



# Klas Fastnet Series Switches KlasOS 5.3 Security Target

July 16, 2021  
Version 1.7

Prepared by:  
Acumen Security  
2400 Research Blvd  
Rockville MD 20850

## Table of Contents

1	Security Target Introduction .....	5
1.1	Security Target and TOE Reference .....	5
1.2	TOE Overview .....	5
1.2.1	TOE Product Type .....	5
1.3	TOE Architecture .....	5
1.4	TOE Evaluated Configuration .....	6
1.5	Physical Scope of the TOE .....	7
1.6	Logical Scope of the TOE .....	7
1.6.1	Security Audit .....	7
1.6.2	Cryptographic Operations .....	7
1.6.3	Identification and Authentication .....	10
1.6.4	Security Management .....	10
1.6.5	Protection of the TSF .....	10
1.6.6	TOE Access .....	11
1.6.7	Trusted Path/Channels .....	11
1.7	Excluded Functionality .....	11
1.8	TOE Documentation .....	11
1.9	Other References .....	11
2	Conformance Claims .....	12
2.1	CC Conformance .....	12
2.2	Protection Profile Conformance .....	12
2.3	Conformance Rationale .....	12
2.4	NIAP Technical Decisions .....	12
3	Security Problem Definition .....	13
3.1	Threats .....	13
3.2	Assumptions .....	15
3.3	Organizational Security Policy .....	16
4	Security Objectives .....	17
4.1	Security Objectives for the Operational Environment .....	17
5	Security Requirements .....	18
5.1	Conventions .....	18
5.2	TOE Security Functional Requirements .....	18
5.2.1	Class: Security Audit (FAU) .....	19
5.2.2	Class: Cryptographic Support (FCS) .....	21
5.2.3	Class: Identification and Authentication (FIA) .....	24
5.2.4	Class: Security Management (FMT) .....	25

5.2.5	Class: Protection of the TSF (FPT) .....	26
5.2.6	Class: TOE Access (FTA) .....	26
5.2.7	Class: Trusted Path/Channels (FTP).....	27
5.3	TOE SFR Dependencies Rationale for SFRs .....	27
5.4	Security Assurance Requirements .....	27
5.5	Rationale for Security Assurance Requirements.....	28
5.6	Assurance Measures .....	28
6	TOE Summary Specification .....	30
6.1	Key Storage and Zeroization .....	37
7	Terms and Definitions .....	39
8	References.....	40

## Revision History

Version	Date	Description
0.1	5/29/2020	Initial version
0.2	6/15/2020	Review comments addressed
0.3	08/21/2020	Updated formatting
0.4	09/04/2020	Revised structure
0.5	10/26/2020	Added new TDs and revised SFRs
0.6	11/02/2020	Updated CAVP list
0.7	11/04/2020	Updated Tables and Diagrams
0.8	11/09/2020	Added new TDs
0.9	11/12/2020	Minor updates based on internal reviews
1.0	11/24/2020	Updated TOE identifier and TOE Summary Specification
1.1	11/27/2020	Addressing TSS requirements
1.2	3/5/2021	Addressing validator check in comments
1.3	4/12/2021	Added new TDs
1.4	5/13/2021	Finalization of the ST
1.5	5/25/2021	Updated new TDs
1.6	6/21/2021	Addressing QA comments
1.7	7/16/2021	Addressing validator comments

# 1 Security Target Introduction

## 1.1 Security Target and TOE Reference

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

Category	Identifier
ST Title	Klas Fastnet Series Switches KlasOS 5.3 Security Target
ST Version	1.7
ST Date	7/16/2021
ST Author	Acumen Security, LLC.
TOE Identifier	Klas Fastnet Series Switches KlasOS 5.3
TOE Software Version	5.3.5
TOE Developer	Klas Telecom Inc.
Key Words	Network Device, Klas

**Table 1: TOE/ST Identification**

## 1.2 TOE Overview


The TOE is the Klas Fastnet Series Switches KlasOS 5.3. (herein referred to as the TOE) It runs the KlasOS firmware, which provides connectivity to multiple devices contained within the same network segment. A real-time clock is present on all KlasOS devices. Authentication can be performed locally or over a trusted channel using SSH. All logs can be securely transferred to a syslog server. KlasOS provides a Command Line Interface (CLI) for device configuration. The Klas Fastnet switches range of products provide expandable, enterprise-grade, rugged mobility solutions.

### 1.2.1 TOE Product Type

The TOE is classified as a network device which is composed of hardware and software that offers scalable solutions to its end-users. It satisfies all the criterion needed to meet the collaborative Protection Profile for Network Devices, Version 2.2e [NDcPP v2.2e] requirements.

## 1.3 TOE Architecture

The TOE consists of the following models:

Hardware Platforms	Specifications
<p>Klas Voyager TDC 10G Switch</p> 	<ul style="list-style-type: none"><li>• 512 GB RAM</li><li>• 32 Physical CPU Cores</li><li>• Up to 32 TBs of raw storage</li><li>• 10 GB/s networking</li><li>• Ten 10-Gigabit Switch ports (4 available as copper or SFP to support fiber-optic connectivity)</li><li>• 1 gigabit management port</li><li>• 1 VIK slot (for removable storage)</li><li>• 1 console port</li><li>• Processor: Marvell Prestera 98DX8212 (ARM v7)</li></ul>


Hardware Platforms	Specifications
<p>Klas Voyager TDC 12GG Switch</p> 	<ul style="list-style-type: none"> <li>• Small form factor variant of the Voyager TDC Switch, the first 10 Gb/s switch available for the tactical market</li> <li>• 121 Gb/s backplane for line-speed processing simultaneously on all ports</li> <li>• 40 Gb/s trunk for speeds</li> <li>• 1x 40 Gb/s QSFP+ high-speed uplink port or 4x 10-Gigabit SFP+ ports (based on breakout cable selection)</li> <li>• 8x 1- Gb/s SFP+ ports</li> <li>• 1 Gigabit management port,</li> <li>• 1 VIK slot (for removable storage)</li> <li>• 1 console port</li> <li>• Processor: Marvell Prestera 98DX8212 (ARM v7)</li> </ul>

Table 1: TOE Models

### 1.4 TOE Evaluated Configuration

The TOE also supports (sometimes optionally) secure connectivity with several other IT environment devices, including,

Component	Required	Usage/Purpose Description for TOE performance
Management Workstation/SSH Client	Yes	This includes any IT Environment Management workstation with a SSH client installed that is used by the TOE administrator to support TOE administration through SSH protected channel. Any SSH client that supports SSHv2 may be used.
Syslog server	Yes	The syslog audit server is used for remote storage of audit records that have been generated by and transmitted from the TOE.

