



## Security Target Junos OS 22.3R1 for MX204, ACX5448 and ACX5448-D

---

Juniper Networks, Inc.

Version 1.1

10 September, 2024

Prepared for:  
Juniper Networks, Inc.  
1133 Innovation Way  
Sunnyvale, CA 94089  
[www.juniper.net](http://www.juniper.net)

## *Abstract*

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), Junos OS 22.3R1 for MX204, ACX5448 and ACX5448-D. This Security Target (ST) is conformant to the Collaborative Protection Profile for Network Devices v2.2E [NDcPP2.2E].

## *References*

- [CC1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1 Revision 5, April 2017
- [CC2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1 Revision 5, April 2017.
- [CC3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components, CCMB-2017-04-003, Version 3.1 Revision 5, April 2017
- [CEM] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, CCMB-2017-04-004, Version 3.1, Revision 5, April 2017.
- [CC\_Add] CC and CEM addenda; Exact Conformance, Selection-Based SFRs, Optional SFRs, Version: 2.0 Final, Date of issue: 2021-Sep-30, CCDB-013-v2.0
- [NDcPP2.2E] Collaborative Protection Profile for Network Devices (NDcPP), Version 2.2E, 23-March-2018
- [SD] Supporting Document, Evaluation Activities for Network Device cPP, December-2019, version 2.2, CCDB-2019-12-004.

## Table of Contents

1	Introduction .....	5
1.1	ST reference .....	5
1.2	TOE Reference.....	5
1.3	About this document .....	5
1.4	Document Conventions .....	5
1.5	TOE Overview.....	6
1.6	TOE Description.....	6
1.6.1	Overview .....	6
1.6.2	Physical boundary .....	7
1.6.3	Logical Scope of the TOE.....	8
1.6.4	Non-TOE hardware/software/firmware .....	9
1.6.5	Summary of out scope items .....	9
2	Conformance Claim.....	10
2.1	CC Conformance Claim .....	10
2.2	PP Conformance claim .....	10
2.3	Conformance Rationale .....	10
2.4	Technical Decisions .....	10
3	Security Problem Definition .....	12
3.1	Threats .....	12
3.2	Assumptions.....	13
3.3	Organizational Security Policies.....	14
4	Security Objectives.....	15
4.1	Security Objectives for the TOE .....	15
4.2	Security Objectives for the Operational Environment.....	15
4.3	Security Objectives rationale .....	15
5	Security Functional Requirements.....	16
5.1	Security Audit (FAU).....	16
5.1.1	Security Audit Data generation (FAU_GEN).....	16
5.1.2	Security audit event storage (Extended – FAU_STG_EXT).....	18
5.2	Cryptographic Support (FCS).....	18
5.2.1	Cryptographic Key Management (FCS_CKM).....	18
5.2.2	Cryptographic Operation (FCS_COP) .....	19
5.2.3	FCS_RBG_EXT.1 Random Bit Generation .....	20
5.2.4	Cryptographic Protocols (Extended – FCS_SSHS_EXT SSH Protocol).....	20
5.3	Identification and Authentication (FIA) .....	21
5.3.1	Authentication Failure Management (FIA_AFL) .....	21

5.3.2	Password Management (Extended – FIA_PMG_EXT).....	21
5.3.3	User Identification and Authentication (Extended – FIA_UIA_EXT) .....	21
5.3.4	User authentication (FIA_UAU) (Extended – FIA_UAU_EXT).....	22
5.4	Security Management (FMT) .....	22
5.4.1	Management of functions in TSF (FMT_MOF).....	22
5.4.2	Management of TSF Data (FMT_MTD) .....	22
5.4.3	Specification of Management Functions (FMT_SMF).....	23
5.4.4	Security management roles (FMT_SMR) .....	23
5.5	Protection of the TSF (FPT) .....	23
5.5.1	Protection of TSF Data (Extended – FPT_SKP_EXT) .....	23
5.5.2	Protection of Administrator Passwords (Extended – FPT_APW_EXT).....	24
5.5.3	TSF testing (Extended – FPT_TST_EXT) .....	24
5.5.4	Trusted Update (FPT_TUD_EXT) .....	24
5.5.5	Time stamps (Extended – FPT_STM_EXT)) .....	24
5.6	TOE Access (FTA).....	24
5.6.1	TSF-initiated Session Locking (Extended – FTA_SSL_EXT) .....	24
5.6.2	Session locking and termination (FTA_SSL) .....	25
5.6.3	TOE access banners (FTA_TAB).....	25
5.7	Trusted path/channels (FTP).....	25
5.7.1	Trusted Channel (FTP_ITC).....	25
5.7.2	Trusted Path (FTP_TRP).....	25
6	Security Assurance Requirements .....	26
7	TOE Summary Specification .....	27
7.1	Protected communications.....	27
7.1.1	Algorithms and zeroization .....	27
7.1.2	Random Bit Generation .....	33
7.1.3	SSH .....	33
7.2	Administrator Authentication.....	37
7.3	Correct Operation .....	39
7.4	Trusted Update .....	40
7.5	Audit.....	40
7.6	Management.....	41
8	Glossary.....	43

## 1 Introduction

1. This section identifies the Security Target (ST), Target of Evaluation (TOE), Security Target organization, document conventions, and terminology. It also includes an overview of the evaluated products.

### 1.1 ST reference

<b>ST Title</b>	Security Target Junos OS 22.3R1 for MX204, ACX5448 and ACX5448-D
<b>ST Revision</b>	1.1
<b>ST Draft Date</b>	10 September, 2024
<b>Author</b>	Juniper Networks, Inc.
<b>cPP/EP Conformance</b>	[NDcPP2.2E] Collaborative Protection Profile for Network Devices (NDcPP), Version 2.2E, 23-March-2020

### 1.2 TOE Reference

<b>TOE Title</b>	Junos OS 22.3R1 for MX204, ACX5448 and ACX5448-D
<b>TOE Software</b>	Junos OS 22.3R1

### 1.3 About this document

2. This Security Target follows the following format:

Section	Title	Description
1	Introduction	Provides an overview of the TOE and defines the hardware and software that make up the TOE as well as the physical and logical boundaries of the TOE
2	Conformance Claims	Lists evaluation conformance to Common Criteria versions, Protection Profiles, or Packages where applicable
3	Security Problem Definition	Specifies the threats, assumptions and organizational security policies that affect the TOE
4	Security Objectives	Defines the security objectives for the TOE/operational environment and provides a rationale to demonstrate that the security objectives satisfy the threats
5	Security Functional Requirements	Contains the functional requirements for this TOE
6	Security Assurance Requirements	Contains the assurance requirements for this TOE
7	TOE Summary Specification	Identifies the IT security functions provided by the TOE and also identifies the assurance measures targeted to meet the assurance requirements
8	Acronyms	The acronyms used in the ST are explained

Table 1 Document Organization

### 1.4 Document Conventions

3. This document follows the same conventions as those applied in [NDcPP2.2E] in the completion of operations on Security Functional Requirements, namely:
  - Unaltered SFRs are stated in the form used in [CC2] or their extended component definition (ECD);
  - Refinement made in the ST: the refinement text is indicated with **bold text** and ~~strikethroughs~~;
  - Selection completed in the ST: the selection values are indicated with underlined text

e.g. “[*selection: disclosure, modification, loss of use*]” in [CC2] or an ECD might become “disclosure” (completion);

- Assignment completed in the ST: indicated with *italicized text*;
- Assignment completed within a selection in the ST: the completed assignment text is indicated with *italicized and underlined text*

e.g. “[*selection: change\_default, query, modify, delete, [assignment: other operations]*]” in [CC2] or an ECD might become “change\_default, select\_tag” (completion of both selection and assignment);

- Iteration: indicated by adding a string starting with “/” (e.g. “FCS\_COP.1/Hash”).

## 1.5 TOE Overview

4. The Target of Evaluation (TOE) are the Juniper Networks, Inc. MX204, ACX5448 and ACX5448-D 5G Universal Routing Platforms executing the Junos OS 22.3R1 software. The TOE is a complete virtual appliance consisting of all hardware and software.
5. The variants of the TOE are secure network devices which protect themselves largely by offering only a minimal logical interface to the network and the attached nodes. Junos OS 22.3R1 is a special purpose OS that does not provide any general-purpose computing capabilities. Junos OS implements both management and control functions as well as all IP routing.
6. The TOE supports the definition and enforcement of information flow policies among subnetworks. Each information flow from one subnetwork to another passes through an instance of the TOE and the TOE decides, based on defined policies, whether the traffic is forwarded or dropped. Information flows are controlled based on network addresses and protocols. The TOE also ensures that security-relevant activity is audited and provides the necessary functions to manage all the security functions.

