 | None | None |
| FPT_SKP_EXT.1 | None | None |
| FPT_APW_EXT.1 | None | None |
| FPT_TST_EXT.1 | None. | None. |
| FPT_STM_EXT.1 | Discontinuous changes to time - either Administrator actuated or changed via an automated process<br>(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: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address). |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FPT_TUD_EXT.1 | Initiation of update; result of the update attempt (success or failure) | None |
| FTA_SSL.3 | The termination of a remote session by the session locking mechanism | None |
| FTA_SSL.4 | The termination of an interactive session | None |
| FTA_SSL_EXT.1 | The termination of a local session by the session locking mechanism | 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 |

**Application Note:** The following table extends Table 10a above to include events indicated in FAU_GEN.1.1 a) through c).

**Table 10b – Security Functional Requirements and Auditable Events (FAU_GEN.1.1**

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FAU_GEN.1.1 a) | • Start-up of audit functions<br>• Shut-down of audit functions | None |
| FAU_GEN.1.1 b) | None | None |
| FAU_GEN.1.1 c) Bullet 1 | • Administrative login<br>• Administrative logout | User account name |
| FAU_GEN.1.1 c) Bullet 2 | TSF data related configuration parameter change | Configuration parameter name |
| FAU_GEN.1.1 c) Bullet 3 | • Generate/import key<br>• Change key<br>• Delete key | Key name or identifier |
| FAU_GEN.1.1 c) Bullet 4 | • Reset password | User account name |

### 5.2.1.2 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.

#### 5.2.1.3 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. In addition [

- *The TOE shall consist of a single standalone component that stores audit data locally,*

].

**FAU_STG_EXT.1.3**
The TSF shall [*overwrite previous audit records according to the following rule: [overwrite oldest log file once configured log file limits are reached]*] when the local storage space for audit data is full.

### 5.2.2 Cryptographic Support (FCS)

#### 5.2.2.1 FCS_CKM.1 Cryptographic Key Generation

**FCS_CKM.1.1**
The TSF shall generate **asymmetric** cryptographic key 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 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 3526].*

] ~~and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].~~

#### 5.2.2.2 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: [

- *Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";*
- *FFC Schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [groups listed in RFC 3526].*

] ~~that meets the following: [assignment: list of standards].~~

#### 5.2.2.3 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*

### 5.2.2.4 FCS_COP.1/DataEncryption Cryptographic Operations (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* [*CTR, GCM*] *mode* and cryptographic key sizes [*128 bits, 256 bits*] that meet the following: *AES as specified in ISO 18033-3,* [*CTR as specified in ISO 10116, GCM as specified in ISO 19772*].

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

**FCS_COP.1.1/SigGen**
The TSF shall perform *cryptographic signature services (generation and verification)* in accordance with a specified cryptographic algorithm [

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

]
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,*

].

### 5.2.2.6 FCS_COP.1/Hash Cryptographic Operations (Hash Algorithm)

**FCS_COP.1.1/Hash**
The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm [*SHA-256, SHA-384, SHA-512*] and **message digest sizes [*256, 384, 512*] bits** that meet the following: *ISO/IEC 10118-3:2004*.

### 5.2.2.7 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-256, HMAC-SHA-384, HMAC-SHA-512*] and cryptographic key sizes *[256, 384, 512 bits]* **and message digest sizes [*256, 384, 512*] bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

### 5.2.2.8 FCS_HTTPS_EXT.1 HTTPS Protocol

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

**FCS_HTTPS_EXT.1.2**
The TSF shall implement the HTTPS protocol using TLS.

**FCS_HTTPS_EXT.1.3**
If a peer certificate is presented, the TSF shall [*not establish the connection*] if the peer certificate is deemed invalid.

### 5.2.2.9 FCS_SSHC_EXT.1 SSH Client Protocol

**FCS_SSHC_EXT.1.1**
The TSF shall implement the SSH protocol in accordance with: RFCs *4251, 4252, 4253, 4254*, *[4256, 4344, 5656, 6668, 8268, 8308 section 3.1, 8332].*

**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 *[262126]* 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-ctr, aes256-ctr, aes128-gcm@openssh.com, aes256-gcm@openssh.com*].

**FCS_SSHC_EXT.1.5**
The TSF shall ensure that the SSH public-key based authentication implementation uses [*rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256*] 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, implicit*] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

**FCS_SSHC_EXT.1.7**
The TSF shall ensure that [*ecdh-sha2-nistp256] and [diffie-hellman-group14-sha256*] 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 each encryption key is used to protect no more than one gigabyte of data. After any 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 and [*no other methods*] as described in RFC 4251 section 4.1.

### 5.2.2.10 FCS_SSHS_EXT.1 SSH Server Protocol

**FCS_SSHS_EXT.1.1**
The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [*4256, 4344, 5656, 6668, 8268, 8308* section 3.1, *8332*].

**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 *[262126]* 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-ctr, aes256-ctr, aes128-gcm@openssh.com, aes256-gcm@openssh.com*].