Table 2: IT Environment Components

### Klas Voyager TDC Switch Deployment Diagram

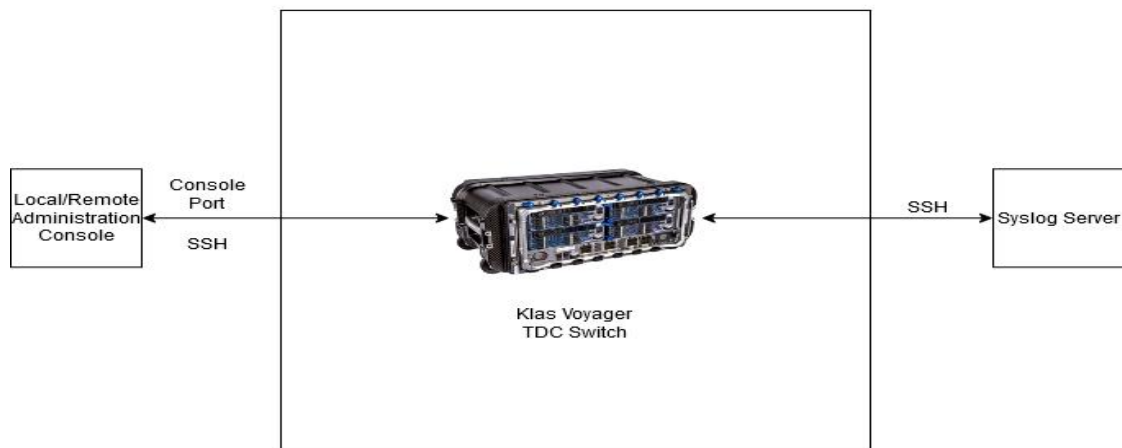


Figure 1: Klas Voyager TDC Deployment Diagram

## 1.5 Physical Scope of the TOE

The TOE boundary is the hardware appliance which is comprised of hardware and software components. It is deployed in an environment which contains the various IT components as depicted in Figure 1.

The TOE consists of the following devices:

- Klas Voyager TDC 10G Switch and Klas Voyager TDC 12GG Switch running KlasOS v 5.3.5 on Marvell Prestera 98DX8212 (ARM v7) processor.

## 1.6 Logical Scope of the TOE

The TOE implements the following security functional requirements:

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

Each of these security functionalities are covered in more detail below.

### 1.6.1 Security Audit

The TOE generates audit events for all start-up and shutdown functions as well as all auditable events specified in Table 13 'Auditable Events'. Audit events are also generated for management actions specified in FAU\_GEN.1. The TOE can store audit records locally and export them to an external syslog server using SSHv2. Each audit record contains the date and time of the event, type of event, subject identity, and other relevant data of the event. Only a Security Administrator can enable logging to a syslog server.

### 1.6.2 Cryptographic Operations

The TOE provides cryptographic support for the services described in Table 3. The related CAVP validation details are provided in Table 4. The operating system used is Klas OS v5.3.5. The TOE leverages OpenSSL 1.0.1u for cryptographic algorithms and OpenSSH 7.7p1 for SSH.

Cryptographic Method	Usage
FCS_CKM.1 Cryptographic Key Generation	<ul style="list-style-type: none"><li>• Cryptographic key generation conforming to FIPS PUB 186-4 "Digital Signature Standard (DSS)", Appendix B.3.</li><li>• RSA Key sizes supported are 2048-bit or greater.</li><li>• Cryptographic key generation conforming to FIPS PUB 186-4 "Digital Signature Standard (DSS)", Appendix B.4.</li><li>• Elliptic NIST curves supported are: P-256 and P-384.</li><li>• 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.</li></ul>
FCS_CKM.2 Cryptographic Key Establishment	<ul style="list-style-type: none"><li>• RSA-based key establishment conforming to RSAES-PKCS1-v1_5 as specified in</li></ul>

Cryptographic Method	Usage
	<p>Section 7.2 of RFC 8017, Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1</p> <ul style="list-style-type: none"> <li>• Elliptical curve-based establishment conforming to NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”;</li> <li>• 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]].</li> </ul>
FCS_CKM.4 Cryptographic Key Destruction	<ul style="list-style-type: none"> <li>• Refer to Table 17 for Key Zeroization details.</li> </ul>
FCS_COP.1/DataEncryption	<ul style="list-style-type: none"> <li>• AES encryption and decryption conforming to CBC as specified in ISO 10116.</li> <li>• AES key size supported is 128 and 256 bits</li> <li>• AES mode supported is CBC.</li> </ul>
FCS_COP.1/SigGen	<ul style="list-style-type: none"> <li>• RSA digital signature algorithm conforming to 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.</li> <li>• RSA key sizes supported are: 2048 and 3072 bits.</li> <li>• Elliptical curve digital signature algorithm conforming to FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing NIST curves ISO/IEC 14888-3, Section 6.4.</li> <li>• Elliptical curve key sizes supported are 256, and 384 bits.</li> </ul>
FCS_COP.1/Hash	<ul style="list-style-type: none"> <li>• Cryptographic hashing services conforming to ISO/IEC 10118-3:2004.</li> <li>• Hashing algorithms supported are: SHA-1, SHA-256, SHA-384, and SHA-512.</li> <li>• Message digest sizes supported are: 160, 256, 384, and 512 bits.</li> </ul>
FCS_COP.1/KeyedHash	<ul style="list-style-type: none"> <li>• Keyed-hash message authentication conforming to ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm2”.</li> <li>• Keyed hash algorithm supported are: HMAC-SHA1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512.</li> </ul>



Cryptographic Method	Usage
	<ul style="list-style-type: none"> <li>• Key sizes supported are: 160, 256, 384 and 512 bits.</li> <li>• Message digest sizes supported are: 160, 256, 384, and 512 bits.</li> </ul>
FCS_RBG_EXT.1 Random Bit Generation	<ul style="list-style-type: none"> <li>• Random number generation conforming to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”</li> <li>• The TOE leverages CTR_DRBG(AES)</li> <li>• CTR_DRBG seeded with a minimum of 256 bits of entropy.</li> </ul>
FCS_SSHC_EXT.1 SSH Client Protocol	<ul style="list-style-type: none"> <li>• The TOE supports SSH v2 protocol complaint to the following RFCs:4251, 4252, 4253, 4254, 5656, and 6668.</li> <li>• The TOE supports public-key based authentication.</li> <li>• SSH public-key authentication uses ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384.</li> <li>• SSH transport uses the following encryption algorithms: aes128-cbc, aes256-cbc.</li> <li>• Packets greater than 33,292 bytes in an SSH transport connection are dropped.</li> <li>• SSH transport uses the following data integrity MAC algorithms: hmac-sha1, hmac-sha2-256, and hmac-sha2-512</li> <li>• Key exchange algorithms supported are: diffie-hellman-group14-sha1, ecdh-sha2-nistp256 and ecdh-sha2-nistp384.</li> <li>• The TOE ensures that during SSH connections, the same session keys are used for a threshold of no longer than one hour and no more than one gigabyte of transmitted data.</li> <li>• The TOE 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 as described in RFC 4251 Section 4.1.</li> </ul>
FCS_SSHS_EXT.1 SSH Server Protocol	<ul style="list-style-type: none"> <li>• The TOE supports SSH v2 protocol complaint to the following RFCs:4251, 4252, 4253, 4254, 5656, and 6668.</li> <li>• The TOE supports password-based and public-key-based authentication.</li> <li>• SSH public-key authentication uses ssh-rsa, ecdsa-sha2-nistp256 and ecdsa-sha2-nistp384.</li> </ul>

Cryptographic Method	Usage
	<ul style="list-style-type: none"> <li>SSH transport uses the following encryption algorithms: aes128-cbc, and aes256-cbc.</li> <li>Packets greater than 33,292 bytes in an SSH transport connection are dropped.</li> <li>SSH transport uses the following data integrity MAC algorithms: hmac-sha1, hmac-sha2-256, hmac-sha384, and hmac-sha2-512</li> <li>Key exchange algorithms supported are: diffie-hellman-group14-sha1, ecdh-sha2-nistp256 and ecdh-sha2-nistp384.</li> <li>The TOE ensures that during SSH connections, the same session keys are used for a threshold of no longer than one hour and no more than one gigabyte of transmitted data.</li> </ul>

**Table 3: TOE Cryptography Implementation**

Cryptographic Algorithms	CAVP
RSA	C2000
ECDSA	C2000
DRBG	C2000
SHS	C2000
HMAC-SHS	C2000
AES	C2000
KAS	C2000

**Table 4: Cryptographic Algorithm Certificates**

### 1.6.3 Identification and Authentication

All users must be authenticated by the TOE prior to carrying out any administrative actions. The TOE supports password-based and public-key based authentication. An administrator can set a minimum password length on the TOE which can be a minimum of 15 characters.