## 1.6 TOE Description

### 1.6.1 Overview

7. The Juniper Networks MX204, ACX5448 and ACX5448-D 5G Universal Routing Platforms are Ethernet-optimized edge routers with 400-Gbps capacity. They provide both switching and carrier-class Ethernet routing, and deliver an end-to-end infrastructure security solution for enterprises looking to move business-critical applications to public clouds. They are complete routing systems which deliver features, functionality, and secure services at scale in the 5G era. They share common Junos software, features, and technology for compatibility across platforms.
8. The variants of the TOE are appliances which are physically self-contained, housing the software and hardware necessary to perform all routing functions. The architecture components of the TOE are:
  - Routing Engine (Control Board) – the Routing Engine (RE) runs the Junos OS 22.3R1 software and implements Layer 3 routing services and Layer 2 switching services. The RE also implements a network management interface for the configuration and operation of the TOE. The RE controls the flow of information through the TOE, including support for appliance interface control and control plane functions such as chassis component, system management and user access to the appliance.
  - The Packet Forwarding Engine (PFE) – implements all operations necessary for transit packet forwarding. The TOE support an extensive set of Layer 2 and Layer 3 services that can be deployed in any combination of L2- L3 applications.

- Power – power supply bays allow flexibility for provisioning and redundancy. The power supplies distribute the different output voltages produced by the power supplies to the TOE components depending on their voltage requirements.
- The RE and the PFE are implemented on top of a virtualization software. Virtualization hides the hardware details from the RE and PFE. Re and PFE perform their primary tasks independently, while constantly communicating through a high-speed internal link. This enables streamlined forwarding and routing control and the capability to run Internet-scale networks at high speeds.
  - The functions of the TOE can be managed using a Command Line Interface (CLI) implemented by the Junos OS. The CLI may be accessed from a connected terminal console or via a network connection. Management over a network connection is secured using the SSH protocol. All management accesses require successful authentication. The TOE implements measures to prevent access by the parties not successfully authenticated and to make it hard for unauthorized parties to gain access to the CLI e.g. by guessing the credentials of an authorized user.

### 1.6.2 Physical boundary

- The TOE is the complete appliance consisting of the Junos OS 22.3R1 software running on the MX204, ACX5448 or ACX5448-D chassis. The TOE is fully contained within the physical boundary of the appliance chassis as illustrated in Figure 1. The physical boundary of the TOE is the entire chassis of the appliance.

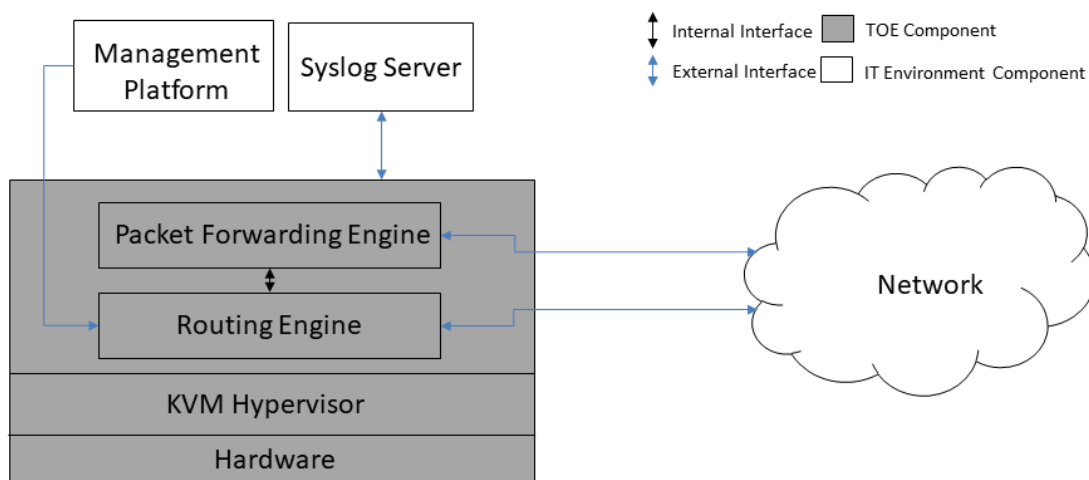


Figure 1 TOE Boundary

- The interfaces the TOE implements are the network interfaces which controls the traffic between the connected subnetworks and the management interface for administering the TOE.
- The details of the appliances, including the supported interface options are described in the following CC Guidance documents. The guidance documents are part of the TOE.

[ECG 1] Junos® OS Common Criteria Evaluated Configuration Guide for MX204 Devices, Published 2023-01-11, RELEASE22.3R1

[ECG 2] Junos® OS Common Criteria Configuration Guide for ACX5448 Devices, Published 2024-06-20, RELEASE 22.3R1

- The software image of the TOE is `junos-vmhost-install-mx-x86-64-22.3R1.8.tgz` for the MX204 model and `junos-vmhost-install-x86-64-22.3R1.8.tgz` for the ACX5448 models.

15. The software version can be viewed by an administrator by the `show version` command executed on the CLI of the TOE.

### 1.6.3 Logical Scope of the TOE

16. The logical boundary of the TOE includes the following security functionality:

Security Functionality	Description
<b>Security Audit</b>	The TOE generates an audit record for each auditable event. The audit records are stored in syslog files on the TOE and can be sent to an external audit server via Netconf over SSH. Auditable events include start-up and shutdown of the audit functions, authentication events, and all events listed in Table 8. Audit records include the date and time of the event, event category, event type, username, and the outcome of the event (success or failure). Local syslog storage limits are configurable and are monitored. If the storage limit is reached the oldest logs will be overwritten.
<b>Cryptographic Support</b>	The TOE implements an SSH server for administrators to establish secure sessions between the network management station and the TOE, and to connect to external audit servers. Each remote host must be successfully authenticated prior to the TOE allowing any communication with it. The TOE includes cryptographic modules that implement the underlying cryptographic services, including key management and protection of stored keys, cryptographic algorithms, random bit generation and administration of the cryptographic functions. SSH implemented with the cryptographic modules enforce authenticity, confidentiality and integrity of all communication and administrative accesses to the TOE.
<b>Identification and Authentication</b>	The TOE implements Role Based Access Control. All users must be successfully authenticated and assigned to the role Security Administrator by the TOE prior to the TOE granting them access to the CLI for the management of the TOE. Human users are authenticated with a password while the remote hosts are authenticated with public key cryptography. Based on the assigned role, a user is granted a set of privileges to access the system. Authentication data entered and stored on the TOE is protected. The TOE can be configured to terminate interactive user sessions and to present an access banner with warning messages prior to authentication.
<b>Security Management</b>	The TOE implements a Security Administrator role. Users successfully authenticated and assigned to the role are granted the right to: <ul style="list-style-type: none"> <li>• configuration and maintenance of cryptographic functions used for the establishment of secure connections to and from the TOE;</li> <li>• review all audit data;</li> <li>• initiation of trusted updates; and</li> <li>• all other administrative tasks.</li> </ul> The TOE is managed through a Command Line Interface (CLI) which is accessible through local (serial) console or remotely over SSH.
<b>Protection of the TSF</b>	The TOE protects all passwords, pre-shared keys, symmetric keys and private keys from unauthorized disclosure. Passwords are stored using SHA-1, SHA-256 or SHA-512. The TOE executes a suite of self-tests during the initial start-up to ensure the correct operation of critical security functions. All software updates may be verified for authenticity. The TOE also implements an



	internal clock to maintain the date and time and to issue time stamps for other security functions.
<b>TOE Access</b>	The TOE displays an access banner in each user authentication exchange. The banner messaging is customizable to allow the Security Administrator to inform the users of the conditions of access and sanctions of attempted unauthorized access. The TOE maintains an inactivity timer for each session and will terminate each interactive session after a period of inactivity. The CLI implements a command <code>exit</code> which allows each user to terminate their session.
<b>Trusted Path/Trusted Channel</b>	The TOE implements SSH v2 for secure communication between itself and a remote Syslog server and between itself and a network management station.

Table 2 Logical Scope of TOE

#### 1.6.4 Non-TOE hardware/software/firmware

17. The TOE requires the following elements in the network environment:

- Syslog server supporting SSHv2 connections for sending audit logs;
- Management workstation for local or remote administration,
- SSHv2 client for remote administration; and
- Serial connection client for local administration.

#### 1.6.5 Summary of out scope items

- Use of telnet, since it violates the Trusted Path requirement set (see Section 5.7.2)
- Use of FTP, since it violates the Trusted Path requirement set (see Section 5.7.2)
- Use of SNMP, since it violates the Trusted Path requirement set (see Section 5.7.2)
- Use of SSL, including management via J-Web, JUNOScript and JUNOScope, since it violates the Trusted Path requirement set (see Section 5.7.2)
- Use of CLI account super-user and linux root account.

## 2 Conformance Claim

### 2.1 CC Conformance Claim

18. 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 Evaluation Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluation Part 2, Version 3.1, Revision 5, April 2017: Part 2 extended
- Common Criteria for Information Technology Security Evaluation Part 3, Version 3.1, Revision 5, April 2017: Part 3 conformant

### 2.2 PP Conformance claim

19. This TOE is PP-Conformant to and claims exact conformance to:

- [NDcPP2.2E] Collaborative Protection Profile for Network Devices (NDcPP), Version 2.2,E 23-March-2020

### 2.3 Conformance Rationale

20. This Security Target claims 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. Only operations allowed in the Protection Profile are performed on the Security Requirements.