**FCS_SSHS_EXT.1.5**
The TSF shall ensure that the SSH public-key based authentication implementation uses [*rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256*] 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, implicit*] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

**FCS_SSHS_EXT.1.7**
The TSF shall ensure that [*ecdh-sha2-nistp256] and [diffie-hellman-group14-sha256*] 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 each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

### 5.2.2.11 FCS_TLSC_EXT.1 TLS Client Protocol without Mutual Authentication

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

[

- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*

*] and no other ciphersuites.*

**FCS_TLSC_EXT.1.2**
The TSF shall verify that the presented identifier matches [*the reference identifier per RFC 6125 section 6, IPv4 address in CN or SAN, IPv4 address in SAN, and no other attribute types*].

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

- *Not implement any administrator override mechanism*].

**FCS_TLSC_EXT.1.4**
The TSF shall [*present the Supported Elliptic Curves/Supported Groups Extension with the following curves/groups: [secp256r1] and no other curves/groups*] in the Client Hello

#### 5.2.2.12 FCS_TLSS_EXT.1 TLS Sever Protocol Without Mutual Authentication

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

[

- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*

*] and no other ciphersuites.*

**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 perform key establishment for TLS using [*Diffie-Hellman parameters with size [2048 bits], ECDHE curves [secp256r1] and no other curves]*].

**FCS_TLSS_EXT.1.4**
The TSF shall support [*no session resumption or session tickets*].

#### 5.2.2.13 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 [*Hash_DRBG (any), CTR_DRBG (AES)*].

**FCS_RBG_EXT.1.2**
The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [[*1*] 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.

### 5.2.3 Identification and Authentication (FIA)

#### 5.2.3.1 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 using a password*.

**FIA_AFL.1.2**
When the defined number of unsuccessful authentication attempts has been met, the TSF shall [*prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until [invoking an account unlocking command] is taken by an Administrator; prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until an Administrator defined time period has elapsed*]. [3]

[3] Account unlocking by an Administrator is supported by both CLI and web GUI interfaces; whereas Administrator defined time-based account unlocking is only supported by web GUI interface.

### 5.2.3.2 FIA_PMG_EXT.1 Password Management

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

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

*b)* Minimum password length shall be configurable to between [*8*] and [*100*] characters.

### 5.2.3.3 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]*].

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

### 5.2.3.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

**FIA_UAU_EXT.2.1**
The TSF shall provide a local [*password-based*] authentication mechanism to perform local administrative user authentication.

### 5.2.3.5 FIA_UAU.7.1 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**.

### 5.2.3.6 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 certification path validation **supporting a minimum path length of three certificates** .
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [*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 extendedKeyUsagefield.*

- *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.

### 5.2.3.7 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 [*HTTPS, 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*].

### 5.2.3.8 FIA_X509_EXT.3 X.509 Certificate Requests

**FIA_X509_EXT.3.1**
The TSF shall generate a Certificate Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [*Common Name, Organization, Organizational Unit, Country*].

**FIA_X509_EXT.3.2**
The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

## 5.2.4 Security Management (FMT)

### 5.2.4.1 FMT_MOF.1/ManualUpdate Management of Security Functions Behavior

**FMT_MOF.1.1/ManualUpdate**
The TSF shall restrict the ability to enable the functions *to perform manual updates to Security Administrators*.

### 5.2.4.2 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.*

### 5.2.4.3 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*.

### 5.2.4.4 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 manage the cryptographic keys;*
    - *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;*
    - *Ability to import X.509v3 certificates to the TOE's trust store;*

].

#### 5.2.4.5 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.

### 5.2.5 Protection of the TSF (FPT)

#### 5.2.5.1 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.

#### 5.2.5.2 FTP_APW_EXT.1 Protection of Administrator Passwords

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

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

#### 5.2.5.3 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: [*Integrity, Pairwise Consistency, and Known Answer Tests;*].

#### 5.2.5.4 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*].

#### 5.2.5.5 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.

### 5.2.6 TOE Access (FTA)

#### 5.2.6.1 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.*

#### 5.2.6.2 FTA_SSL.4 User-initiated Termination

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

#### 5.2.6.3 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

#### 5.2.6.4 FTA_TAB.1 Default TOE Access Banners

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

### 5.2.7 Trusted Path/Channels (FTP)

#### 5.2.7.1 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, [NIKSUN appliances, SSH (SCP) sever, SMTP server]*]** 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]*.

#### 5.2.7.2 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*.

## 5.3 TOE SFR Dependencies Rationale for SFRs

The PP contain(s) all the requirements claimed in this ST. As such, the dependencies are not applicable since the PP has been approved.

## 5.4 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from the PP which is/are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in the Table 11.