### 1.6.4 Security Management

The TOE supports local and remote management of its security functions including:

- Local console CLI administration
- Remote CLI administration via SSHv2
- Configurable banner displayable at login
- Timeouts to terminate administrative sessions after a set period of inactivity
- Timed user lockout after multiple failed authentication attempts
- Configurable authentication failure parameters
- Re-enabling locked accounts
- Configurable cryptographic parameters

The administrative user can perform all the above security related management functions.

### 1.6.5 Protection of the TSF

The TOE protects all passwords, pre-shared keys, symmetric keys, and private keys from unauthorized disclosure. Passwords are stored as SHA 512 hashes. The TOE executes self-tests during initial start-up to

ensure correct operation and enforcement of its security functions. The TOE internally maintains the date and time. An administrator can install software updates to the TOE after they are verified using a digital signature mechanism.

### 1.6.6 TOE Access

The TOE displays a customizable banner before any administrative session can be established with it. The TOE will terminate local or remote interactive sessions after a specified period of session inactivity configured by an administrator. An administrator can terminate their own interactive local or remote sessions.

### 1.6.7 Trusted Path/Channels

The TOE supports SSH for secure communications with authorized IT entities such as syslog servers. The TOE supports SSHv2 (remote CLI) for secure remote administration.

## 1.7 Excluded Functionality

The following functionalities are excluded from the evaluation:

Excluded Functionality	Exclusion Rationale
SNMP	Not within the scope of evaluation
NTP	Not within the scope of evaluation

Table 5: Excluded Functionality

## 1.8 TOE Documentation

The table below lists the TOE guidance documentation. CC and ST are provided in .pdf form on the NIAP portal.

Reference	Title	Version	Date
[CC]	Klas Fastnet Series Switches KlasOS 5.3 Common Criteria Configuration Guide	1.0	August 9, 2021

Table 6: Documentation for the TOE

## 1.9 Other References

- collaborative Protection Profile for Network Devices, Version 2.2e [NDcPP v2.2e]

## 2 Conformance Claims

### 2.1 CC Conformance

The TOE and ST are compliant with the Common Criteria (CC) Version 3.1, Revision 5, dated: April 2017.

This TOE is conformant to:

- 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: Part 2 extended
- Common Criteria for Information Technology Security Evaluations Part 3, Version 3.1, Revision 5, April 2017: Part 3 conformant

### 2.2 Protection Profile Conformance

This TOE is conformant to:

- Collaborative Protection Profile for Network Devices, Version 2.2e [NDcPP v2.2e]

### 2.3 Conformance Rationale

This Security Target provides exact conformance to Version 2.2e of the Collaborative Protection Profile for Network Devices. The security problem definition, security objectives and security requirements in this Security Target are all taken from the Protection Profile performing only operations defined in Section 3.

### 2.4 NIAP Technical Decisions

NIAP Technical Decisions for NDcPP v2.2e (TDs)		
Technical Decisions	Applicable	Exclusion Rationale (if applicable)
TD0592: NIT Technical Decision for Local Storage of Audit Records	Yes	
TD0591: NIT Technical Decision for Virtual TOEs and hypervisors	No	Not applicable as the TOE is not virtual.
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	No	Not applicable as TLSC is functionality is not claimed.
TD0571 – NiT Technical Decision for Guidance on how to handle FIA_AFL.1	Yes	
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	No	Not applicable as DTLS functionality is not claimed.

NIAP Technical Decisions for NDcPP v2.2e (TDs)		
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	No	Not applicable as TLSS functionality is not claimed.
TD0555: NIT Technical Decision for RFC Reference incorrect in TLSS Test	No	Not applicable as TLSS functionality is not claimed.
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	Not applicable as DTLS functionality is not claimed.
TD0538 – NIT Technical Decision for Outdated link to allowed with list	Yes	
TD0537 – NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	No	Not applicable as X.509 functionality is not claimed.
TD0536 – NIT Technical Decision for Update Verification Inconsistency	Yes	
TD0528 – NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	No	Not applicable as FCS_NTP_EXT.1.4 is not claimed.
TD0527 – Updates to Certificate Revocation Testing (FIA_x509_EXT.1)	No	Not applicable as X.509 functionality is not claimed.

Table 7: NIAP Technical Decisions

### 3 Security Problem Definition

The security problem definition has been taken from [NDcPP v2.2e] and is reproduced here for the convenience of the reader. The security problem is described in terms of the threats that the TOE is expected to address, assumptions about the operational environment, and any organizational security policies that the TOE is expected to enforce.

#### 3.1 Threats

The threats for the Network Device are grouped according to functional areas of the device in the sections below.

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

ID	Threat
	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 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

ID	Threat
	device. Having privileged access to the device provides the attacker unfettered access to the network traffic and may allow them to take advantage of any trust relationships with other Network Devices.
T.SECURITY_FUNCTIONALITY_FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.

**Table 8: Threats**

### 3.2 Assumptions

This section describes the assumptions made in identification of the threats and security requirements for Network Devices. The Network Device is not expected to provide assurance in any of these areas, and as a result, requirements are not included to mitigate the threats associated with them. The table below describes conditions which are assumed to exist in the environment where the TOE is deployed. These assumptions are referenced from the PP and remain unchanged from their original source.

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	<p>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).</p> <p>In the case of vNDs, the VS is considered part of the TOE with only one vND instance for each physical hardware platform. The exception being where components of the distributed TOE run inside more than one virtual machine (VM) on a single VS. There are no other guest VMs on the physical platform providing non-Network Device functionality.</p>
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 NDcPP. It is assumed that this

ID	Assumption
	protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall).
A.TRUSTED_ADMINISTRATOR	<p>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.</p> <p>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).</p>
A.REGULAR_UPDATES	The Network Device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_CREDENTIALS_SECURE	The Administrator's credentials (private key) used to access the Network Device are protected by the platform on which they reside.
A.RESIDUAL_INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

**Table 9: Assumptions**

### 3.3 Organizational Security Policy

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. For the purposes of this cPP, a single policy is described in the section below.

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.

**Table 10: Organizational Security Policy**



## 4 Security Objectives

The security objectives have been taken from [NDcPP v2.2e] and are reproduced here for the convenience of the reader.

### 4.1 Security Objectives for the Operational Environment

The following subsections describe objectives for the Operational Environment.

ID	Objective for the Operation 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_ADMIN	<p>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.</p> <p>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.</p>
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.

**Table 11: Security Objectives for the Operational Environment**

## 5 Security Requirements

This section identifies the Security Functional Requirements for the TOE and/or Platform. The Security Functional Requirements contained within this security target are directly sourced from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, dated April 2017 and all international interpretations.