### 2.4 Technical Decisions

21. The NIAP Technical Decisions (TD) applicable to the TOE are listed in Table 3.

Technical Decisions	Applicable	Exclusion Rationale (if applicable)
0800 – Updated NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	No	The TOE does not implement IPsec.
0792 – NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR	Yes	
0790 – NIT Technical Decision: Clarification Required for testing IPv6	No	The TOE does not claim TLS, DTLS or IPv6.
0738 – NIT Technical Decision for Link to Allowed-With List	Yes	
0670 – NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	No	The TOE does not claim TLS.
0639 – NIT Technical Decision for Clarification for NTP MAC Keys	No	The TOE does not claim NTP.
0638 – NIT Technical Decision for Key Pair Generation for Authentication	Yes	
0636 – NIT Technical Decision for Clarification of Public Key User Authentication for SSH	No	The TOE does not claim SSH Client.
0635 – NIT Technical Decision for TLS Server and Key Agreement Parameters	No	The TOE does not claim TLS.

0632 – NIT Technical Decision for Consistency with Time Data for vNDs	Yes <sup>1</sup>	
0631 – NIT Technical Decision for Clarification of public key authentication for SSH Server	Yes	
0592 – NIT Technical Decision for Local Storage of Audit Records	Yes	
0591 – NIT Technical Decision for Virtual TOEs and hypervisors	Yes	
0581 – NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	Yes	
0580 – NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	Yes	
0572 – NIT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	Yes	
0571 – NIT Technical Decision for Guidance on how to handle FIA_AFL.1	Yes	
0570 – NIT Technical Decision for Clarification about FIA_AFL.1	Yes	
0569 – NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	No	The TOE does not claim DTLS.
0564 – NIT Technical Decision for Vulnerability Analysis Search Criteria	Yes	
0563 – NIT Technical Decision for Clarification of audit date information	Yes	
0556 – NIT Technical Decision for RFC 5077 question	No	The TOE does not claim TLS.
0555 – NIT Technical Decision for RFC Reference incorrect in TLSS Test	No	The TOE does not claim TLS.
0547 – NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	Yes	
0546 – NIT Technical Decision for DTLS - clarification of Application Note 63	No	The TOE does not claim DTLS.
0537 – NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	No	The TOE does not claim TLS.
0536 – NIT Technical Decision for Update Verification Inconsistency	Yes	
0528 – NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	No	The TOE does not claim NTP.
0527 – Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	No	The TOE does not claim X.509 certificate-based authentication.

Table 3 Technical Decisions

<sup>1</sup> The TOE is not a virtual Network Device (vND) but the TD requires modification of FPT\_STM\_EXT.1.2 which the TOE claims. The selection implemented in this ST does not claim the vND specific items.

### 3 Security Problem Definition

22. The security problem definition has been taken from [NDcPP2.2E] and is reproduced here.

#### 3.1 Threats

23. Threats applicable to the TOE are given in Table 4.

Threat	Threat Definition
<b>T.UNAUTHORIZED_ADMINISTRATOR_ACCESS</b>	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.
<b>T.WEAK_CRYPTOGRAPHY</b>	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.
<b>T.UNTRUSTED_COMMUNICATION_CHANNELS</b>	Threat agents may attempt to target Network Devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the Network Device itself.
<b>T.WEAK_AUTHENTICATION_ENDPOINTS</b>	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.
<b>T.UPDATE_COMPROMISE</b>	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.
<b>T.UNDETECTED_ACTIVITY</b>	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.
<b>T.SECURITY_FUNCTIONALITY_COMPROMISE</b>	Threat agents may compromise credentials and device data enabling continued access to the network device and its critical data. The compromise of credentials include replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the administrator or device credentials for use by the attacker.
<b>T.PASSWORD_CRACKING</b>	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.
<b>T.SECURITY_FUNCTIONALITY_FAILURE</b>	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 4 Threats

## 3.2 Assumptions

24. The assumptions applicable to the TOE are given in Table 5.

Assumption	Assumption Definition
<b>A.PHYSICAL_PROTECTION</b>	The network device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security and/or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP will not include any requirements on physical tamper protection or other physical attack mitigations. The cPP will not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device.
<b>A.LIMITED_FUNCTIONALITY<sup>2</sup></b>	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example the device should not provide computing platform for general purpose applications (unrelated to networking functionality). If a virtual TOE evaluated as a pND, following Case 2 vNDs as specified in Section 1.2, the VS is considered part of the TOE with only one vND instance for each physical hardware platform. The exception being where components of a distributed TOE run inside more than one virtual machine (VM) on a single VS. In Case 2 vND, no non-TOE guest VMs are allowed on the platform.
<b>A.NO_THRU_TRAFFIC_PROTECTION</b>	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

<sup>2</sup> In accordance with TD0591

	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).
<b>A.TRUSTED_ADMINISTRATOR</b>	The Security Administrator(s) for the network device are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The network device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device.
<b>A.REGULAR_UPDATES</b>	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.
<b>A.ADMIN_CREDENTIALS_SECURE</b>	The administrator's credentials (private key) used to access the network device are protected by the platform on which they reside.
<b>A.RESIDUAL_INFORMATION</b>	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 5 Assumptions

### 3.3 Organizational Security Policies

25. The OSPs applicable to the TOE are given in Table 6.

Policy Name	Policy Definition
<b>P.ACCESS_BANNER</b>	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 6 Organizational Security Policies

## 4 Security Objectives

26. The security objectives have been taken from [NDcPP2.2E] and are reproduced here.

### 4.1 Security Objectives for the TOE

27. The security objectives for the TOE are trivially determined through the inverse of the statement of threats presented in Sect. 4.1 of [NDcPP2.2E] and are not explicitly stated.

### 4.2 Security Objectives for the Operational Environment

28. Security objectives for the Operational Environment are given in Table 7.

Environment Security Objective	Security Objective Definition
<b>OE.PHYSICAL</b>	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
<b>OE.NO_GENERAL_PURPOSE</b>	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.
<b>OE.NO_THRU_TRAFFIC_PROTECTION</b>	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.
<b>OE.TRUSTED_ADMIN</b>	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.  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.
<b>OE.UPDATES</b>	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.
<b>OE.ADMIN_CREDENTIALS_SECURE</b>	The Administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.
<b>OE.RESIDUAL_INFORMATION</b>	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

Table 7 Security Objectives for Operational Environment

### 4.3 Security Objectives rationale

29. Security objectives for the TOE and for the operational environment are taken from [NDcPP2.2E] and reproduced exactly. Therefore, security objectives rationale is identical to that given in Sect. 4 of [NDcPP2.2E] and shall not be reproduced here.

## 5 Security Functional Requirements

30. All security functional requirements are taken from the [NDcPP2.2E] and are presented in accordance with the conventions described in Sect. 6.1 of [NDcPP2.2E] and Sect. 1.4 of this document. Extended component definitions are given in Annex C of [NDcPP2.2E] and are not repeated here. Security Requirements Rationales are also identical to those given in [NDcPP2.2E] and shall not be repeated.

### 5.1 Security Audit (FAU)

#### 5.1.1 Security Audit Data generation (FAU\_GEN)

##### 5.1.1.1 FAU\_GEN.1 Audit data generation

##### 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 8.*

**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 ST, *information specified in column three of Table 8.*

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FAU_STG_EXT.1	None	None
FAU_STG.1	None	None
FCS_CKM.1	None	None
FCS_CKM.2	None	None
FCS_CKM.4	None	None
FCS_COP.1/DataEncryption	None	None
FCS_COP.1/SigGen	None	None
FCS_COP.1/Hash	None	None
FCS_COP.1/KeyedHash	None	None
FCS_RBG_EXT.1	None	None
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_MTD.1/CoreData	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 interactive session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None
FTA_SSL.4	The termination of an interactive session.	None
FTA_TAB.1	None	None
FTP_ITC.1	<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.
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FMT_MOF.1/Functions	None.	None.

FMT_MOF.1/Services	None.	None
FMT_MTD.1/CryptoKeys	None.	None.

Table 8 Security Functional Requirements and Auditable Events

### 5.1.1.2 FAU\_GEN.2 User identity association

#### 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.

### 5.1.2 Security audit event storage (Extended – FAU\_STG\_EXT)

#### 5.1.2.1 FAU\_STG\_EXT.1 Protected Audit Event Storage

##### 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 [

- *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: [oldest log is overwritten] when the local storage space for audit data is full.

#### 5.1.2.2 FAU\_STG.1 Protected audit trail storage (Optional)

##### FAU\_STG.1 Protected audit trail storage

**FAU\_STG.1.1** The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

**FAU\_STG.1.2** The TSF shall be able to prevent unauthorised modifications to the stored audit records in the audit trail.

## 5.2 Cryptographic Support (FCS)

### 5.2.1 Cryptographic Key Management (FCS\_CKM)

#### 5.2.1.1 FCS\_CKM.1 Cryptographic Key Generation (Refinement)

##### FCS\_CKM.1 Cryptographic Key Generation

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

- RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3;
- ECC schemes using “NIST curves” [P-256, P-384, P-521] 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*].

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

#### FCS\_CKM.2 Cryptographic Key Establishment

**FCS\_CKM.2.1<sup>3</sup>** 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*].

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

#### 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 [destruction of reference to the key directly followed by a request for garbage collection];*
- *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 [single overwrite consisting of [zeroes]]*

that meets the following: *No Standard*.