**Table 11 – Security Assurance Requirements**

| Assurance Class | Assurance Components | Component Description |
|---|---|---|
| Security Target | ASE_CCL.1 | Conformance claims |
| | ASE_ECD.1 | Extended components definition |
| | ASE_INT.1 | ST introduction |
| | ASE_OBJ.1 | Security objectives for the operational environment |
| | ASE_REQ.1 | Stated security requirements |
| | ASE_SPD.1 | Security problem definition |
| | ASE_TSS.1 | TOE Summary Specification |
| Development | ADV_FSP.1 | Basic functional specification |
| Guidance Documents | AGD_OPE.1 | Operational user guidance |
| | AGD_PRE.1 | Preparative user guidance |
| Life Cycle Support | ALC_CMC.1 | Labelling of the TOE |
| | ALC_CMS.1 | TOE CM coverage |
| Tests | ATE_IND.1 | Independent testing – conformance |
| Vulnerability Assessment | AVA_VAN.1 | Vulnerability survey |

## 5.5 Assurance Measures

The TOE satisfied the identified assurance requirements. This section identifies the Assurance Measures applied by NIKSUN to satisfy the assurance requirements. The following table lists the details.

**Table 12 TOE Security Assurance Measures**

| SAR Component | How the SAR will be met |
|---|---|
| ADV_FSP.1 | The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes). |
| AGD_OPE.1 | The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance. |
| AGD_PRE.1 | The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration. |
| ALC_CMC.1 | The Configuration Management (CM) documents describe how the consumer identifies the evaluated TOE. The CM documents identify the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked and how potential changes are incorporated. |
| ALC_CMS.1 | |
| ATE_IND.1 | Vendor will provide the TOE for testing |
| AVA_VAN.1 | Vendor will provide the TOE for testing<br>Vendor will provide a document identifying the list of software and hardware components. |

# 6 TOE Summary Specifications

This chapter identifies and describes how the Security Functional Requirements identified above are met by the TOE.

**Table 13 – TOE Summary Specification SFR Description**

| Requirement | TSS Description |
|---|---|
| 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.<br>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. 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 an Admin deletes a log file, an audit record of that action is also recorded. All actions performed on the TOE are logged, including the auditable events defined in Table 10a and Table 10b.<br>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). Each audit record provides a timestamp, username and a description of the action performed including a success/failure indication. The web GUI also provides the IP address from where the administrator is managing the TOE.<br>In order to identify the key being operated on, the following details are recorded for all administrative actions relating to cryptographic keys (generating, importing, changing and deleting keys):<br>• HTTPS/TLS – certificate id will be recorded when generating or deleting a key pair<br>• TLS/SSH session keys– key reference provided by process id<br>• SSH key configured for SSH public key authentication –the hash of the public key that is to be used for authentication is recorded in syslog |
| 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:<br>Username=admin,<br>System name= Process crond (Cron Daemon),<br>IP address= 10.115.0.108. |
| 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., log.1, log.2, etc.) and a new log file is opened as the current log. Once the local log file reaches configured limit, the oldest file is deleted to provide space for the new files. In addition, many applications run from the CLI keep their own |

| Requirement | TSS Description |
|---|---|
| | 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.<br>The TOE will automatically manage audit files in two cases:<br>a) if the individual log file type has exhausted its allotted space, or<br>b) when the entire storage space for the TOE approaches full consumption.<br>If the storage space allocated for an individual file type is filled the TOE will overwrite the oldest log file of that type. If the entire 32 GBs allocated to /var/log is is 90% exhausted the TOE will initiate overwriting (overwriting) of oldest log files.<br>The vcr and root users are the only users that have access to the CLI and as a Security Administrator are 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 cannot modify or delete the audit records for malicious purposes. |
| FCS_CKM.1 | The TOE implements a FIPS PUB 186-4 conformant key generation mechanism for RSA key generation schemes for establishing TLS and SSH connections. 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.<br>The TOE also implements a FIPS PUB 186-4 conformant key generation mechanism for ECDSA key generation schemes for establishing TLS and SSH connections. Specifically, the TOE complies with the FIPS 186-4 (Digital Signature Standard (DSS) Appendix B.4). This is used to generate the ECDSA key pairs using the P-256 curve which has an equivalent key strength of 256 bits.<br>In addition, the TOE implements FFC schemes using safe primes that meet NIST SP 800-56A Revision 3 conformant key establishment mechanism for Diffie-Hellman key establishment schemes for SSH. This is used to generate the keys of size 2048 bits for diffie-hellman-group14-sha256.<br>In addition, the TOE implements EC schemes that meet NIST SP 800-56A Revision 3 conformant key establishment mechanism for ECDH key establishment schemes for SSH and TLS. This is used to generate the keys of size 256 bits for the P-256 curve supported by the TOE.<br>The TOE's key generation functions have the following ACVP certificates:<br>RSA: # A6781, A6782<br>DSA: # A6781, A6782 |
| FCS_CKM.2 | The TOE implements a FIPS PUB 186-4 conformant key establishment mechanism for ECDH key establishment schemes. Specifically, the TOE complies with *Elliptic curve-based key establishment* schemes that meet NIST Special Publication 800-56A Revision 3 requirements using the P-256 curve.<br>The TOE also implements a NIST SP 800-56A conformant key establishment mechanism for Diffie-Hellman key establishment schemes. Specifically, the TOE complies with *FFC Schemes using 'safe-prime' groups* that meet NIST Special |