### 5.1 Conventions

The CC defines operations on Security Functional Requirements: assignments, selections, assignments within selections and refinements. This document uses the following font conventions to identify the operations defined by the CC:

- Assignment: Indicated with *italicized text*;
- Refinement made by PP author: Indicated with **bold text** and ~~strikethroughs~~, if necessary;
- Selection: Indicated with underlined text;
- Assignment within a Selection: Indicated with *italicized and underlined text*;
- Iteration: Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3).
- Where operations were completed in the PP itself, the formatting used in the PP has been retained.

Explicitly stated SFRs are identified by having a label 'EXT' after the requirement name for TOE SFRs. Formatting conventions outside of operations matches the formatting specified within the PP.

### 5.2 TOE Security Functional Requirements

This section identifies the Security Functional Requirements for the TOE. The TOE Security Functional Requirements that appear below in Table 12 are described in more detail in the succeeding subsections.

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 (Refinement)
FCS_CKM.2	Cryptographic Key Establishment (Refinement)
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_RBG_EXT.1	Random Bit Generation
FCS_SSHC_EXT.1	SSH Client Protocol
FCS_SSHS_EXT.1	SSH Server Protocol
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
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_MOF.1/Functions	Management of Security Functions Behaviour
FMT_MOF.1/Services	Management of Security Functions Behaviour
FMT_SMF.1	Specification of Management Functions

Requirement	Description
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_TUD_EXT.1	Trusted Update
FPT_STM_EXT.1	Reliable Time Stamps
FTA_SSL_EXT.1	TSF-initiated Session Locking
FTA_SSL.3	TSF-initiated Termination
FTA_SSL.4	User-initiated Termination
FTA_TAB.1	Default TOE Access Banner
FTP_ITC.1	Inter-TSF trusted channel
FTP_TRP.1/Admin	Trusted Path

**Table 12: TOE Security Functional Requirements**

### 5.2.1 Class: Security Audit (FAU)

#### FAU\_GEN.1 Audit Data Generation

**FAU\_GEN.1.1** The TSF shall be able to generate an audit record of the following auditable events:

- a) Start-up and shut-down of the audit functions;
- b) All auditable events for the not specified level of audit; and
- c) *All administrative actions comprising:*
  - *Administrative login and logout (name of user account shall be logged if individual user accounts are required for Administrators).*
  - *Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).*
  - *Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).*
  - *Resetting passwords (name of related user account shall be logged).*
  - *[no other actions];*
- d) *Specifically defined auditable events listed in Table 13.*

**FAU\_GEN.1.2** The TSF shall record within each audit record, at least the following information:

- a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
- b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, *information specified in column three of Table 13.*

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FCS_RBG_EXT.1	None.	None.
FCS_SSHC_EXT.1	Failure to establish an SSH Session	Reason for failure
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
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.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MOF.1/Functions	None.	None.
FMT_MTD.1/CoreData	None.	None.
FMT_MTD.1/CryptoKeys	None.	None.
FMT_MTD.1/Services	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_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1 (if "terminate the session" is selected)	The termination of a local session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FTP_ITC.1	<ul style="list-style-type: none"> <li>• Initiation of the trusted channel.</li> <li>• Termination of the trusted channel.</li> <li>• Failure of the trusted channel functions.</li> </ul>	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	<ul style="list-style-type: none"> <li>• Initiation of the trusted path.</li> <li>• Termination of the trusted path.</li> <li>• Failure of the trusted path functions</li> </ul>	None.

Table 13: Auditable Events

## FAU\_GEN.2 User Identity Association

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

## FAU\_STG\_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: [Delete the current logfile and start a new one]] when the local storage space for audit data is full.

## 5.2.2 Class: Cryptographic Support (FCS)

### FCS\_CKM.1 Cryptographic Key Generation (For Asymmetric Keys)

**FCS\_CKM.1.1: Refinement:** The TSF shall generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm: [

- RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;
- ECC schemes using "NIST curves" [P-256, P-384] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;
- 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*].

### FCS\_CKM.2 Cryptographic Key Establishment (Refinement)

**FCS\_CKM.2.1** The TSF shall **perform** cryptographic **key establishment** in accordance with a specified cryptographic key **establishment** method: [

- RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1\_5 as specified in Section 7.2 of RFC 3447, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1";
- 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*].

#### **FCS\_CKM.4 Cryptographic Key Destruction**

**FCS\_CKM\_EXT.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 [
  - o logically addresses the storage location of the key and performs a [single] overwrite consisting of [a new value of the key]
  - o instructs a part of the TSF to destroy the abstraction that represents the key]

that meets the following: No Standard.

#### **FCS\_COP.1/DataEncryption Cryptographic Operation (AES Data Encryption/Decryption)**

**FCS\_COP.1.1/DataEncryption** The TSF shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm *AES used in [CBC] mode* and cryptographic key sizes [*128 bits, 256 bits*] that meet the following: *AES as specified in ISO 18033-3, [CBC as specified in ISO 10116].*

#### **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 , 3072 bits]
- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256 bits, 384 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,
- For ECDSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" [P-256, P-384]; ISO/IEC 14888-3, Section 6.4

].

#### **FCS\_COP.1/Hash Cryptographic Operation (Hash Algorithm)**

**FCS\_COP.1.1/Hash** The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm [SHA-1, SHA-256, SHA-384, SHA-512] and cryptographic key sizes ~~[assignment: cryptographic key sizes]~~ and **message digest sizes** [160, 256, 384, 512] bits that meet the following: ISO/IEC 10118-3:2004.

#### **FCS\_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)**

**FCS\_COP.1.1/KeyedHash** The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm [HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512] and cryptographic key sizes [160, 256, 384, and 512 bits] and **message digest sizes** [160, 256, 384, 512] bits that meet the following: ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”.

#### **FCS\_RBG\_EXT.1 Random Bit Generation**

**FCS\_RBG\_EXT.1.1** The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [CTR\_DRBG (AES)].

**FCS\_RBG\_EXT.1.2** The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [5] [software-based noise source] with a minimum of [256 bits] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”, of the keys and hashes that it will generate.

#### **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, [5656, 6668].

**FCS\_SSHC\_EXT.1.2** The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, [no other method].

**FCS\_SSHC\_EXT.1.3** The TSF shall ensure that, as described in RFC 4253, packets greater than [33,292] bytes in an SSH transport connection are dropped.

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

**FCS\_SSHC\_EXT.1.5** The TSF shall ensure that the SSH public-key based authentication implementation uses [ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384] as its public key algorithm(s) and rejects all other public key algorithms.

**FCS\_SSHC\_EXT.1.6** The TSF shall ensure that the SSH transport implementation uses [hmac-sha1, hmac-sha2-256, hmac-sha2-512] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

**FCS\_SSHC\_EXT.1.7** The TSF shall ensure that [diffie-hellman-group14-sha1, ecdh-sha2-nistp256] and [ecdh-sha2-nistp384] 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.

#### **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, [5656, 6668].

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

**FCS\_SSHS\_EXT.1.3** The TSF shall ensure that, as described in RFC 4253, packets greater than [33,292] bytes in an SSH transport connection are dropped.

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

**FCS\_SSHS\_EXT.1.5** The TSF shall ensure that the SSH public-key based authentication implementation uses [ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2nistp384] as its public key algorithm(s) and rejects all other public key algorithms.

**FCS\_SSHS\_EXT.1.6** The TSF shall ensure that the SSH transport implementation uses [hmac-sha1, hmac-sha2-256 and hmac-sha2-512] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

**FCS\_SSHS\_EXT.1.7** The TSF shall ensure that [diffie-hellman-group14-sha1, ecdh-sha2-nistp256] and [ecdh-sha2-nistp384] 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.