## 5.2.2 Cryptographic Operation (FCS\_COP)

### 5.2.2.1 FCS\_COP.1 Cryptographic Operation

#### 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, CTR] 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, CTR 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, 4096 bits],*
- *Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [P-256, P-384, P-521 bits]*

]

that meet the following: [

<sup>3</sup> In accordance with TD0580, TD0581

- *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, P-521]; 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-512] and cryptographic key sizes [*160, 256 and 512 bits*] and **message digest sizes [160, 256, 512] bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”*.

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

#### **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 [HMAC\_DRBG (any)].

**FCS\_RBG\_EXT.1.2** The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [*[4] software-based noise source, [0] platform-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.

### **5.2.4 Cryptographic Protocols (Extended – FCS\_SSHS\_EXT SSH Protocol)**

#### **5.2.4.1 FCS\_SSHS\_EXT.1 SSH Server Protocol**

#### **FCS\_SSHS\_EXT.1 SSH Server Protocol**

**FCS\_SSHS\_EXT.1.1** The TSF shall implement the SSH protocol that complies with RFC(s) [4251, 4252, 4253, 4254, 4344, 5656, 6668].

**FCS\_SSHS\_EXT.1.2<sup>4</sup>** 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 [*256K*] bytes in an SSH transport connection are dropped.

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

**FCS\_SSHS\_EXT.1.5** The TSF shall ensure that the SSH public-key based authentication implementation uses [ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, ecdsa-sha2-

<sup>4</sup> In accordance with TD0631

*nistp384, ecdsa-sha2-nistp521*] 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, 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, ecdh-sha2-nistp521*] are the only allowed key exchange methods used for the SSH protocol.

**FCS\_SSHS\_EXT.1.8** The TSF shall ensure that within SSH connections the same session keys are used for a threshold of no longer than one hour, and no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

## 5.3 Identification and Authentication (FIA)

### 5.3.1 Authentication Failure Management (FIA\_AFL)

#### 5.3.1.1 FIA\_AFL.1 Authentication Failure Management (Refinement)

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

**FIA\_AFL.1.1** The TSF shall detect when an Administrator configurable positive integer within [*1 to 10*] unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely*.

**FIA\_AFL.1.2** When the defined number of unsuccessful authentication attempts has been met, the TSF shall prevent the offending remote Administrator from successfully authenticating until an Administrator defined time period has elapsed].

### 5.3.2 Password Management (Extended – FIA\_PMG\_EXT)

#### 5.3.2.1 FIA\_PMG\_EXT.1 Password Management

##### **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: [*“!”*, *“@”*, *“#”*, *“\$”*, *“%”*, *“^”*, *“&”*, *“\*”*, *“(”*, *“)”*, *[and all other standard ASCII, extended ASCII and Unicode characters]*];
- b) Minimum password length shall be configurable to between [*10*] and [*20*] characters.

### 5.3.3 User Identification and Authentication (Extended – FIA\_UIA\_EXT)

#### 5.3.3.1 FIA\_UIA\_EXT.1 User Identification and Authentication

##### **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;
- [[Negotiation of SSH session, ICMP echo].

**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.

### 5.3.4 User authentication (FIA\_UAU) (Extended – FIA\_UAU\_EXT)

#### 5.3.4.1 FIA\_UAU\_EXT.2 Password-based Authentication Mechanism

##### 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.

#### 5.3.4.2 FIA\_UAU.7 Protected Authentication Feedback

##### 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**.

## 5.4 Security Management (FMT)

### 5.4.1 Management of functions in TSF (FMT\_MOF)

#### 5.4.1.1 FMT\_MOF.1/ManualUpdate Management of security functions behaviour

##### FMT\_MOF.1/ManualUpdate Management of security functions behaviour

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

#### 5.4.1.2 FMT\_MOF.1/Services Management of security functions behaviour

##### FMT\_MOF.1/Services Management of security functions behaviour

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

#### 5.4.1.3 FMT\_MOF.1/Functions Management of security functions behaviour

##### FMT\_MOF.1/Functions Management of security functions behaviour

**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, handling of audit data*] to *Security Administrators*.

### 5.4.2 Management of TSF Data (FMT\_MTD)

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

##### 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.

#### 5.4.2.2 FMT\_MTD.1/CryptoKeys Management of TSF data

##### 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*.

### 5.4.3 Specification of Management Functions (FMT\_SMF)

#### 5.4.3.1 FMT\_SMF.1 Specification of Management Functions

##### 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 configure audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full);*
  - *Ability to modify the behaviour of the transmission of audit data to an external IT entity;*
  - *Ability to manage the cryptographic keys;*
  - *Ability to configure the cryptographic functionality;*
  - *Ability to configure thresholds for SSH rekeying;*
  - *Ability to re-enable an Administrator account;*
  - *Ability to set the time which is used for time-stamps;*
  - *Ability to manage the trusted public keys database<sup>5</sup>].*

### 5.4.4 Security management roles (FMT\_SMR)

#### 5.4.4.1 FMT\_SMR.2 Restrictions on security roles

##### 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.

## 5.5 Protection of the TSF (FPT)

### 5.5.1 Protection of TSF Data (Extended – FPT\_SKP\_EXT)

#### 5.5.1.1 FPT\_SKP\_EXT.1 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)

##### 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.

<sup>5</sup> In accordance with TD0631

## 5.5.2 Protection of Administrator Passwords (Extended – FPT\_APW\_EXT)

### 5.5.2.1 FPT\_APW\_EXT.1 Protection of Administrator Passwords

#### FPT\_APW\_EXT.1 Protection of Administrator Passwords

FPT\_APW\_EXT.1.1 The TSF shall store passwords in non-plaintext form.

FPT\_APW\_EXT.1.2 The TSF shall prevent the reading of plaintext passwords.

## 5.5.3 TSF testing (Extended – FPT\_TST\_EXT)

### 5.5.3.1 FPT\_TST\_EXT.1 TSF Testing (Extended)

#### 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: [*Power on test, File integrity test, Crypto integrity test, Authentication test, Algorithm known answer tests*].

## 5.5.4 Trusted Update (FPT\_TUD\_EXT)

### 5.5.4.1 FPT\_TUD\_EXT.1 Trusted Update

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

## 5.5.5 Time stamps (Extended – FPT\_STM\_EXT)

### 5.5.5.1 FPT\_STM\_EXT.1 Reliable Time Stamps

#### 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<sup>6</sup> The TSF shall [*allow the Security Administrator to set the time*].

## 5.6 TOE Access (FTA)

### 5.6.1 TSF-initiated Session Locking (Extended – FTA\_SSL\_EXT)

#### 5.6.1.1 FTA\_SSL\_EXT.1 TSF-initiated Session Locking

#### 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.

<sup>6</sup> In accordance with TD0632



## 5.6.2 Session locking and termination (FTA\_SSL)

### 5.6.2.1 FTA\_SSL.3 TSF-initiated Termination (Refinement)

#### 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*.

### 5.6.2.2 FTA\_SSL.4 User-initiated Termination (Refinement)

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

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

## 5.6.3 TOE access banners (FTA\_TAB)

### 5.6.3.1 FTA\_TAB.1 Default TOE Access Banners (Refinement)

#### FTA\_TAB.1 Default TOE Access Banners (Refinement)

**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.

## 5.7 Trusted path/channels (FTP)

### 5.7.1 Trusted Channel (FTP\_ITC)

#### 5.7.1.1 FTP\_ITC.1 Inter-TSF trusted channel (Refinement)

#### 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 [*no communication*].

### 5.7.2 Trusted Path (FTP\_TRP)

#### 5.7.2.1 FTP\_TRP.1/Admin Trusted Path (Refinement)

#### 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*.

## 6 Security Assurance Requirements

31. The security assurance requirements are from Sect. 7 of [NDcPP2.2E] and listed in Table 9.

Assurance Class	Assurance Component
Security Target (ASE)	Conformance claims (ASE_CCL.1)
	Extended components definition (ASE_ECD.1)
	ST introduction (ASE_INT.1)
	Security objectives for the operational environment (ASE_OBJ.1)
	Stated security requirements (ASE_REQ.1)
	Security Problem Definition (ASE_SPD.1)
	TOE summary specification (ASE_TSS.1)
Development (ADV)	Basic functional specification (ADV_FSP.1)
Guidance documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life cycle support (ALC)	Labelling of the TOE (ALC_CMC.1)
	TOE CM coverage (ALC_CMS.1)
Tests (ATE)	Independent testing – conformance (ATE_IND.1)
Vulnerability assessment (AVA)	Vulnerability survey (AVA_VAN.1)

**Table 9 Security Assurance Requirements**

## 7 TOE Summary Specification

### 7.1 Protected communications

32. Local console access is gained by connecting an RJ-45 cable between the console port on the TOE and a workstation with a serial connection client.

