Juniper Networks® ACX7024 and ACX7024X routers with Junos® OS Evolved Version 24.4R2 Security Target

Version:	**1.4**
Status:	**Final**
Last Update:	**2025-11-05**
Classification:	**Public**
Authors:	**atsec information security corporation**

Trademarks

Juniper Networks, Junos EVO, Junos OS Evolved, and the Juniper logo are registered trademarks of Juniper Networks, Inc.

OpenSSL is a registered trademark of OpenSSL Software Foundation.

Legal Notice

This document is provided AS IS with no express or implied warranties. Use the information in this document at your own risk.

This document may be reproduced or distributed in any form without prior permission provided the copyright notice is retained on all copies. Modified versions of this document may be freely distributed provided that they are clearly identified as such, and this copyright is included intact.

Revision History

Version	Date	Author(s)	Changes to Previous Revision
1.0	2025-06-27	Alejandro Masino	First version.
1.1	2025-07-08	Alejandro Masino	Add abbreviations. Apply feedback from Juniper's review.
1.2	2025-08-25	Alejandro Masino	Specify RSA modulus size in key generation and signature generation and verification. Add keyboard-interactive authentication for SSHv2. Restore FMT_MOF.1/Services. Update CAVP certs. Address ECRs.
1.3	2025-10-03	Alejandro Masino	Editorial updates. Update versions of Junos® and OpenSSL. Update TSS based on evaluator's comments. Update CAVP certs.
1.4	2025-11-05	Alejandro Masino	Update Intel processor names. Resolve ECRs.

Copyright © 2025 Juniper Networks, Inc.

Table of Contents

Copyright © 2025 Juniper Networks, Inc.

Copyright © 2025 Juniper Networks, Inc.

List of Tables

Copyright © 2025 Juniper Networks, Inc.

List of Figures

1 Introduction

1.1 Security Target Identification

Title:	Juniper Networks® ACX7024 and ACX7024X routers with Junos® OS Evolved Version 24.4R2 Security Target
Version:	1.4
Status:	Final
Date:	2025-11-05
Sponsor:	Juniper Networks, Inc.
Developer:	Juniper Networks, Inc.
Validation Body:	NIAP
Validation ID:	VID 11626
Keywords:	ACX7024, ACX7024X, Juniper Networks, Common Criteria, NIAP, CPP_ND_V3.0E, PKG_SSH_V1.0

1.2 TOE Identification

The TOE is Juniper Networks® ACX7024 and ACX7024X routers with Junos® OS Evolved version 24.4R2.

1.3 TOE Type

The TOE type is a Network Device.

1.4 TOE Overview

The Target of Evaluation (TOE) is the Juniper Networks® ACX7024 and ACX7024X routers with Junos® OS Evolved version 24.4R2. The TOE meets the requirements of the collaborative Protection Profile for Network Devices Version 3.0e ([CPP_ND_V3.0E]☝) and the Functional Package for SSH Version 1.0 [PKG_SSH_V1.0]☝.

The TOE is a fixed configuration packet transport router which provides the foundation for a scale-out core backbone architecture.The TOE is composed of a hardware component (the ACX7024 or ACX7024X chassis), and the Junos® OS Evolved Operating System. In concert, they implement the functions of a complete network appliance. The TOE is a physical and standalone network device.

The TOE provides the following security functionality:

- *Security Audit*. The TOE implements an audit function to collect detailed information about the state of the TOE to allow the administrator[1] to troubleshoot the TOE and investigate possible security-related incidents. Security logs are stored locally and remotely.
- *Cryptographic Support*. The TOE implements a suite of cryptographic algorithms for the protection of user passwords, secure communication during transmission of audit logs and remote management sessions, authentication in the NTP protocol, and digital signature verification for TOE trusted updates.

[1] The terms "administrator" and "security administrator" are used interchangeably in this document.

- *Identification and Authentication*. The TOE implements password-based and public-key-based authentication to users, allowing configuration of password policies and authentication failure policies.
- *Security Management*. The TOE implements a Command Line Interface (CLI) made available to administrators for managing the TOE.
- *Protection of the TSF*. The TOE protects itself from tampering, provides a reliable system clock for timestamps, and verifies the integrity of software updates.
- *TOE Access*. The TOE allows the display of a warning banner when initiating management sessions and monitors and closes idle sessions.
- *Trusted Path and Trusted Channels*. The TOE implements secure channels to protect audit data sent to external servers and remote management sessions initiated by administrators.