<table>
<tr><th>Requirement</th><th>TSS Description</th></tr>
<tr><td></td><td>Publication 800-56A Revision 3 requirements. The TSF uses Diffie-Hellman-group14-SHA256 in accordance with RFC 3526, Section 3.
<table>
<tr><th>Scheme</th><th>SFR</th><th>Service</th></tr>
<tr><td>ECDH</td><td>FCS_SSHC_EXT.1</td><td>SCP/SSH to external file server (trusted channel)</td></tr>
<tr><td>ECDH</td><td>FCS_SSHS_EXT.1</td><td>SSH remote management (trusted path)</td></tr>
<tr><td>ECDH</td><td>FCS_TLSC_EXT.1</td><td>Audit server connection (trusted channel)<br>Peer appliance connection (NetOmni, trusted channel)<br>Authentication/LDAP server connection (trusted channel)<br>Email/SMTPS server connection (trusted channel)</td></tr>
<tr><td>ECDH</td><td>FCS_TLSS_EXT.1</td><td>Web/GUI remote management (trusted path)<br>Peer appliance connection (NetVCR/NetDetector/Logwave, trusted channel)</td></tr>
<tr><td>Diffie-Hellman (Group 14)</td><td>FCS_SSHC_EXT.1</td><td>SCP/SSH to external file server (trusted channel)</td></tr>
<tr><td>Diffie-Hellman (Group 14)</td><td>FCS_SSHS_EXT.1</td><td>SSH remote management (trusted path)</td></tr>
</table>
</td></tr>
<tr><td>FCS_CKM.4</td><td>The (EC) Diffie-Hellman Shared Secret, (EC) Diffie Hellman private and public parameters, TLS session keys, and SSH session keys are 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 and when the TOE is shut down as well as when power is lost. The SSH private and public keys and SSL server are stored on the local filesystem and RAM. When stored in RAM, the keys will be zeroized as described above. When new keys are generated, the TOE overwrites the location where the keys are stored on the local filesystem 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. None of the keys are stored in encrypted form. All keys are stored in plaintext in RAM or local filesystem.</td></tr>
<tr><td>FCS_COP.1/DataEncryption</td><td>The TOE performs encryption and decryption using the AES algorithm in CTR, and GCM mode with key sizes of 128 and 256 bits. This algorithm has ACVP AES certificates # A6781, A6782. The AES algorithm meets <i>CTR as specified in ISO 10116, GCM as specified in ISO 19772</i>.</td></tr>
<tr><td>FCS_COP.1/SigGen</td><td>The TOE performs cryptographic digital signature verification and generation in accordance with FIPS PUB 186-4: RSA Digital Signature Algorithm (rDSA) and Elliptic Curve Digital Signature Algorithm (ECDSA) with an RSA key size (modulus) of 2048 bits or greater and an ECDSA key size of 256 bits.<br>The algorithm has ACVP RSA certificate # A6781, A6782and ECDSA certificate # A6781, A6782. Note that FIPS 186-4 supersedes FIPS 186-3.</td></tr>
</table>

| Requirement | TSS Description |
|---|---|
| FCS_COP.1/Hash | The TOE provides cryptographic hashing services using SHA-256, SHA-384 and SHA-512 with message digest sizes of 256, 384, and 512 bits respectively, as specified in FIPS PUB 180-4. The TSF also uses SHA-256, SHA-384, and SHA-512 for HMAC message authentication, health tests, TLS certificate authentication and SSH. SHA-256 and SHA-512 are used in RSA and ECDSA by SSH and SHA-256 is used for RSA and ECDSA by TLS for Signature Generation and Signature Verification. The SHA algorithm meets ISO/IEC 10118-3:2004 and has ACVP SHS certificates A6781, A6782. |
| FCS_COP.1/KeyedHash | The TOE provides keyed-hashing message authentication services using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with key sizes *256, 384, 512 bits* and digest sizes of 256, 384, 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 certificates A6781, A6782. |
| 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 NikSun appliances. The NetOmni will initiate the connection and it will only be established if the peer certificate provided by the NetDetector/NetVCR/LogWave to the NetOmni is valid. A 2048 bit RSA is used by the HTTPS/TLS protocol.<br>HTTPS uses TLSv1.2 (as specified by FCS_TLSC_EXT.1 for the connecting to NIKSUN appliances and FCS_TLSS_EXT.1 for the web GUI) to securely establish the AES encrypted session which uses the TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, or TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 cipher suite.<br>Port 443 is used for initial HTTPS connections in accordance with RFC 2818. HTTPS services are provided by Apache HTTPD and Apache Tomcat modules in the TOE. |
| FCS_SSHC_EXT.1 | The TOE acts as an SSH Client when SCP is used to transfer the software image updates to the TOE.<br>SSH functionality is compliant with RFCs 4251, 4252, 4253, 4254, 4256, 4344, 5656, 6668 and 8332. 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 rsa-sha2-256, rsa-sha2-512 or an ECDSA key of 256 bits in length as described in RFC 5656 using ecdsa-sha2-nistp256 as its public key authentication algorithm.<br>Encryption is provided by aes128-ctr and aes256-ctr, with data integrity MAC algorithms hmac-sha2-256 and hmac-sha2-512, and diffie-hellman-group14-sha256 and ecdh-sha2-nistp256 as its key exchange algorithm.<br>The SSH connection will drop any connection when a packet greater than 262126 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. |
| FCS_SSHS_EXT.1 | The TOE acts as an SSH server for remote CLI management.<br>SSH functionality is compliant with RFCs 4251, 4252, 4253, 4254, 4256, 4344, 5656, |