#### 7.1.1 Algorithms and zeroization

33. All FIPS-approved cryptographic functions implemented by the TOE are implemented in the following libraries:
- OpenSSL for Junos OS 22.3R1 (based on version 1.0.2p)
  - LibMD for Junos OS 22.3R1 (created from same sources as OpenSSL version 1.0.2p)
  - Kernel for Junos OS 22.3R1 (based on FreeBSD-11 Stable release)
34. Random number generation is implemented in accordance with NIST Special Publication 800-90 using HMAC\_DRBG implemented in the OpenSSL library and kernel library (**FCS\_RBG\_EXT.1.1**). Additionally, SHA-256 and SHA-512 are implemented in the LibMD library and used for password hashing by Junos' MGD daemon. **The appliance is to be operated with FIPS mode enabled.**
35. Each implementation of a cryptographic function by the TOE is CAVP validated. Only FIPS-approved cryptographic functions are used. CAVP certificate references are given in

Library	NIST Standard	Algorithm, Mode, Key size, Function, Hashing, Usage	Cryptographic Operation	SFR(s) supported	CAVP Reference
OpenSSL	FIPS 197, SP 800-38A	AES-CBC (128, 256)	Encrypt, Decrypt in SSH	FCS_COP.1/DataEncryption on FCS_SSHS_EXT.1	<a href="#">A4210</a>
	FIPS 197, SP800-38A	AES-CTR (128, 256)	Encrypt, Decrypt in SSH	FCS_COP.1/DataEncryption on FCS_SSHS_EXT.1	<a href="#">A4210</a>
	FIPS 180-4	SHA1, SHA-256, SHA-384, SHA-512 (byte Oriented)	Message Digest Generation in SSH	FCS_CKM.2 FCS_COP.1/Hash FCS_SSHS_EXT.1	<a href="#">A4210</a>
	FIPS 198-1	HMAC-SHA1, HMAC-SHA-256, HMAC-SHA-512 (byte Oriented)	Message Authentication in SSH and DRBG primitive for OpenSSL DRBG	FCS_COP.1/KeyedHash FCS_SSHS_EXT.1	<a href="#">A4210</a>

Library	NIST Standard	Algorithm, Mode, Keysize, Function, Hashing, Usage	Cryptographic Operation	SFR(s) supported	CAVP Reference
	FIPS 186-4	ECDSA (P-256 w/ SHA-256) ECDSA (P-384 w/ SHA-384) ECDSA (P-521 w/ SHA-512)	SigGen, SigVer, KeyGen for ECDH in SSH	FCS_COP.1/SigGen FCS_SSHS_EXT.1 FPT_TUD_EXT.1	<a href="#">A4210</a>
	SP800-56A	CVL/KAS ECC Key Agreement EC (P-256, SHA-256), ED (P-384, SHA-384), EE (P-521, SHA-512)	Public key Validation, Key Pair Generation, Initiator and Responder for SSH ECDSA	FCS_CKM.1 FCS_CKM.2 FCS_COP.1/SigGen FCS_SSHS_EXT.1	<a href="#">A4419</a>
	FIPS 186-4	RSA PKCS1_V1_5 (n=2048 (SHA 256), n=4096 (SHA 256))	SigGen, SigVer in SSH	FCS_CKM.1 FCS_CKM.2 FCS_COP.1/SigGen FCS_SSHS_EXT.1	<a href="#">A4210</a>
	SP 800-90A	DRBG (HMAC-SHA-256) Prediction Resistance: Enabled	Random Bit Generation for key establishment	FCS_CKM.2 FCS_RBG_EXT.1 FCS_SSHS_EXT.1	<a href="#">A4210</a>
LibMD	FIPS 180-4	SHA-256, SHA-512 (byte Oriented)	Message Digest Generation in password hashing, and in veriexec	FCS_COP.1/Hash FPT_APW_EXT.1 FPT_TST_EXT.1	<a href="#">A4208</a>

Library	NIST Standard	Algorithm, Mode, Key size, Function, Hashing, Usage	Cryptographic Operation	SFR(s) supported	CAVP Reference
Kernel	FIPS 180-4	SHA1, SHA-256, SHA-384, SHA-512 (byte Oriented)	Message Digest Generation in verified-exec kernel support	FCS_COP.1/Hash FPT_TST_EXT.1	<a href="#">A4417</a>
	FIPS 198-1	HMAC-SHA1, HMAC-SHA-256 (byte Oriented)	Message Authentication in Kernel provided DRBG	FCS_COP.1/KeyedHash	<a href="#">A4417</a>
	SP 800-90A	DRBG (HMAC-SHA-256) Prediction Resistance: Enabled	Random Bit Generation, provides /dev/random to user applications such as SSH client and server	FCS_RBG_EXT.1	<a href="#">A4417</a>

36. Table 10.

Library	NIST Standard	Algorithm, Mode, Keysize, Function, Hashing, Usage	Cryptographic Operation	SFR(s) supported	CAVP Reference
OpenSSL	FIPS 197, SP 800-38A	AES-CBC (128, 256)	Encrypt, Decrypt in SSH	FCS_COP.1/DataEncryption FCS_SSHS_EXT.1	<a href="#">A4210</a>
	FIPS 197, SP800-38A	AES-CTR (128, 256)	Encrypt, Decrypt in SSH	FCS_COP.1/DataEncryption FCS_SSHS_EXT.1	<a href="#">A4210</a>
	FIPS 180-4	SHA1, SHA-256, SHA-384, SHA-512 (byte Oriented)	Message Digest Generation in SSH	FCS_CKM.2 FCS_COP.1/Hash FCS_SSHS_EXT.1	<a href="#">A4210</a>
	FIPS 198-1	HMAC-SHA1, HMAC-SHA-256, HMAC-SHA-512 (byte Oriented)	Message Authentication in SSH and DRBG primitive for OpenSSL DRBG	FCS_COP.1/KeyedHash FCS_SSHS_EXT.1	<a href="#">A4210</a>
	FIPS 186-4	ECDSA (P-256 w/ SHA-256) ECDSA (P-384 w/ SHA-384) ECDSA (P-521 w/ SHA-512)	SigGen, SigVer, KeyGen for ECDH in SSH	FCS_COP.1/SigGen FCS_SSHS_EXT.1 FPT_TUD_EXT.1 <sup>7</sup>	<a href="#">A4210</a>
	SP800-56A	CVL/KAS ECC Key Agreement EC (P-256, SHA-256), ED (P-384, SHA-384), EE (P-521, SHA-512)	Public key Validation, Key Pair Generation, Initiator and Responder for SSH ECDSA	FCS_CKM.1 FCS_CKM.2 FCS_COP.1/SigGen FCS_SSHS_EXT.1	<a href="#">A4419</a>
	FIPS 186-4	RSA PKCS1_V1_5 <sup>8</sup> (n=2048 (SHA 256), n=4096 (SHA 256))	SigGen, SigVer in SSH	FCS_CKM.1 FCS_CKM.2 FCS_COP.1/SigGen FCS_SSHS_EXT.1	<a href="#">A4210</a>

<sup>7</sup> P-256 w/ SHA-256 only

<sup>8</sup> Including PKCS#1 v1.5 padding

Library	NIST Standard	Algorithm, Mode, Keysize, Function, Hashing, Usage	Cryptographic Operation	SFR(s) supported	CAVP Reference
	SP 800-90A	DRBG <sup>9</sup> (HMAC-SHA-256) Prediction Resistance: Enabled	Random Bit Generation for key establishment	FCS_CKM.2 FCS_RBG_EXT.1 FCS_SSHS_EXT.1	<a href="#">A4210</a>
LibMD	FIPS 180-4	SHA-256, SHA-512 (byte Oriented)	Message Digest Generation in password hashing, and in veriexec	FCS_COP.1/Hash FPT_APW_EXT.1 FPT_TST_EXT.1	<a href="#">A4208</a>
Kernel	FIPS 180-4	SHA1, SHA-256, SHA-384, SHA-512 (byte Oriented)	Message Digest Generation in verified-exec kernel support	FCS_COP.1/Hash FPT_TST_EXT.1	<a href="#">A4417</a>
	FIPS 198-1	HMAC-SHA1, HMAC-SHA-256 (byte Oriented)	Message Authentication in Kernel provided DRBG	FCS_COP.1/KeyedHash	<a href="#">A4417</a>
	SP 800-90A	DRBG (HMAC-SHA-256) Prediction Resistance: Enabled	Random Bit Generation, provides /dev/random to user applications such as SSH client and server	FCS_RBG_EXT.1	<a href="#">A4417</a>

Table 10 CAVP Certificate References

<sup>9</sup> A Juniper HMAC\_DRBG is used in place of the OpenSSL versions of DRBG.