### 5.2.3 Class: Identification and Authentication (FIA)

#### FIA\_AFL.1 Authentication Failure Management

**FIA\_AFL.1.1** The TSF shall detect when an Administrator configurable positive integer within [1-255] 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 [manual account unlocking] is taken by an Administrator].

#### 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 [15] and [128] characters.

#### 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;
- [Respond to ICMP requests;]

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



## 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.

## FIA\_UAU.7 Protected Authentication Feedback

FIA\_UAU.7.1 The TSF shall provide only *obscured feedback* to the administrative user while the authentication is in progress **at the local console**.

## 5.2.4 Class: Security Management (FMT)

### FMT\_MOF.1/Functions Management of Security Functions Behavior

FMT\_MOF.1.1/Functions The TSF shall restrict the ability to [modify the behaviour of] the functions [transmission of audit data to an external IT entity] to Security Administrators.

### 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.

### FMT\_MTD.1/CoreData Management of TSF Data

FMT\_MTD.1.1 The TSF shall restrict the ability to manage the TSF data to Security Administrators.

### 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.

### FMT\_MOF.1/Services

FMT\_MOF.1.1/Services The TSF shall restrict the ability to **start and stop** the functions **services** to Security Administrators.

### 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 [digital signature] capability prior to installing those updates;*
  - *Ability to configure the authentication failure parameters for FIA\_AFL.1;*
- [
- *Ability to start and stop services;*
  - *Ability to modify the behaviour of the transmission of audit data to an external IT entity;*
  - *Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA\_UIA\_EXT.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;*
- ].

## **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 Class: Protection of the TSF (FPT)**

**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.

**FPT\_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.

**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*].

**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 [*the most recently installed version of the TOE firmware/software*].

**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 [*digital signature*] prior to installing those updates.

**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: [*digital signature verification of the TOE firmware*]

## **5.2.6 Class: TOE Access (FTA)**

**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.

**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*.

**FTA\_SSL.4 User-initiated Termination**

**FTA\_SSL.4.1** The TSF shall allow **Administrator**-initiated termination of the **Administrator’s** own interactive session.

**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 Class: Trusted Path/Channels (FTP)**

**FTP\_ITC.1 Inter-TSF trusted channel**

**FTP\_ITC.1.1** The TSF shall **be capable of using [SSH]** to provide a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [no other capabilities]** 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 server (Syslog)]*.

**FTP\_TRP.1/Admin Trusted Path**

**FTP\_TRP.1.1/Admin** The TSF shall **be capable of using [SSH]** 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 Collaborative Protection Profile for Network Devices contains all the requirements claimed in this Security Target. 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 Collaborative Protection Profile for Network Devices which are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in the table below.

Assurance Class	Components	Components Description
Security Target(ASE)	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

Assurance Class	Components	Components Description
	ASE_TSS.1	TOE summary specification
Development (ADV)	ADV_FSP.1	Basic functional specification
Guidance documents (AGD)	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Life cycle support (ALC)	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM Coverage
Tests (ATE)	ATE_IND.1	Independent Testing – conformance
Vulnerability Assessment (AVA)	AVA_VAN.1	Vulnerability survey

**Table 14: Security Assurance Requirements**

## 5.5 Rationale for Security Assurance Requirements

The Security Assurance Requirements (SARs) in this Security Target represent the SARs identified in the NDcPPv2.2e. As such, the NDcPPv2.2e SAR rationale is deemed acceptable since the PP itself has been validated.

## 5.6 Assurance Measures

The TOE satisfies the identified assurance requirements. This section identifies the Assurance Measures applied by Apple to satisfy the assurance requirements. The table below lists the details.

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	Klas Telecom Inc. will provide the TOE for testing.

<b>SAR Component</b>	<b>How the SAR will be met</b>
AVA_VAN.1	Klas Telecom Inc. will provide the TOE for testing.

**Table 15: TOE Security Assurance Measures**

## 6 TOE Summary Specification

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

TOE SFR	Rationale
FAU_GEN.1	<p>The TOE generates a comprehensive set of audit logs that identify specific TOE operation whenever an auditable event occurs. Auditable events are specified in Table 13. Each of the events specified in the audit records is in enough detail to identify the user for which the event is associated, when the event occurred, where the event occurred, the outcome of the event and the type of event that occurred. Administrative tasks of generating, deleting cryptographic keys contain the necessary audit information as mandated by FAU_GEN.1.1.</p> <p>The TOE does not have an interface to modify audit records, but may be read by a Security Administrator. The log files can also be deleted by an authorized Security Administrator by issuing a reboot command to delete log files.</p>
FAU_GEN.2	<p>The TOE ensures that each auditable event is associated with the identity of the user that triggered the event.</p>
FAU_STG_EXT.1	<p>The TOE is a standalone device that can be configured to export audit events securely to an external syslog server using SSHv2. The audit logs are transmitted to the external syslog server in real time. The TOE also stores audit records locally in a local audit log file store in volatile memory. The TOE stores log files locally as Audit log and System log. The Audit log file stores the CLI commands entered by the user while the System log stores the general system log messages.</p> <p>The log files can be read only by an authorized Security Administrator but cannot be modified. Each log file is deleted when it reaches a size of 10MB and a new log file is created.</p>
FCS_CKM.1	<p>The TOE supports RSA key sizes of 2048 bits, and 3072 bits for key generation conforming to FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3. The RSA keys are used for public key authentication in SSH.</p> <p>The TOE supports Elliptic NIST Curve sizes of P-256 and P-384 conforming to FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4. The Elliptic keys are used in support of EC SSH session key establishment. EC key pairs are generated for SSH public/private authentication. The TOE supports 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. The TOE supports DHG14 key generation in support of DH key exchanges as part of SSH.</p>
FCS_CKM.2	<p>The TOE supports Cryptographic Key Establishment using the following schemes:</p> <ul style="list-style-type: none"> <li>• RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public Key Cryptography Standards (PKCS) #1; RSA Cryptography Specifications Version 2.1. The TOE implements RSA key establishment scheme with key sizes of 2048 and 3072 bits.</li> <li>• 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";</li> </ul>