1.5 TOE Description

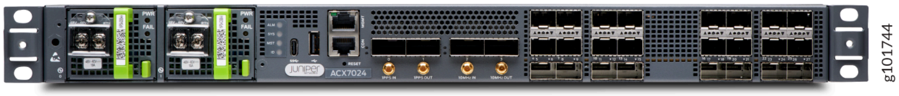
The TOE is a network device which includes hardware and software components. The TOE hardware is either the ACX7024 or the ACX7024X chassis. The chassis implements the casing and the physical ports, the motherboard, and the hardware foundation for all those functions of the TOE which are implemented in hardware. The TOE software is implemented as the Junos® OS Evolved Operating System.

The ACX7024 and ACX7024X, from the ACX7000 family, are compact, fixed, 1 U (24 cm deep), high-performance multiservice routers. Designed for Cloud Metro deployments in industrial and commercial temperature environments, both routers incorporate 6 integrated fans (5+1 redundancy) for front-to-back airflow. They come with 2x field replaceable AC or DC power supplies (1+1 redundancy).

The ACX7024 and ACX7024X routers share identical features and capabilities while providing operators with key differentiators to fit the exacting needs of each deployment. The ACX7024 is designed to work in less-than-desirable, industrial-temperature applications. The ACX7024X, with its eight-core processor and 64MB of RAM, provides high-scale and low-latency capabilities in commercial-temperature applications.

The ACX7024 is an industrial-rated (I-Temp) multiservice router. The ACX7024 router's fixed ports include 24 multi-rate (SFP28) ports, each configurable as 1GbE, 10GbE, and 25GbE, enabling operators to perform upgrades on a port-by-port basis. An additional 4 fixed (QSFP-28) 100GbE uplinks are available to support scale.

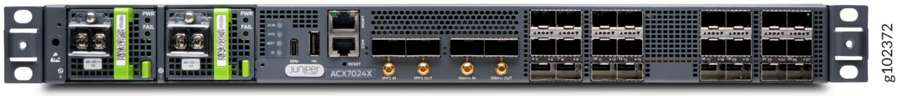
Figure 1: Juniper Networks® ACX7024



The ACX7024X is a commercial-rated (C-Temp), high scale multiservice router. The ACX7024X router fixed ports include 24 multi-rate (SFP28) ports, each configurable as 1GbE, 10GbE, and 25GbE, enabling operators to perform upgrades on a port-by-port basis. An additional 4 fixed (QSFP-28) 100GbE uplinks are available to support scale.

Copyright © 2025 Juniper Networks, Inc.

Figure 2: Juniper Networks® ACX7024X



The table below summarizes a comparison between both models.

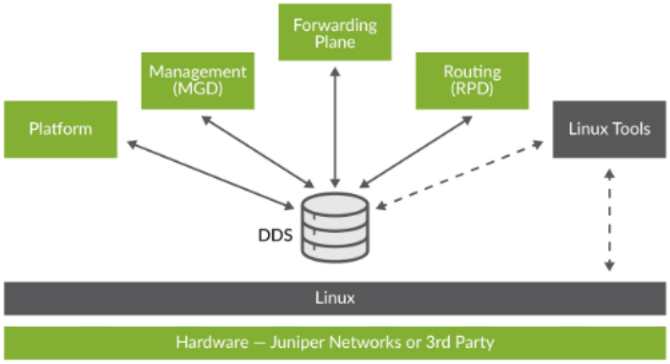
Table 1: ACX7024 and ACX7024X Summary Specification

Specification	ACX7024	ACX7024X
CPU	Intel Denverton Atom C3508 4-core 1.60GHz, 8 MB cache	Intel Denverton Atom C3758R 8-core 2.40GHz, 16 MB cache
Memory	RAM: 16GB DDR4	RAM: 64GB DDR4
Chassis type	Fixed	Fixed
ASIC throughput	360 Gbps	360 Gbps
Interfaces	24x 1GbE/10GbE/25GbE SFP28 4x 100GbE QSFP-28	24x 1GbE/10GbE/25GbE SFP28 4x 100GbE QSFP-28
Synchronization interfaces	• 1x RJ-45 port + TOD • 1 PPS/10 MHz input and output • GNSS Antenna (via USB)	• 1x RJ-45 port + TOD • 1 PPS/10 MHz input and output • GNSS Antenna (via USB)
Dimensions (W x H x D)	19 x 1.75 x 9.6 in (48.2 x 4.4 x 24.4 cm)	19 x 1.75 x 9.6 in (48.2 x 4.4 x 24.4 cm)
Weight (lb/kg) fully configured	12.5 lb (5.66 kg)	12.5 lb (5.66 kg)
Power (DC)	-48 VDC through -60 VDC	-48 VDC through -60 VDC
Power (AC)	90 VAC to 264 VAC	90 VAC to 264 VAC
Typical power draw (without optics)*	97 W @ 25° C	97 W @ 25° C
Maximum power draw (without optics)*	150 W	150 W
Operating temperature	-40° C to +65° C GR3108-class-2	0° C to 40° C GR-63-CORE
Cooling	6 fans, front-to-back airflow, baffle for side-side	6 fans, front-to-back airflow, baffle for side-side
Humidity	5% to 90% RH (noncondensing) operating	5% to 90% RH (noncondensing) operating