37. The FIPS approved algorithms are used when the FIPS mode is enabled<sup>10</sup>. The relevant FIPS knobs are specified in [ECG 1] and [ECG 2]. (**FCS\_COP.1/DataEncryption, FCS\_COP.1/SigGen, FCS\_COP.1/Hash, FCS\_COP.1/KeyedHash, FCS\_RBG\_EXT.1, FCS\_CKM.1, FMT\_SMF.1**)
38. Asymmetric keys used by SSH are generated in accordance with FIPS PUB 186-4 Appendix B.3 for RSA Schemes and Appendix B.4 for ECC Schemes. The TOE implements Diffie-Hellman group 14, using the modulus and generator specified by Section 3 of RFC3526. (**FCS\_CKM.2, FCS\_CKM.1**).
39. The TOE acts only as the server for SSH in the supported protocols listed in Table 11:

Protocol	Key Exchange	Authentication	Encryption Algorithms	Data Integrity Algorithms
SSHv2	ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521 Diffie-Hellman group 14 (modp 2048)	ssh-rsa rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256 ecdsa-sha2-nistp384 ecdsa-sha2-nistp521	AES CTR 128 AES CTR 256 AES CBC 128 AES CBC 256	HMAC-SHA-1 HMAC-SHA-256 HMAC-SHA-512

Table 11 Supported SSH Protocols

40. The HMAC algorithms use the values specified in Table 12:

	HMAC-SHA-1	HMAC-SHA-256	HMAC-SHA-512
Key Length	160 bits	256 bits	512 bits
Hash function	SHA-1	SHA-256	SHA-512
Block Size	512 bits	512 bits	1024 bits
Output MAC	160 bits	256 bits	512 bits

Table 12 HMAC Values

41. Junos OS handles zeroization for all CSP, plaintext secret and private cryptographic keys according to Table 13. (**FCS\_CKM.4**).

CSP	Description	Method of storage	Storage location	Zeroization Method
<b>SSH Private Host Key</b>	The first time SSH is configured, the key is generated. Used to identify the host.	Plaintext	File format on SDD	When recommissioned, the config files of the TOE (incl. CSP files such as SSH keys) are removed using the “request vmhost zeroize” option.
	Loaded into memory to complete session establishment	Plaintext	Memory	free() is called by the TOE software at the session termination.
<b>SSH Session Key</b>	Session keys used with SSH, AES 128, 256, hmac-sha-1, hmac-sha2-256 or hmac-sha2-512 key (160, 256 or 512), DH Private Key (2048 or elliptic curve 256/384/521-bits)	Plaintext	Memory	free() is called by the TOE Software at the session termination.

<sup>10</sup> The knob “set system fips chassis level 1” will enforce strict compliance to FIPS and enable restrictions on algorithms and keys sizes as required by FIPS requirements.



CSP	Description	Method of storage	Storage location	Zeroization Method
<b>User Password</b>	Plaintext value entered by user	Plaintext as entered	Processed in Memory	free() is called by the TOE software at the completion of authentication.
		Hashed (HMAC-sha1)	Stored on disk	When the TOE is recommissioned, the config files (including the obfuscated password) are removed using the "request vmhost zeroize" option.
<b>RNG State</b>	Internal state and seed key of RNG	Plaintext	Memory	Handled by kernel, overwritten with zero's at reboot.

Table 13 CSP Storage and Zeroization

42. The CLI implemented by the TOE does not permit the viewing of cryptographic keys. The keys are protected through the enforcement of kernel-level file access rights which limit access to the contents of cryptographic key containers to processes with cryptographic rights or shell users with root permission. Security Administrators do not have root access rights to the kernel (**FPT\_SKP\_EXT.1**)

### 7.1.2 Random Bit Generation

43. The TOE generates random bits in accordance with NIST SP 800-90 using HMAC\_DRBG, SHA-256. The RBG in the RE-S-1600x8 RE is seeded from the following software sources of entropy:
- **RANDOM\_INTERRUPT**: Hardware devices whose real-time interrupts are known to provide some amount of entropy. The internal representation of handling these interrupts provides entropy. This source can provide entropy both during system boot and steady state.
  - **RANDOM\_NET\_ETHER**: Timings (CPU counter values at the time of the event) together with the internal representation of network packets are used to harvest entropy that is further fed into the DRBG.
  - **RANDOM\_FS\_ETIME**: Associated to the time slices during access of the temporary file storage such as a tmpfs. The continuous creation, access and destruction of files in the temporary space in a running system provides randomness. Unpredictability comes from the timing of the time slices.
  - **RANDOM\_ATTACH**: Associated with the elapsed cycle count for each device-driver as it attaches to the associated devices in the system and provides entropy during boot-up. Unpredictability comes from the timing of the attachments.

### 7.1.3 SSH

44. The TOE implements a SSHv2 server to implement Trusted Channels and Trusted Paths between itself and a remote audit server and between itself and a remote management workstation. SSH connection protects the content of the communication from unauthorized disclosure or modification. (**FPT\_ITC.1, FPT\_TRP.1/Admin**)
45. Export of audit information to a secure, remote server constitutes a Trusted Channel which is achieved by setting up an event trace monitor that sends event log messages by NETCONF over SSH to the remote audit server. The remote audit server initiates the connection. (**FPT\_ITC.1, FCS\_SSHS\_EXT.1**)

46. For remote administration, the remote administrator initiates communication with the TOE through a SSH tunnel created by a SSH session. Authentication of the peers is through public key cryptography. (*FTP\_TRP.1/Admin, FCS\_SSHS\_EXT.1*)
47. The SSH server is implemented in accordance with RFCs 4251, 4252, 4253, 4254, 4344, 5656 and 6668. The TOE implements both public key and password-based authentication of administrative users. The conformance to RFCs is given in Table 14.

RFC	Summary	TOE implementation of Security
RFC 4251	The Secure Shell (SSH) Protocol Architecture	<p><b>Host Keys:</b> The TOE uses an ECDSA Host Key for SSH v2, with a key size of 256 bits or greater, which is generated on initial setup of the TOE. It can be de-configured via the CLI and the key will be deleted and thus unavailable during connection establishment. This key is randomly generated to be unique to each TOE instance. The TOE presents the client with its public key and the client matches this key against its known_hosts list of keys. When a client connects to the TOE, the client will be able to determine if the same host key was used in previous connections, or if the key is different (per the SSHv2 protocol). Junos OS also supports RSA-based key establishment schemes with a key size of 2048 bits.</p> <p><b>Policy Issues:</b> The TOE implements all mandatory algorithms and methods. The TOE can be configured to accept public-key based authentication and/or password-based authentication. The TOE does not require multiple authentication mechanisms for users. The TOE allows port forwarding and sessions to clients. The TOE has no X11 libraries or applications and X11 forwarding is prohibited.</p> <p><b>Confidentiality:</b> The TOE does not accept the “none” cipher. Supports AES-CBC-128, AES-CBC-256, AES-CTR-128, AES-CTR-256 encryption algorithms for protection of data over SSH and uses keys generated in accordance with “ssh-rsa”, “rsa-sha2-256”, “rsa-sha2-512”, “ecdsa-sha2-nistp256”, “ecdsa-sha2-nistp384” or “ecdsa-sha2-nistp521” to perform public-key based device authentication. For ciphers whose blocksize <math>\geq 16</math>, the TOE rekeys every <math>(2^{32}-1)</math> bytes. The client may explicitly request a rekeying event as a valid SSHv2message at any time and the TOE will honor this request. Re-keying of SSH session keys can be configured using the sshd_config knob. The data-limit must be between 51200 and 4294967295 <math>(2^{32}-1)</math> bytes and the time-limit must be between 1 and 1440 minutes. In the evaluated deployment the time-limit must be set within 1 and 60 minutes.</p> <p><b>Denial of Service:</b> When the SSH connection is brought down, the TOE does not attempt to re-establish it.</p> <p><b>Ordering of Key Exchange Methods:</b> Key exchange is performed only using one of the supported key exchange algorithms, which are ordered as follows: ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521 (all specified in RFC 5656), diffie-hellman-group14-sha1 (specified in RFC 4253).</p> <p><b>Debug Messages:</b> The TOE sshd server does not support debug messages via the CLI.</p> <p><b>End Point Security:</b> The TOE permits port forwarding.</p> <p><b>Proxy Forwarding:</b> The TOE permits proxy forwarding.</p> <p><b>X11 Forwarding:</b> The TOE does not support X11 forwarding.</p>

RFC	Summary	TOE implementation of Security
RFC 4252	The Secure Shell (SSH) Authentication Protocol	<p><b>Authentication Protocol:</b> The TOE does not accept the “none” authentication method. The TOE implements a timeout period of 30seconds for authentication of the SSHv2 protocol and provides a limit of three failed authentication attempts before sending a disconnect to the client.</p> <p><b>Authentication Requests:</b> The TOE does not accept authentication if the requested service does not exist. The TOE does not allow authentication requests for a non-existent username to succeed – it sends back a disconnect message as it would for failed authentications and hence does not allow enumeration of valid usernames. The TOE denies “none” authentication method and replies with a list of permitted authentication methods.</p> <p><b>Public Key Authentication Method:</b> The TOE supports public key authentication for SSHv2 session authentication. Authentication succeeds if the correct private key is used. The TOE does not require multiple authentications (public key and password) for users.</p> <p><b>Password Authentication Method:</b> The TOE supports password authentication. Expired passwords are not supported and cannot be used for authentication.</p> <p><b>Host-Based Authentication:</b> The TOE does not support the configuration of host-based authentication methods.</p>
RFC 4253	The Secure Shell (SSH) Transport Layer Protocol	<p><b>Encryption:</b> The TOE offers the following for encryption of SSH sessions: aes128-cbc and aes256-cbc, aes128-ctr, aes256-ctr. The TOE permits negotiation of encryption algorithms in each direction. The TOE does not allow the “none” algorithm for encryption.</p> <p><b>Maximum Packet length:</b> Packets greater than 256Kbytes in an SSH transport connection are dropped and the connection is terminated by Junos OS.</p> <p><b>Data Integrity:</b> The TOE permits negotiation of HMAC-SHA1 in each direction for SSH transport.</p> <p><b>Key Exchange:</b> The TOE supports diffie-hellman-group14-sha1.</p> <p><b>Key Re-Exchange:</b> The TOE performs a re-exchange when SSH_MSG_KEXINIT is received.</p>