TOE SFR	Rationale												
	<ul style="list-style-type: none"> <li>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.</li> </ul> <p>RSA and ECC schemes are used in support of SSH communications and FFC based key exchange based on NIST SP 800-56Ar3/Diffie-Hellman Group 14 (RFC3526).</p> <table border="1" data-bbox="475 526 1297 781"> <thead> <tr> <th>Scheme</th> <th>SFR</th> <th>Service</th> </tr> </thead> <tbody> <tr> <td>RSA</td> <td>FCS_SSHC_EXT.1</td> <td>SSH Session Establishment</td> </tr> <tr> <td>ECC</td> <td>FCS_SSHS_EXT.1</td> <td>Audit Server Administration</td> </tr> <tr> <td>FFC/DHG14</td> <td>FCS_SSHS_EXT.1</td> <td>Syslog Administration</td> </tr> </tbody> </table> <p>Please refer to Table 4 ‘Cryptographic Algorithm Certificates’ for NIST CAVP certificate numbers for RSA and ECDSA.</p>	Scheme	SFR	Service	RSA	FCS_SSHC_EXT.1	SSH Session Establishment	ECC	FCS_SSHS_EXT.1	Audit Server Administration	FFC/DHG14	FCS_SSHS_EXT.1	Syslog Administration
Scheme	SFR	Service											
RSA	FCS_SSHC_EXT.1	SSH Session Establishment											
ECC	FCS_SSHS_EXT.1	Audit Server Administration											
FFC/DHG14	FCS_SSHS_EXT.1	Syslog Administration											
FCS_CKM.4	The TOE satisfies all requirements as specified in FCS_CKM.4 of NDcPP v2.2e for destruction of keys and CSPs. Please refer to Table 17 ‘Key Storage and Zeroization’ in section 6.1 of this document.												
FCS_COP.1/DataEncryption	<p>The TOE supports AES encryption and decryption conforming to CBC as specified in ISO 10116. The AES key size supported is 128 and 256 bits and the AES mode supported is CBC.</p> <p>AES is implemented in the following protocols: SSH.</p> <p>Please refer to Table 4 ‘Cryptographic Algorithm Certificates’ for NIST CAVP certificate numbers for AES.</p>												
FCS_COP.1/SigGen	<p>The TOE provides cryptographic signature generation and verification services in accordance with the following cryptographic algorithms:</p> <ul style="list-style-type: none"> <li>RSA digital signature conforming to 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.</li> <li>The RSA key sizes supported are: 2048, and 3072 bits.</li> <li>The TOE uses Elliptical curve digital signature algorithm conforming to FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing “NIST curves” P-256, P-384; ISO/IEC 14888-3, Section 6.4</li> <li>The Elliptical curve key size supported is 256 bits, and 384 bits.</li> </ul> <p>Please refer to Table 4 ‘Cryptographic Algorithm Certificates’ for NIST CAVPs for RSA and ECDSA.</p>												
FCS_COP.1/Hash	The TOE supports Cryptographic Hashing services conforming to ISO/IEC 10118-3:2004. The hashing algorithms are used for SSH and digital signature generation. The following hashing algorithms are supported: SHA-1, SHA-256, SHA-384, and												

TOE SFR	Rationale																									
	<p>SHA-512. The message digest sizes supported are: 160, 256, 384, and 512 bits. SHS is implemented in the following parts of the TOE:</p> <ul style="list-style-type: none"> <li>• SSH – HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA2-512</li> <li>• Digital signature generation – SHA-256</li> </ul> <p>Please refer to Table 4 ‘Cryptographic Algorithm Certificates’ for NIST CAVPs for SHS.</p>																									
FCS_COP.1/KeyedHash	<p>The TOE supports Keyed-hash message authentication conforming to ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm. HMAC algorithm is used in support of SSH sessions. The characteristics of the HMACs used in the TOE are given in the following table:</p> <table border="1" data-bbox="430 628 1393 956"> <thead> <tr> <th>HMAC Algorithms</th> <th>Hash Functions</th> <th>Block Size</th> <th>Key Lengths</th> <th>Digest Sizes</th> </tr> </thead> <tbody> <tr> <td>HMAC-SHA-1</td> <td>SHA-1</td> <td>512 bits</td> <td>160 bits</td> <td>160 bits</td> </tr> <tr> <td>HMAC-SHA-256</td> <td>SHA-256</td> <td>512 bits</td> <td>256 bits</td> <td>256 bits</td> </tr> <tr> <td>HMAC-SHA-384</td> <td>SHA-384</td> <td>1024 bits</td> <td>384 bits</td> <td>384 bits</td> </tr> <tr> <td>HMAC-SHA-512</td> <td>SHA-512</td> <td>1024 bits</td> <td>512 bits</td> <td>512 bits</td> </tr> </tbody> </table> <p>Please refer to Table 4 ‘Cryptographic Algorithm Certificates’ for NIST CAVPs for HMAC.</p>	HMAC Algorithms	Hash Functions	Block Size	Key Lengths	Digest Sizes	HMAC-SHA-1	SHA-1	512 bits	160 bits	160 bits	HMAC-SHA-256	SHA-256	512 bits	256 bits	256 bits	HMAC-SHA-384	SHA-384	1024 bits	384 bits	384 bits	HMAC-SHA-512	SHA-512	1024 bits	512 bits	512 bits
HMAC Algorithms	Hash Functions	Block Size	Key Lengths	Digest Sizes																						
HMAC-SHA-1	SHA-1	512 bits	160 bits	160 bits																						
HMAC-SHA-256	SHA-256	512 bits	256 bits	256 bits																						
HMAC-SHA-384	SHA-384	1024 bits	384 bits	384 bits																						
HMAC-SHA-512	SHA-512	1024 bits	512 bits	512 bits																						
FCS_RBG_EXT.1	<p>The TOE implements a NIST-approved AES-CTR Deterministic Random Bit Generator (DRBG) that complies with ISO 18031:2011. The entropy source used to seed the Deterministic Random Bit Generator is a random set of bits regularly supplied to the DRBG from five separate software sources to generate entropy. The DRBG is seeded with at least 256 bits of entropy software-based noise source.</p> <p>The relevant NIST CAVP certificate numbers are listed in Table 4.</p>																									
FIA_AFL.1	<p>An administrator can configure the maximum number of failed attempts using the CLI interface. The configurable range is between 1 and 255 attempts. When a user account has sequentially failed authentication for the configured number of times, the account will be locked, until a local administrator manually unlocks the account. If the lockout attempts are set to, for example, 5 attempts, then the user will be locked out after the 5<sup>th</sup> consecutive failed login attempt. This means that the 6<sup>th</sup> and subsequent attempts will fail to gain access to the TOE even if the credential being offered is correct. All failed attempts and lockouts are tracked by the TOE audit logs.</p> <p>The TOE will always allow a user to authenticate using the local console port, even if the user account is locked. This behavior is not configurable.</p>																									
FIA_PMG_EXT.1	<p>The TOE supports the local definition of users with corresponding passwords. The passwords can be composed of any combination of upper- and lower-case letters, numbers, and special characters that include these characters include the following:</p> <p>“!”, “@”, “#”, “\$”, “%”, “^”, “&amp;”, “*”, “(”, “)”, “~”, “&lt;”, “&gt;”, “,”, “:”, “/”, “.”, “;”, “_”, “+”, “-”, “=”, “{”, “}”, “[”, “]”, “ ”</p> <p>The minimum password length can be configured by the Administrator and can range from 15 to 128 characters.</p>																									