The Junos® OS Evolved is the Juniper Linux-based operating system for network devices. It implements a flexible Software Defined Networking (SDN) allowing the tailoring of the software to several applications. The Junos® OS Evolved is a horizontal software layer that decouples the application processes from the hardware on which the processes run. Effectively, this decoupling creates a general-purpose software infrastructure spanning all different computing resources on the system. Application processes (protocols, services, and so on) run on top of this infrastructure and communicate with each other by publishing and consuming (that is, subscribing to) the state.

State is the retained information or status about physical or logical entities that the system preserves and shares across the system and supplies during restarts. State includes both operational and configuration state, including committed configuration, interface state, routes, and hardware state.

Figure 3: TOE Software Architecture



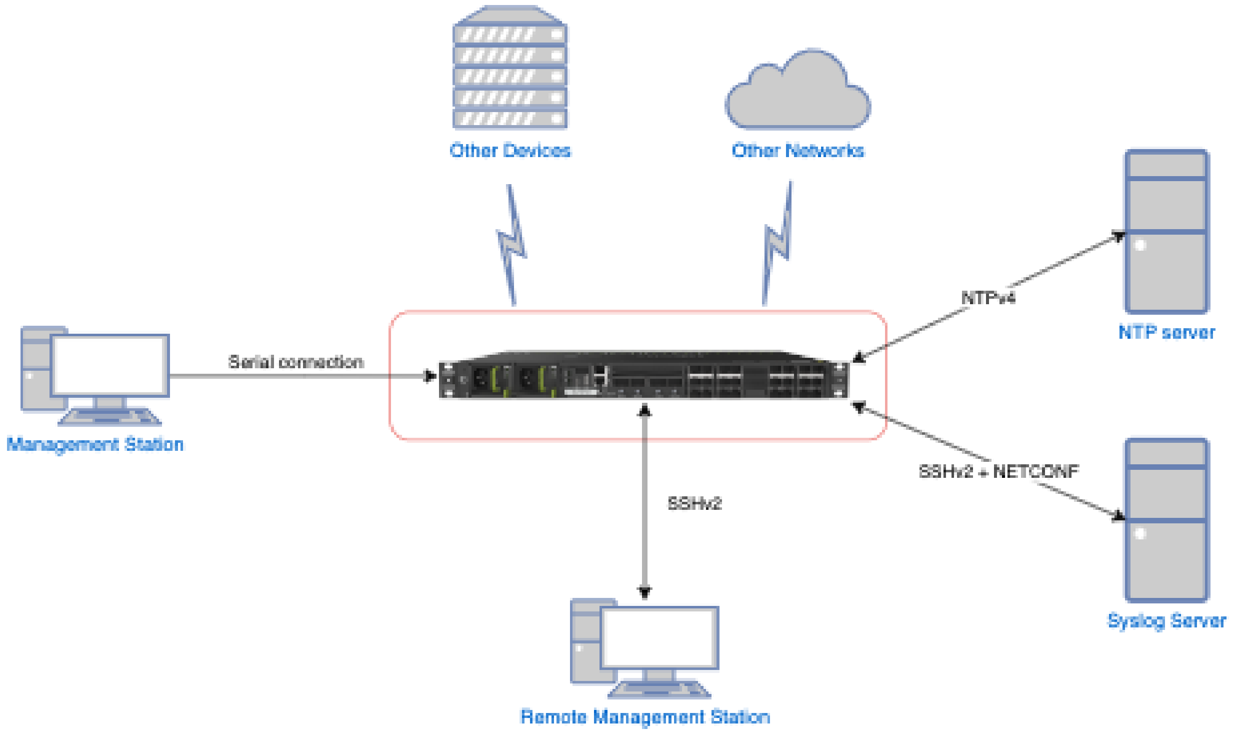
In Junos® OS Evolved, the state is held in a database called the Distributed Data Store (DDS). The DDS does not interpret state. It only holds the state received from subscribers and propagates it to consumers. It implements the publish-subscribe messaging pattern for communicating state between applications that are originators of a state to applications that are consumers of that state. Each application publishes state to and subscribes to state from the DDS directly, making applications independent of each other.