RFC	Summary	TOE implementation of Security
RFC 4254	Secure Shell (SSH) Connection Protocol	<p><b>Multiple channels:</b> The TOE assigns each channel a number (as detailed in RFC 4251, see above).</p> <p><b>Data transfers:</b> The TOE supports a maximum window size of 256K bytes for data transfer.</p> <p><b>Interactive sessions:</b> The TOE only supports interactive sessions that do NOT involve X11 forwarding.</p> <p><b>Forwarded X11 connections:</b> This is not supported in the TOE.</p> <p><b>Environment variable passing:</b> The TOE only sets variables once the server process has dropped privileges.</p> <p><b>Starting shells/commands:</b> The TOE supports starting one of shell, application program or command (only one request per channel). These will be run in the context of a channel, and will not halt the execution of the protocol stack.</p> <p><b>Window dimension change notices:</b> The TOE will accept notifications of changes to the terminal size (dimensions) from the client.</p> <p><b>Port forwarding:</b> This is fully supported by the TOE.</p>
RFC4344	Secure Shell (SSH) Transport Layer Encryption Modes	<p><b>Encryption Modes:</b> The TOE implements the recommended modes aes128-ctr and aes256-ctr (it does not implement the recommended modes aes192-ctr or 3des-ctr, nor does it implement any of the optional modes).</p>
RFC5656	SSH ECC Algorithm Integration	<p><b>ECDH Key Exchange:</b> The support key exchange methods specified in this RFC are ecdh-sha2-nistp256, ecdh-sha2-nistp384, or ecdh-sha2-nistp521. The client matches the key against its known_hosts list of keys.</p> <p><b>Hashing:</b> Junos OS supports cryptographic hashing via the SHA-256 and SHA-512 algorithms, provided it has a message digest size of either 256 or 512 bits.</p> <p><b>Required Curves:</b> All required curves are implemented: ecdh-sha2-nistp256, ecdh-sha2-nistp384, or ecdh-sha2-nistp521. None of the Recommended Curves are supported as they are not included in [NDCPP2.2E].</p>
RFC 6668	sha2-Transport Layer Protocol	<p><b>Data Integrity Algorithms:</b> Both the recommended and optional algorithms hmac-sha1, mac-sha2-256 and hmac-sha2-512 (respectively) are implemented for SSH transport.</p>

Table 14 SSH RFC conformance

## 7.2 Administrator Authentication

48. The TOE enforces binding between human users and subjects. The Security Administrator is responsible for provisioning user accounts, and only the Security Administrator can do so. (*FMT\_SMR.2, FMT\_MTD.1/CoreData*)
49. Users are configured under “system login user” and exported to the password database `/var/etc/master.passwd`. A Junos user is an entry in the password database. Each entry in the password database has fields corresponding to the attributes of “system login user”, including username, (obfuscated) password and login class.
50. The internal architecture supporting Authentication includes an active process, associated linked libraries and supporting configuration data. The Authentication process and library are
  - `login()`
  - PAM Library module

51. Following TOE initialization, the `login()` process is listening for a connection at the local console. This 'login' process can be accessed through either direct connection to the local console or following successful establishment of a remote management connection over SSH, when a login prompt is displayed.
52. This login process identifies and authenticates the user using PAM operations. The login process does two things; it first establishes that the requesting user is whom they claim to be and second provides them with an interactive Junos Command interactive command line interface (CLI).
53. The SSH daemon supports public key authentication by looking up a public key in an authorized keys file located in the directory '.ssh' in the user's home directory (i.e. '~/.ssh/') and this authentication method will be attempted before any other if the client has a key available (**FIA\_UIA\_EXT.1**). The SSH daemon will ignore the authorized keys file if it or the directory '.ssh' or the user's home directory are not owned by the user or are writeable by anyone else.
54. For password authentication, `login()` interacts with a user to request a username and password to establish and verify the user's identity. The username entered by the administrator at the username prompt is reflected to the screen, but no feedback to screen is provided while the entry made by the administrator at the password prompt until the Enter key is pressed (**FIA\_UAU.7**). `login()` uses PAM Library calls for the actual verification of this password. The password is hashed and compared to the stored value, and success/failure is indicated to `login()`, (**FIA\_UIA\_EXT.1**). PAM is used to support authentication management, account management, session management and password management. Login primarily uses the session management and password management functionality offered by PAM.
55. The retry-options can be configured to specify the action to be taken if the administrator fails to enter a valid username/password pair when authenticating from a network management station (**FMT\_MTD.1/CoreData**). The retry-options are applied following the first failed login attempt for a given username (**FIA\_AFL.1**). The length of delay (5-10 seconds) after each failed attempt is specified by the backoff-factor, and the increase of the delay for each subsequent failed attempt is specified by the backoff-threshold (1-3). The tries-before-disconnect sets the maximum number of times (1-10) the administrator is allowed to enter a password to attempt to log in to the device through SSH before the connection is disconnected. The lockout-period sets the amount of time in minutes before the administrator can attempt to log in to the device after being locked out due to the number of failed login attempts (1-43,200 minutes). Even when an account is locked for remote access to the TOE, an administrator is always able to login locally through the serial console and the administrator can attempt authentication via remote access after the maximum timeout period of 24 hours.
56. The TOE requires users to enter correct identification and authentication data before any controlled access is granted. Prior to authentication, the TOE shall only allow displaying of an access banner, responding to an ICMP echo, and negotiation of a SSH session. (**FIA\_UAU\_EXT.2**)
57. Passwords are case-sensitive, alphanumeric values. The password has a minimum length of 10 characters and maximum length of 20 characters. It must contain characters from at least two different character sets (upper, lower, numeric, punctuation). Any standard ASCII, extended ASCII and Unicode characters as well as the following special characters can be selected when choosing a password: "!", "@", "#", "\$", "%", "^", "&", "\*", "(", and ")". (**FIA\_PMG\_EXT.1**)
58. Locally stored authentication credentials are protected (**FPT\_APW\_EXT.1**):
  - The password is hashed when stored using hmac-sha1, sha256 or sha512.
  - Authentication data for public key-based authentication methods are stored in a directory owned by the user (and typically with the same name as the user). This directory contains

the files '.ssh/authorized\_keys' and '.ssh/authorized\_keys2' which are used for SSH public key authentication.

59. The TOE allows Security Administrators to configure an access banner for local and remote SSH connections for display in the authentication prompt. The banner may display warnings against unauthorized access to the secure switch as well as any other information that the Security Administrator wishes to communicate. (**FTA\_TAB.1**)
60. User sessions (local and remote) can be terminated by users (**FTA\_SSL.4**). The administrative user can logout of existing CLI and remote SSH sessions by typing logout to exit the session and the TOE ensures that the current contents unreadable after the admin initiates the termination. No user activity can take place until the user re-identifies and authenticates.
61. Security Administrators may configure the TOE to terminate user sessions after a period of inactivity. (**FTA\_SSL\_EXT.1, FTA\_SSL.3**) For each user session the TOE maintains a count of clock cycles since the last activity. The count is reset each time there is activity related to the user session. When the counter reaches the number of clock cycles equating to the configured period of inactivity, the user session is locked out. The TOE also overwrites the display device and makes the current contents unreadable after the local interactive session is terminated due to inactivity, thus disabling any further interaction with the TOE.

### 7.3 Correct Operation

62. The following self-tests are executed on power-on to verify the correct operation of the TOE software (**FPT\_TST\_EXT.1**):
  - Power on test – determines the boot-device responds and performs a memory size check to confirm the amount of available memory.
  - File integrity test – verifies integrity of all mounted signed packages, to assert that system files have not been tampered with. To test the integrity of the software, the fingerprints of the executables and other immutable files are regenerated and validated against the SHA1 fingerprints contains in the manifest file.
  - Crypto integrity test – checks integrity of major CSPs, such as SSH hostkeys.
  - Authentication error – verifies that verixec is enabled and operates as expected using /opt/sbin/kats/cannot-exec.real.
  - Kernel, libmd, OpenSSL, SSH – verifies correct output from known answer tests for appropriate algorithms.
63. Juniper Networks devices run only binaries supplied by Juniper Networks. Within the package, each Junos OS software image includes fingerprints of the executables and other immutable files. Junos software will not execute any binary without a validating registered fingerprint. This feature protects the system against unauthorized software and activity that might compromise the integrity of the device. These self-tests ensure that only authorized executables are allowed to run thus ensuring the correct operation of the TOE.
64. In the event of a transiently corrupt state or failure condition, the system will panic; the event will be logged and the system restarted, having ceased to process network traffic. When the system restarts, the system boot process does not succeed without passing all applicable self-tests. This automatic recovery and self-test behavior, is discussed in Chapter 11 of the [ECG 1] and [ECG 2].
65. When any self-test fails, the device halts in an error state. No command line input or traffic to any interface is processed. The device must be power cycled to attempt to return to operation. This self-test behavior is discussed in [ECG 1] and [ECG 2]. (**FPT\_TST\_EXT.1**)