| Requirement | TSS Description |
|---|---|
| | 6668 and 8332. 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 rsa-sha2-256, rsa-sha2-512 or an ECDSA key of 256 bits in length as described in RFC 5656 using ecdsa-sha2-nistp256 as its public key authentication algorithm.<br>Encryption is provided by aes128-ctr and aes256-ctr, with data integrity MAC algorithms hmac-sha2-256 and hmac-sha2-512, and diffie-hellman-group14-sha256 as its key exchange algorithm.<br>The SSH connection will drop any connection when a packet greater than 262126 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. |
| FCS_TLSC_EXT.1 | The TOE uses the TLSv1.2 protocol without mutual authentication and TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, or TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ciphersuites to secure the channel for the following purposes with these servers in the operational environment:<br>• sending of audit data to the Syslog Server,<br>• performing authentication requests with the LDAP/AD Server,<br>• sending “Forgot Username/Password” emails to an SMTP Server, and<br>• NIKSUN appliance in the environment (NetOmni appliance sending policy updates to and receiving network event data (metadata) from a NetDetector/NetVCR/LogWave appliance).<br>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 reference identifiers as per RFC 6125 Section 6, IPv4 address in CN or SAN. The TOE supports wildcards.<br>The TOE will present the Supported Elliptic Curves/Supported Groups Extension with P-256 in the Client Hello<br>The TOE uses X.509v3 certificates to support for communication with the syslog server. As part of the TLS session establishment, the TOE will provide its client certificate and validate the 2048-bit RSA certificate received from the operational environment server and will only establish the connection if the certificate is valid.<br>The TOE will verify the identity of the Syslog Server, LDAP/AD Server, SMTP Server, and NIKSUN appliance 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.<br>The TOE also supports IPv4 addresses in CN or SAN. For IP addresses in the CN as reference identifiers, the text representation of the IP address in the CN is converted to a binary representation of the IP address in network byte order separating the pattern at the ‘dot’. The TOE does not enforce canonical format. The TOE does not support certificate pinning. |

| Requirement | TSS Description |
|---|---|
| | The communication between the TOE and another NIKSUN appliance is a TLS channel. The TOE can act as a TLS Client while communicating with another NIKSUN appliance.<br>NetOmni appliance sends policy updates to and receives network event data (metadata) from a NetDetector/NetVCR/LogWave appliance. Therefore, when the TOE is configured as NetOmni, it acts as a TLS Client while sending policy updates to NIKSUN appliance and when the TOE is configured as NetDetector/NetVCR/LogWave, it acts as a TLS Client while sending network event data (metadata) to NIKSUN appliance. |
| FCS_TLSS_EXT.1 | Remote user administration via the web GUI is protected using HTTPS/TLS. The TOE uses the TLSv1.2 protocol and TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, or TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ciphersuites to secure the web GUI path to the TOE.<br>The communication between the TOE and another NIKSUN appliance is a TLS channel. The TOE can act as a TLS Server while communicating with another NIKSUN appliance.<br>NetOmni appliance sends policy updates to and receives network event data (metadata) from a NetDetector/NetVCR/LogWave appliance. Therefore, when the TOE is configured as NetDetector/NetVCR/LogWave, it acts as a TLS Server while receiving policy updates from NIKSUN appliance and when the TOE is configured as NetOmni, it acts as a TLS Server while receiving network event data (metadata) from NIKSUN appliance.<br>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. Though TLS 1.3 is not indicated for the FCS_TLSS_EXT.1.2 SFR above, the TSF will also deny all clients requestion connections for the TLS 1.3 protocol.<br>The TSF uses keys that are established using 2048-bit Diffie-Hellman or 256-bit ECDH parameters.<br>The TOE does not support session resumption based on session IDs or session tickets. |
| FCS_RBG_EXT.1 | The TOE performs random bit generation services in accordance with ISO/IEC 18031:2011 by Hash_DRBG (ACVP certs # A6781) and CTR_DRBG (AES) (ACVP certs # A67812). The TOE is seeded with at least 256 bits of entropy from JitterRNG which is seeded with conditioned random data from the Jitter RNG entropy source supported by the underlying Linux kernel. |
| FIA_AFL.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.<br>The TOE will lock out user accounts after a number of failed attempts set by the security administrator. Afterwards, the user cannot log back in until the account is unlocked by another administrator or until an administrator defined time period has elapsed. |