TOE SFR	Rationale
FIA_UIA_EXT.1	<p>The TOE requires all users to be successfully identified and authenticated as an administrator before allowing any TSF mediated actions to be performed. Access to the TOE is facilitated through one of several interfaces,</p> <ul style="list-style-type: none"> <li>• Directly connecting to the TOE through serial console</li> <li>• Remotely connecting to the TOE through SSHv2</li> </ul> <p>Every user that authenticates is first logged in with non-administrative privileges with limited viewing functionalities. The user may then authenticate as an administrator with additional credentials to gain access to modifying functionalities. Regardless of the interface at which the administrator interacts, the TOE prompts the user for a credential. Only after the administrative user presents the correct authentication credentials will they be granted access to the TOE administrative functionality. No TOE administrative access is permitted until an administrator is successfully identified and authenticated.</p> <p>For remote administration, the TOE supports public key authentication and password-based authentication. If the user uses public key-based authentication and it is successful, then the user is granted access to the TOE. If the user uses password-based authentication and they provide valid username and password, then user is granted access to the TOE. If the user enters invalid user credentials, they will not be granted access. The TOE does not provide a reason for failure in case of a login failure.</p>
FIA_UAU_EXT.2	The TOE provides a local password-based authentication mechanism to perform user authentication for local administration through serial console.
FIA_UAU.7	For all authentication at the local CLI the TOE does not display any authentication data. The TOE does not display even obscured data such as asterisks during authentication attempts.
FCS_SSHC_EXT.1.1	The TOE implements SSH protocol that complies with RFC(s) 4251, 4252, 4253, 4254, 5656, and 6668.
FCS_SSHC_EXT.1.2	The TOE supports public key authentication. The following are the public key algorithms supported: ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384. This list conforms to FCS_SSHC_EXT.1.5.
FCS_SSHC_EXT.1.3	The TOE accepts packet size up to 33,292 bytes and meets the requirements of RFC 4253. Packets exceeding this size are dropped and logged by the TOE.
FCS_SSHC_EXT.1.4	The TOE supports the following encryption algorithms: AES-128-CBC and AES-256-CBC for SSH to ensure confidentiality of the session. There are no optional characteristics specified.
FCS_SSHC_EXT.1.5	When the TOE acts as an SSH client, the TOE authenticates the identity of the remote SSH server using the corresponding public key. The TOE supports the use of ssh-rsa, ecdsa-sha2-nistp256, and ecdsa-sha2-nistp384 public keys for server authentication. There are no optional characteristics specified.
FCS_SSHC_EXT.1.6	The TOE supports the following data integrity algorithms: hmac-sha1, hmac-sha2-256, and hmac-sha2-512 for SSH to ensure integrity of the session. This list conforms to FCS_SSHC_EXT.1.6
FCS_SSHC_EXT.1.7	The TOE supports the following key exchange algorithms: diffie-hellman-group14-sha1, ecdh-sha2-nistp256 and ecdh-sha2-nistp384. This list conforms to FCS_SSHC_EXT.1.7
FCS_SSHC_EXT.1.8	The TOE is capable of rekeying. The TOE verifies the following thresholds:

TOE SFR	Rationale
	<ul style="list-style-type: none"> <li>• No longer than one hour</li> <li>• No more than 950 MB of transmitted data</li> </ul> <p>The TOE continuously checks both conditions. When either of the conditions are met, the TOE will initiate a rekey.</p>
FCS_SSHC_EXT.1.9	The TOE authenticates the identity of the SSH server using a local database which associates each host name with its corresponding public key.
FCS_SSHS_EXT.1.1	The TOE implements SSH protocol that complies with RFC(s) 4251, 4252, 4253, 4254, 5656, and 6668.
FCS_SSHS_EXT.1.2	<p>The TOE supports SSH password-based authentication and public key authentication.</p> <p>The following key pairs are supported: ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384. This list conforms to FCS_SSHC_EXT.1.5.</p>
FCS_SSHS_EXT.1.3	The TOE accepts packet size up to 33,292 bytes and meets the requirements of RFC 4253. Packets exceeding this size are dropped and logged by the TOE.
FCS_SSHS_EXT.1.4	The TOE supports the following encryption algorithms: AES-128-CBC and AES-256-CBC for SSH to ensure confidentiality of the session. There are no optional characteristics specified.
FCS_SSHS_EXT.1.5	The following are the public key algorithms supported: ssh-rsa, ecdsa-sha2-nistp256 and ecdsa-sha2-nistp384. There are no optional characteristics specified for FCS_SSHS_EXT.1.5. This list conforms to FCS_SSHS_EXT.1.5. The TOE identifies the public key that is presented by the client and verifies if it matches one of the stored keys within the server. If the presented key does not match, authentication is prevented.
FCS_SSHS_EXT.1.6	The TOE supports the following data integrity algorithms: hmac-sha1, hmac-sha2-256, and hmac-sha2-512 for SSH to ensure integrity of the session. This list conforms to FCS_SSHC_EXT.1.6
FCS_SSHS_EXT.1.7	The TOE supports the following key exchange algorithms: diffie-hellman-group14-sha1, ecdh-sha2-nistp256 and ecdh-sha2-nistp384. This list conforms to FCS_SSHS_EXT.1.7
FCS_SSHS_EXT.1.8	<p>The TOE is capable of rekeying. The TOE verifies the following thresholds:</p> <ul style="list-style-type: none"> <li>• No longer than one hour</li> <li>• No more than 950 MB of transmitted data</li> </ul> <p>The TOE continuously checks both conditions. When either of the conditions are met, the TOE will initiate a rekey.</p>
FMT_MOF.1/ManualUpdate	The TOE restricts the ability to perform software updates to Security Administrators.
FMT_MTD.1/CoreData FMT_MTD.1/Cryptkeys	<p>The TOE restricts the ability to manage the TOE to Security Administrators. Administrative users are required to login before being provided with access to any administrative functions. Non-security administrators are not allowed to modify any TOE functions. No interface is available to an unauthenticated user except the login prompt. Any commands used to modify, and TOE functions is not made available to non-administrative users and its attempt to use them will result in an invalid action error.</p> <p>The security administrator can generate, import, and delete cryptographic keys through the TOE's Global Configuration mode.</p>

TOE SFR	Rationale
FMT_SMF.1 FMT_MOF.1/Services	<p>The TOE may be managed via the CLI (console and remote SSH). The specific management capabilities include:</p> <ul style="list-style-type: none"> <li>• Ability to administer the TOE locally and remotely;</li> <li>• Ability to configure the access banner;</li> <li>• Ability to configure the session inactivity time before session termination or locking;</li> <li>• Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;</li> <li>• Ability to configure the authentication failure parameters for FIA_AFL.1;</li> <li>• Ability to start and stop services;</li> <li>• Ability to modify the behaviour of the transmission of audit data to an external IT entity;</li> <li>• Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1;</li> <li>• Ability to manage the cryptographic keys;</li> <li>• Ability to configure the cryptographic functionality;</li> <li>• Ability to re-enable an Administrator account;</li> <li>• Ability to set the time which is used for time-stamps;</li> </ul> <p>The Security Administrator can start the SSH tunnel and stop the SSH tunnel.</p> <p>Local console and remote administration provide the same functionalities based on the level of authentication.</p>
FMT_MOF.1/Functions	<p>The Security administrator is able to modify the behaviour of transmission of audit data to the syslog server.</p> <p>The log files can be read only by an authorized Security Administrator but cannot be modified. Each log file is deleted when it reaches a size of 10MB and a new log file is created.</p>
FMT_SMR.2	<p>The TOE maintains the role of a Security Administrator. The Security Administrator is capable of managing the device. Users who have not authenticated with administrative credentials have the capabilities of viewing certain parameters but are incapable of making any changes.</p>
FPT_SKP_EXT.1	<p>The TOE stores all private keys in a secure storage and is not accessible through an interface to administrators. Passwords are obscured from the user from local and remote CLI interfaces. Private keys may be destroyed or replaced but cannot be read. Refer to Section 6.1 'Key Storage and Zeroization' Table 17 for key storage details.</p>
FPT_APW_EXT.1	<p>The TOE stores all password authentication data in a secure directory that is not readily accessible to administrators. Passwords are obscured from the user from both local and remote CLI interfaces. The passwords are stored as SHA-512 hash and are not in plaintext.</p>
FPT_TST_EXT.1	<p>The TOE runs a suite of self-tests during initial start-up to verify its correct operation. If any of the tests fail, the TOE will enter an error state. The TOE executes the following self-tests when powered on:</p> <ul style="list-style-type: none"> <li>• Integrity check – The TOE performs an integrity check of the installed firmware by comparing the 4096-bit digital signature of the complete firmware image during bootup before any configuration is loaded and interfaces are enabled.</li> <li>• Cryptographic known-answer Tests – The TOE performs the KATs on all</li> </ul>