## 7.4 Trusted Update

66. Security Administrators are able to query the current version of the TOE software using the CLI command “show version” (**FPT\_TUD\_EXT.1**) If a new version is available, they may initiate an update of the TOE software. Junos OS does not provide partial updates for the TOE. Updates are downloaded and applied manually. There is no automatic updating of the Junos OS. The installable software package containing the Junos OS has a digital signature that is checked when the Security Administrator attempts to install the package. (**FPT\_TUD\_EXT.1, FMT\_SMF.1, FMT\_MOF.1/ManualUpdate,**)
67. The Junos OS kernel maintains a set of fingerprints (SHA1 digests) for executable files and other files which should be immutable, as described in Section 7.3. The manifest file is signed using the Juniper package signing key, and is verified by the TOE. ECDSA (P-256) with SHA-256 is used for digit signature package verification.
68. The fingerprint loader will only process a manifest for which it can verify the signature. Without a valid digital signature, an executable cannot be run. When the command is issued to install an update, the manifest file for the update is verified and stored, and each executable/immutable file is verified before being executed. If any of the fingerprints in an update are not correctly verified, the TOE uses the last known verified image. (**FCS\_COP.1/SigGen, FPT\_TUD\_EXT.1**)

## 7.5 Audit

69. The TOE creates and stores audit records for a right set of events. Each event and the content recorded is detailed in Table 8 (**FAU\_GEN.1**). Auditing is implemented using syslog.
70. The detail of what events are to be recorded by syslog are determined by the logging level specified the “level” argument of the “set system syslog” CLI command. To ensure compliance with the requirements the audit knobs detailed in [ECG 1] and [ECG 2] must be configured.
71. As a minimum, Junos OS records with each log entry the date and time of the event and/or reaction, the type of event and/or reaction, subject identity (where applicable) and the outcome (success or failure) of the event (where applicable).
72. 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):
  - SSH session keys– key reference provided by process id
  - 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
73. For SSH (ephemeral) session keys the PID is used as the key reference to relate the key generation and key destruction audit events. The key destruction event is recorded as a session disconnect event. For example, key generation and key destruction events for a single SSH session key would be reflected by records similar to the following:
 

```
Sep 27 15:09:36 yeti sshd[6529]: Accepted publickey for root from 10.163.18.165 port 45336
ssh2: RSA SHA256:l1vri77TPQ4VaupE2NMYiUXPnGkqBWlgD5vW0OuglGI
...
Sep 27 15:09:40 yeti sshd[6529]: Received disconnect from 10.163.18.165 port 45336:11:
disconnected by user
Sep 27 15:09:40 yeti sshd[6529]: Disconnected from 10.163.18.165 port 45336
```
74. SSH keys used for trusted channels are NOT deleted by mgd when SSH is de-configured. Hence, the only time SSH keys used for trusted channels are deleted is when a “request vmhost zeroize”



action is performed and the whole appliance is zeroized (which by definition cannot be recorded).

75. All events recorded by syslog are timestamped. The clock function of Junos OS provides a source of date and time information for the appliance, used in audit timestamps, which is maintained using the hardware Time Stamp Counter as the clock source. (*FAU\_GEN.2, FPT\_STM\_EXT.1*)
76. Syslog can be configured to store the audit logs locally (*FAU\_STG\_EXT.1*) and to send them to one or more syslog log servers in real time via Netconf over SSH. If the sending of the audit logs is done automatically without Administrator intervention (*FAU\_STG.1, FMT\_MOF.1/Functions*). Local audit log are stored in /var/log/ in the underlying filesystem. Only a Security Administrator can read or delete log and archive files through the CLI interface or through direct access to the filesystem. The syslogs are automatically deleted locally according to configurable limits on storage volume. The default maximum size is 1Gb. The default maximum size can be modified by the user, using the "size" argument for the "set system syslog" CLI command.
77. The Junos OS defines an active log file and a number of "archive" files (10 by default, but configurable from 1 to 1000). When the active log file reaches its maximum size, the logging utility closes the file, compresses it, and names the compressed archive file 'logfile.0.gz'. The logging utility then opens and writes to a new active log file. When the new active log file reaches the configured maximum size, 'logfile.0.gz' is renamed 'logfile.1.gz', and the active log file is closed, compressed, and renamed 'logfile.0.gz'. When the maximum number of archive files is reached and when the size of the active file reaches the configured maximum size, the contents of the oldest archived file are deleted so the current active file can be archived.
78. A 1Gb syslog file takes approximately 0.25Gb of storage when archived. Syslog files can acquire complete storage allocated to /var filesystem, which is platform specific. However, when the filesystem reaches 92% storage capacity an event is raised to the administrator but the eventd process (being a privileged process) still can continue using the reserved storage blocks. This allows the syslog to continue storing events while the administrator frees the storage. If the administrator does not free the storage in time and the /var filesystem storage becomes exhausted a final entry is recorded in the log reporting "No space left on device" and logging is terminated. The appliance continues to operate in the event of exhaustion of audit log storage space.

## 7.6 Management

79. Accounts assigned to the Security Administrator role are used to manage Junos OS in accordance with [NDcPP2.2E]. User accounts in the TOE have the following attributes: user identity (user name), authentication data (password) and role (privilege). The Security Administrator is associated with the defined login class "security-admin", which has the necessary permission set to permit the administrator to perform all tasks necessary to manage Junos OS in accordance with the requirements of [NDcPP2.2E]. (*FMT\_SMR.2*)
80. The TOE allows user access either through the system console or remotely over SSH. Users are required to provide unique identification and authentication data before any access to the system is granted, as detailed in Sect. 7.2. (*FMT\_SMR.2, FMT\_SMF.1*)
81. The Security Administrator has the capability to:
  - Administer the TOE locally via the serial ports on the physical device or remotely over an SSH connection.
  - Initiate a manual update of TOE software (*FMT\_MOF.1/ManualUpdate*):
    - Query currently executing version of TOE software (*FPT\_TUD\_EXT.1*)
    - Verify update using digital signature (*FPT\_TUD\_EXT.1*)

- Manage Functions:
  - Transmission of audit data to an external IT entity, including Start/stop and modify the behaviour of the trusted communication channel to external syslog server (netconf over SSH) and the trusted path for remote Administrative sessions (SSH) (**FMT\_MOF.1/Functions, FMT\_MOF.1/Services, FMT\_SMF.1**)
  - Handling of audit data, including setting limits of log file size (**FMT\_MOF.1/Functions**)
- Manage TSF data (**FMT\_MTD.1/CoreData**)
  - Create, modify, delete administrator accounts, including configuration of authentication failure parameters
  - Reset administrator passwords
  - Re-enable an Administrator account (**FIA\_AFL.1**);
- Manage crypto keys (**FMT\_MTD.1/CryptoKeys**):
  - SSH key generation (ecdsa, ssh-rsa)
- Perform management functions (**FMT\_SMF.1**):
  - Configure the access banner (**FTA\_TAB.1**)
  - Configure the session inactivity time before session termination or locking, including termination of session when serial console cable is disconnected (**FTA\_SSL\_EXT.1, FTA\_SSL.3**)
  - Manage cryptographic functionality (**FCS\_SSHS\_EXT.1**), including:
    - ssh ciphers
    - hostkey algorithm
    - key exchange algorithm
    - hashed message authentication code
    - thresholds for SSH rekeying
  - Set the system time (**FPT\_STM\_EXT.1**)

82. Detailed topics on the secure management of Junos OS are discussed in [ECG 1] and [ECG 2].

## 8 Glossary

AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	Application Program Interface
cPP	collaborative Protection Profile
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFP	C Form-factor Pluggable
CSP	Critical security parameter
DH	Diffie Hellman
EAL	Evaluation Assurance Level
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
EP	Extended Package, defined in [CC1]
ESP	Encapsulating Security Payload
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standard
HMAC	Keyed-Hash Authentication Code
I&A	Identification and Authentication
ID	Identification
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPv6	Internet Protocol Version 6
ISO	International Organization for Standardization
IT	Information Technology
Junos	Juniper Operating System
MIC	Modular Interface Cards
MPC	Modular Port Concentrator
MS-MPC	MultiServices Modular Port Concentrator
NAT	Network Address Translation
NDcPP	Network Device collaborative Protection Profile
NTP	Network Time Protocol
OSI	Open Systems Interconnect
OSP	Organizational Security Policy
PAM	Pluggable Authentication Module
PFE	Packet Forwarding Engine
PIC/PIM	Physical Interface Card/Module
PKI	Public Key Infrastructure
POE	Power over Ethernet
PP	Protection Profile
PRNG	Pseudo Random Number Generator
RE	Routing Engine
RFC	Request for Comment
RNG	Random Number Generator
RSA	Rivest, Shamir, Adelman
SA	Security Association
SFP	Small Form-factor Pluggable
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SSH	Secure Shell

SSL	Secure Sockets Layer
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF interfaces
UDP	User Datagram Protocol