| Requirement | TSS Description |
|---|---|
| | Account unlocking by an Administrator is supported by both CLI and web GUI interfaces. Administrator defined time-based account unlocking is only supported by web GUI interface. An Administrator logging in locally using root account is not subject to account lockout. The CLI and Web GUI user accounts are maintained separately by the TOE. Therefore, the TOE tracks unsuccessful authentication attempts independently between GUI and CLI users. A failure or lockout on GUI will not affect CLI attempts or users. |
| FIA_PMG_EXT.1 | In the evaluated configuration the TOE supports passwords with a configurable minimum length of between 8 and 100 characters. The accepted characters include upper and lower case letters, numbers, and the special characters "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", and "+".<br>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 to 8 characters in order to match the evaluated configuration. For the CLI the root user has the ability to set the minimum character limit for passwords to 8 characters to gain access through the CLI in the /etc/pam.d/passwd file. |
| 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 banner on the CLI; which can be configured by the root user through the CLI.<br>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).<br>When a user utilizes the "Forgot Username/Password" feature on the web GUI login screen, NetOmni'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. |
| 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.<br>SSH public key authentication can be achieved for SSH and SCP sessions with the |

| Requirement | TSS Description |
|---|---|
| | following steps:<br>• 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.<br>• ssh-keygen -t ecdsa – This command generates a private/ public key pair in ~/.ssh in files id_ecdsa and id_ecdsa.pub for private and public keys respectively.<br>• Copy the public key to a remote server and append it to ~/.ssh/authorized_keys.<br>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. |
| 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. |
| FIA_X509_EXT.1/Rev<br>FIA_X509_EXT.2<br>FIA_X509_EXT.3 | The TOE uses X509 certificates as part of TLS and HTTPS protocols. The TOE provides ability for an administrator to generate a CSR and upload a CA signed server certificate, which will be used by the TOE as part of Web GUI and when the TOE is connecting to a remote Syslog server over TLS.<br>The TOE also provides the ability for an administrator to upload certificate chains as part of configuring remote IT entities. The TOE uses the corresponding certificate chain to validate the certificate presented by a remote IT entity as part of TLS handshake. The TOE checks the validity of a client's or server's certificate as part of TLS handshake for the following connections:<br>• when it connects with a NIKSUN appliance through HTTPS/TLS,<br>• when it connects to a Syslog Server for sending audit data over TLS,<br>• when it connects with an LDAP/AD Server for authenticating users over TLS, and<br>• when it connects to an SMTP Server to send emails over TLS.<br>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 basicConstraints 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 HTTPS/TLS and TLS. |

| Requirement | TSS Description |
|---|---|
| | The TOE uses X.509v3 certificates to support authentication for TLS and HTTPS connections in accordance with RFC 5280. The TOE performs revocation check as part of TLS handshake. 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 or determine the validity of a certificate, the TSF rejects the certificate.<br>The TOE can create a Certificate Signing Request (CSR) as specified by RFC 2986. The request includes the following information: Public Key, Common 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. |
| FMT_MOF.1/ManualUpdate | The TOE restricts the ability to perform manual updates to the root user via the CLI. There are no other methods for updating the TOE. |
| FMT_MTD.1/CoreData | The TOE restricts the ability to manage the TSF data to Security Administrators. None of the administrative functions are available to non-administrative users or administrators prior to authentication. 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 when root is accessed via sudo.<br>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 Administrators can also manage TOE's trust store for managing X.509 certificates used by the Web GUI and while communicating with remote IT entities via TLS. The ability to manage the trust stores is restricted to Security Administrators.<br>The only actions allowed before authentication are the use of the "forgot password" function for the web GUI and the display of the security banner for the web GUI and the CLI. |
| FMT_MTD.1/CryptoKeys | The TOE restricts the ability to manage the cryptographic keys to Security Administrators. The Security Administrator is able to manage the cryptographic keys (generating keys, importing keys, or deleting keys) that are used in TLS and SSH communications. These keys can be managed via Web GUI or CLI as part of following operations:<br>● HTTPS/TLS – CSR (keypair) generation, certificate import/export, Trust store management<br>● TLS/SSH session keys– as part of session establishment and termination<br>● SSH public key authentication – generate keypair, import/export public keys |

| Requirement | TSS Description |
|---|---|
| | • Zeroize - delete keys |
| 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 user, while for the web GUI it is fulfilled by the Admin 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:<br>• Ability to administer the TOE locally and remotely;<br>• Ability to configure the access banner;<br>• Ability to configure the session inactivity time before session termination or locking;<br>• Ability to update the TOE, and to verify the updates using a hash comparison prior to installing those updates;<br>• Ability to configure the authentication failure parameters for FIA_AFL.1;<br>• Ability to configure the SSH and TLS protocols, keys and connections;<br>• Ability to configure cryptographic functionality as defined in other SFRs in this document<br>• Ability to configure thresholds for SSH rekeying<br>• Ability to re-enable an Administrator account<br>• Ability to set the time which is used for time-stamps<br>• Ability to configure the reference identifier for the peer<br>• Ability to import X.509v3 certificates to the TOE's trust store |
| FMT_SMR.2 | There are two types of user accounts:<br>• those that access the TOE through the CLI, and<br>• those that access through the web GUI.<br>The TOE maintains the role of Security Administrator which is fulfilled by the vcr/root 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, and it is able to sudo to root. 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.<br>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:<br>CLI users:<br>• Username<br>• Password (SHA512)<br>• Full name (optional)<br>• SSH public key for remote CLI login<br>• User groups (note -- users can be in more than 1 group, including TSF management)<br>• User home directory<br>• User default shell<br>• Last successful authentication time |