TOE SFR	Rationale
	<p>cryptographic modules to verify that they are working as expected. If any cryptographic self-test fails, the TOE immediately reboots and logs an audit message at the console.</p> <ul style="list-style-type: none"> <li>Entropy health testing – If the entropy noise source health testing fails, the TOE immediately reboots and logs an audit message at the local console.</li> </ul>
FPT_TUD_EXT.1	<p>A Security Administrator can query the software version running on the TOE and is able to perform manual software updates to the TOE. When software updates are made available by Klas, the Security Administrator can download, verify the integrity of, and install the updates.</p> <p>The TOE image files are digitally signed using RSA digital signature mechanism, so their integrity can be verified during the update process. An image that fails an integrity check will not be loaded and is automatically deleted by the TOE.</p>
FPT_STM_EXT.1	<p>The TOE provides reliable time stamps. The clock function is reliant on the system clock provided by the underlying hardware. The following security functions make use of the system time:</p> <ul style="list-style-type: none"> <li>Audit events</li> <li>Session inactivity</li> <li>SSH Rekey</li> </ul> <p>The time can be manually updated by a Security Administrator.</p>
FTA_SSL_EXT.1 FTA_SSL.3	<p>A Security Administrator can configure maximum inactivity times for administrative sessions through the TOE, local CLI, and remote SSH interfaces. The configuration of inactivity periods are applied on a per-interface basis and can be applied to both, local, and remote sessions in the same manner. When the interface has been idle for more than the configured period of time, the session will be terminated and will require reauthentication to establish a new session.</p>
FTA_SSL.4	<p>A Security Administrator is able to exit out of both, local, and remote administrative sessions.</p>
FTA_TAB.1	<p>Security Administrators can define a customized login banner that will be displayed at the following interfaces:</p> <ul style="list-style-type: none"> <li>Local CLI</li> <li>Remote CLI</li> </ul> <p>This banner will be displayed prior to allowing Security Administrators access through these interfaces.</p>
FTP_ITC.1	<p>The TOE supports communications with remote Audit servers using SSH v2 protocol. Each of these connections is protected by a secure SSH session, where the TOE acts as a client. This protects the data from disclosure by encryption using AES. SSH provides assured identification of the non-TSF endpoint by validating the remote SSH server's public key. The TOE is responsible for initiating the trusted channel with the external trusted IT entities.</p>
FTP_TRP.1/Admin	<p>All remote administrative communications take place over a secure encrypted SSHv2 session. The SSHv2 session is encrypted using AES encryption to protect confidentiality and uses HMAC to protect integrity of traffic. The remote administrators are able to initiate SSHv2 sessions with the TOE.</p>

**Table 16: TOE Summary Specification SFR Description**

## 6.1 Key Storage and Zeroization

The following table describes the origin, storage and zeroization of keys as relevant to FCS\_CKM.4 and FPT\_SKP\_EXT.1 provided by the TOE.

Key Type	Usage	Storage	Destruction
EC Session Keys	Ephemeral Session Key for SSH session establishment	Ephemeral; stored in RAM	Overwritten with zeroes at end of session.
DH G14 Session Keys	Ephemeral Session Key for SSH session establishment.	Ephemeral; stored in RAM	Overwritten with zeroes at end of session.
RSA Key	Signature Generation, Signature Verification for SSH public key authentication.	Restricted key partition in plaintext	Deleted with read-verify when any of the designated cryptographic key zeroization commands identified in AGD are executed by the administrator. Cryptographic key zeroize Cryptographic key zeroize rsa  Key zeroization will instruct a part of the TOE to destroy the abstraction that represents the key. Generating a new key will overwrite and erase any existing keys and replacing the old keys with a new key value.
		While in use, RSA keys are held in RAM	Overwritten with zeroes when the key is no longer in use (after performing a cryptographic operation) or overwritten with a new value of the key when a new key
ECDSA Key	Signature Generation. Signature Verification for SSH public key authentication and verification of trusted updates.	Restricted key partition in plaintext	Deleted with read-verify when any of the designated cryptographic key zeroization commands identified in AGD are executed by the administrator. Cryptographic key zeroize Cryptographic key zeroize etc  Key zeroization will instruct a part of the TOE to destroy the abstraction that represents the key. Generating a new key will overwrite and erase any existing keys and replacing the old keys with a new key value.
		While in use, ECDSA keys are held in RAM	Overwritten with zeroes when the key is no longer in use (after performing a cryptographic operation)
HMAC Key	Keyed Hashing for SSH	While in use, keys for HMAC keyed hashing are held in RAM	Overwritten with zeroes when the key is no longer in use (after performing a cryptographic operation)
AES Session Keys	SSH Data Encryption	Ephemeral; stored in RAM	Overwritten with zeroes at end of session

Key Type	Usage	Storage	Destruction

**Table 17: Key Storage and Zeroization**

## 7 Terms and Definitions

Abbreviations/ Acronyms	Description
AES	Advanced Encryption Standard
CA	Certificate Authority
CBC	Cipher Block Chaining
CLI	Command Line Interface
CRL	Certificate Revocation List
CSR	Certificate Signing Request
DH	Diffie-Hellman
DHE	Diffie-Hellman Ephemeral
DNS	Domain Name System
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
ECDH	Elliptic Curve Diffie-Hellman
ECDHE	Elliptic Curve Diffie-Hellman Ephemeral
ECDSA	Elliptic Curve Digital Signature Algorithm
GCM	Galois Counter Mode
GUI	Graphical User Interface
HMAC	Hash Message Authentication Code
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
OCSP	Online Certificate Status Protocol
RFC	Requests for Comments
RSA	Rivest-Shamir-Adleman
SA	Security Association
SAN	Subject Alternative Name
SHA	Secure Hash Algorithm
SPD	Security Policy Database
SSH	Secure Shell
TLS	Transport Layer Security
UI	User Interface

**Table 18: TOE Abbreviations and Acronyms**

Abbreviations/ Acronyms	Description
CC	Common Criteria
FIPS	Federal Information Processing Standards
OSP	Organizational Security Policy
PP	Protection Profile
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface
TSS	TOE Summary Specification

**Table 19: CC Abbreviations and Acronyms**

## 8 References

### Common Criteria References

Reference	Description	Version	Date
[C1]	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model CCMB-2017-04-001	V3.1 R5	April 2017
[C2]	Common Criteria for Information Technology Security Evaluation Part 2: Security functional components CCMB-2017-04-002	V3.1 R5	April 2017
[C3]	Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components CCMB-2017-04-003	V3.1 R5	April 2017
[CEM]	Common Methodolgy for Information Technology Security Evaluation Evaluation Methodology CCMB-2017-04-004	V3.1 R5	April 2017

**Table 20: Common Criteria v3.1 References**



### Supporting Documentation

Reference	Description	Version	Date
[cPP]	Collaborative Protection Profile for Network Devices + Errata 20200323	2.2E	March 23, 2020
[SD]	Supporting Document Mandatory Technical Document Evaluation Activities for Network Device cPP	2.2	December 20, 2019

**Table 21: Supporting Documentation**