| Requirement | TSS Description |
|---|---|
| | • Number of failed authentications<br>• Status (locked or unlocked)<br><br>Web GUI users:<br>• Username<br>• Password (SHA256)<br>• Full name<br>• Description (optional)<br>• Email<br>• Phone (optional)<br>• User group (note -- users can only be in 1 group)<br>• User rights (note -- ACL defined by group)<br>• Password policy (note -- defined by group)<br>• Status (enabled or disabled)<br>• Last password change date<br>• Password expiration date<br>• Password expiration notification policy (how many days before expiration for reminders)<br>• Password age<br>• Old passwords (SHA256 list)<br>• First authentication failure time<br>• Last authentication failure time<br>• Number of failed authentications<br><br>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.<br><br>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:<br>• administrator,<br>• Account Administrator,<br>• Advanced Users, and<br>• User.<br><br>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 its 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. |
| FPT_SKP_EXT.1 | The (EC) Diffie-Hellman Shared Secret, (EC) Diffie Hellman private and public parameters, TLS session keys, and SSH session keys are stored in volatile memory (RAM) and are not accessible by any user. The SSH private and public keys and SSL |

| Requirement | TSS Description |
|---|---|
| | server key are stored on the local filesystem and RAM. The keys stored on the local file system are protected by access controls. Only authorized users can access keys. Because the 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.<br>The root user has the ability to delete SSH keys using the "rm <keyfile>" command. |
| 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, web GUI users, and accounts with supportnet.niksun.com, and each registered NIKSUN appliance. The vcr user's password is stored in the OS's password file which has configurable hashing and in the evaluated configuration uses SHA-512. Web GUI user passwords are stored in an internal Postgres Database which is hashed using SHA-256. The connection between the TOE and a NIKSUN appliance requires the username/password of an admin for each device. The vcr user is able to view the locations of these passwords but can only see their hash values.<br>A password is stored for connecting to supportnet.niksun.com for downloading signature updates which are used for NetOmni's functionality that is outside the scope of the evaluation. This password is a non-administrative password and is protected by AES-128 encryption that uses a key that is protected by access control and is only accessible by authorized users. |
| FPT_TST_EXT.1 | During power-on the TOE performs an integrity check on all firmware packages using SHA-256.<br>All of the cryptographic algorithms used by the TOE are tested during power-on.<br>The OpenSSL FIPS_mode_set() function performs all self-tests of its supported cryptographic algorithms with no operator intervention required, returning a "1" if all self-tests succeed, and a "0" otherwise. If any component of the self-test fails an internal flag is set to prevent subsequent invocation of any cryptographic function calls.<br>Bouncy Castle performs all self-tests as its various cryptographic Java classes (engines) are instantiated via the Java class loader. Bouncy Castle classes that perform self-tests will fail to load in the event of self-test failures. The FipsStatus.isReady() method is triggered when any cryptographic functionality is called and will throw an Error exception if the module is in an error state preventing subsequent invocation of any cryptographic function calls. |
| 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). The root user can also configure all aspects of the clock using the local or remote CLI. The TOE uses the clock for several purposes in the time format selected including for:<br>• Audit records<br>• Inactivity timeout for local CLI sessions<br>• Inactivity timeout for remote CLI sessions |

| Requirement | TSS Description |
|---|---|
| | • Inactivity timeout for remote web GUI sessions<br>• X.509 certificate validation |
| 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.<br><br>The TOE software is updated by the root 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 root 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.<br><br>When an update is performed, the TOE conducts an RSA hash verification on the system image. The TOE will verify the image integrity and RSA digital signature using a NIKSUN public key for the update image before installation. The Security Administrator must manually confirm that the update's hash and the published hash must match prior to installing the update. If the two hashes match the Administrator continues with the install 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.. |
| 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. The web GUI timeout can be configured by the root user via the CLI by modifying the session_timeout variable in the "/usr/local/niksun/apps/etc/apps.conf" file. |
| FTA_SSL.4 | The root or vcr user accessing the TOE remotely (vcr) or locally through the CLI (vcr or root) 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. |
| 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 root user via the CLI in /etc/profile TMOUT value with a default of 600 (10 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 root 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. |
| FTA_TAB.1 | There are three possible ways to log in to the TOE:<br>• local CLI,<br>• remote CLI, and<br>• a remote web GUI. |

| Requirement | TSS Description |
|---|---|
| | When logging in locally or remotely through the CLI, the pre-authentication banner is displayed. It can be configured by the root 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. |
| FTP_ITC.1 | The TOE connects with a Syslog Server, LDAP/AD Server, SMTP Server, SCP Server, and instances of NIKSUN appliances to send and receive data. All devices reside in the operational environment and communicates with the TOE via trusted channels.<br>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 connecting 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 between 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 the TOE, to connect to the NIKSUN appliance, to send policy updates and receive network metadata.<br>The TOE initiates communication with each of the servers in the operational environment using the protocols discussed in the relevant SFRs to protect the data traversing the channel from disclosure and/or modification. |
| 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.<br>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. |

## 6.1 CAVP Algorithm Certificate Details

Each of these cryptographic algorithms have been validated as identified in the table below.

**Table 14 – CAVP Algorithm Certificate References**

| ALGORITHM | CAVP CERT # | STANDARD | OPERATION | SFR |
|---|---|---|---|---|
| RSA | A6781<br>A6782 | FIPS 186-4 | Key Generation<br>Signature Generation/ Verification<br>Mod lengths: 2048 (bits) | FCS_CKM.1<br>FCS_COP.1/SigGen |
| ECDSA | A6781<br>A6782 | FIPS 186-4 | Key Generation / Validation<br>Signature Generation/ Signature Verification<br>Curves: P-256 | FCS_CKM.1<br>FCS_CKM.2.1 |
| SP 800-90 DRBG | A6781 | SP 800-90A | Random Bit Generation<br>Hash_DRBG (AES-512) | FCS_RBG_EXT.1 |
| | A6782 | | CTR_DRBG (AES-256) | |
| SHS | A6781<br>A6782 | ISO/IEC 10118-3:2004 | Hashing<br>SHA-1, SHA-256, SHA-384, SHA-512 | FCS_COP.1/Hash |
| HMAC-SHS | A6781<br>A6782 | ISO/IEC 9797-2:2011 | Keyed-Hashing<br>HMAC-SHA-1,<br>HMAC-SHA-256,<br>HMAC-SHA-384,<br>HMAC-SHA-512 | FCS_COP.1/ KeyedHash |
| AES | A6781<br>A6782 | AES specified in ISO 18033-3<br>CTR as specified in ISO 10116<br>GCM specified in ISO 19772 | Encryption/ Decryption<br>CTR, GCM<br>Key Lengths: 128, 256 | FCS_COP.1/ DataEncryption |
| KAS-ECC-SSC | A6781<br>A6782 | SP 800-56A Revision 3 | Key Establishment<br>Curves: P-256 | FCS_CKM.2.1 |

## 6.2 Cryptographic Key Destruction

The table below describes the key zeroization provided by the TOE and as referenced in FCS_CKM.4.

**Table 15 – Keys/CSPs Destruction**

| Keys/CSPs | Purpose | Storage Location | Method of Zeroization |
|---|---|---|---|
| Diffie Hellman private exponent | DH Key | RAM | One-pass overwrite with zeroes |
| Diffie Hellman public exponent | DH Key | RAM | One-pass overwrite with zeroes |
| DH Shared Secret | DH Key | RAM | One-pass overwrite with zeroes |
| EC Diffie Hellman private parameters | ECDH Key | RAM | One-pass overwrite with zeroes |
| EC Diffie Hellman public parameters | ECDH Key | RAM | One-pass overwrite with zeroes |
| ECDH Shared Secret | ECDH Key | RAM | One-pass overwrite with zeroes |
| SSH Private Keys | RSA/ECDSA Private Keys | ACL protected directory | Three-pass overwrite with zeroes |
| SSH Private Keys | RSA/ECDSA Private Keys | RAM | One-pass overwrite with zeroes |
| SSH Public Keys | RSA/ECDSA Public Keys | n/a - public | Three-pass overwrite with ones and zeroes |
| SSL Server Key | RSA Private Key | Local filesystem | Three-pass overwrite with ones and zeroes |
| SSL Server Key | RSA Private Key | RAM | One-pass overwrite with zeroes |
| SSH Session Encryption Keys | AES Keys | RAM | One-pass overwrite with zeroes |
| SSH Session Integrity Keys | HMAC Keys | RAM | One-pass overwrite with zeroes |
| TLS Session Encryption Keys | AES Keys | RAM | One-pass overwrite with zeroes |
| TLS Session Integrity Keys | HMAC Keys | RAM | One-pass overwrite with zeroes |

# 7

# 7 Acronym Table

Acronyms should be included as an Appendix in each document.

**Table 16 – Acronyms**

| Acronym | Definition |
|---|---|
| **AES** | Advanced Encryption Standard |
| **CC** | Common Criteria |
| **CRL** | Certificate Revocation List |
| **DTLS** | Datagram Transport Layer Security |
| **EP** | Extended Package |
| **GUI** | Graphical User Interface |
| **IP** | Internet Protocol |
| **NDcPP** | Network Device Collaborative Protection Profile |
| **NIAP** | Nation Information Assurance Partnership |
| **OCSP** | Online Certificate Status Protocol |
| **PP** | Protection Profile |
| **RSA** | Rivest, Shamir, & Adleman |
| **SFR** | Security Functional Requirement |
| **SSH** | Secure Shell |
| **ST** | Security Target |
| **TOE** | Target of Evaluation |
| **TLS** | Transport Layer Security |
| **TSS** | TOE Summary Specification |