The TOE software is represented by the Junos® OS Evolved Operating System. As illustrated in Figure 3, it implements the routing, filtering, management, and platform functions. The TOE software is provided as an ISO image, as well as necessary TOE updates.

1.5.1 TOE Physical boundary

The physical boundary of the TOE is ACX7024 or ACX7024X chassis. The figure below shows the boundary of the TOE, enclosed in the red box, with the rest of the elements that comprise the Operational Environment and the interfaces with the TOE.

Figure 4: TOE Physical Boundary



The TOE is connected to management workstations, to one or more NTP servers, and to a syslog server. The management workstations can be local or remote. The TOE is also connected to the networks which it interconnects. Neither the management console, the NTP servers, the syslog server, nor the interconnected networks are part of the TOE.

The TOE implements the following distinct sets of interfaces:

1. The operationally required interfaces. These include the power management and the mechanical interfaces used for the cooling and ventilation of the TOE as well as the LEDs informing the user of the status of the TOE.
2. Network interfaces, which are used for the following purposes:
    - Connecting the TOE to the interconnected networks. They are the interfaces for the ingress and egress network traffic and are physically separate from all other network interfaces. The TOE implements the functionality for the network traffic to traverse through it but does not implement any security functions for processing the data on the network interfaces.
    - Connecting the TOE with the NTP servers. The TOE implements the NTP v4 protocol and authenticates NTP servers.
    - Connecting the TOE with the syslog server. The TOE implements the SSHv2 protocol with NETCONF to establish a secure channel with the syslog server, thus protecting the integrity and confidentiality of the audit data.
3. Management interfaces are used by the administrators to manage the TOE. The management interface is available through dedicated network ports and may be accessed locally from a console or remotely over a SSH connection. The management interface implements the CLI which is the only means of administering the TOE.

1.5.2 Operational Environment of the TOE

The TOE is the entire network appliance. Nevertheless, it does require external IT devices to be properly operated. Specifically, the TOE requires the following items in the network environment:

- An external server, supporting a syslog server and a SSHv2 client for connecting to the TOE via NETCONF, so the TOE can send audit logs.
- Optionally, one or more NTP servers for synchronizing the reliable system clock included with the TOE. If NTP is not used, the system clock can be configured manually by an administrator through the management console.
- A management station with a SSHv2 client for the remote administration of the TOE.
- A management station connected to the TOE through a serial connection for the local administration of the TOE.

1.5.3 TOE Guidance

The TOE guidance consists of the following documents:

- Common Criteria Evaluated Configuration Guide for ACX7024 and ACX7024X Devices ([CCGUIDE]).

The specifications for configuring the TOE in the evaluated configuration are located in the TOE guidance documentation. The consumer must read, understand, and follow the guidance documentation provided as part of the TOE for the evaluated configuration.

1.5.4 TOE Logical boundary

The TOE provides the security functionality required by [CPP_ND_V3.0E]☝ and [PKG_SSH_V1.0]☝. All of the security functionality is implemented in the Junos® OS Evolved Operating System as described in the following sections.

1.5.4.1 Security Audit

The TOE implements an audit function. A rich set of audit data is collected and stored as audit records. Each audit record includes a time stamp stating the exact time at which the audit record was generated. Each audit record also includes sufficient information to allow administrators of the TOE to examine the events and investigate possible security violations and attempts thereof.

Audit records are stored in log files within the TOE. The administrator also configures the TOE to forward the audit records to an external syslog server. The syslog server is not part of the TOE. Forwarding the audit records to a syslog server takes place over a trusted channel protected with the SSHv2 protocol.

1.5.4.2 Cryptographic Support

The TOE implements cryptographic functionality for the following purposes:

- protection of user passwords;
- establishment of trusted channels and trusted paths using the SSHv2 protocol;
- symmetric key authentication for the NTP protocol; and
- digital signature verification for TOE trusted updates.

The TOE includes several cryptographic libraries for providing this functionality:

- The Junos® OS Evolved Kernel Cryptographic Module provides a Deterministic Random Bit Generation (DRBG), compliant with SP800-90A, for the creation of random data and cryptographic keys; and hashing algorithms for the protection of user's passwords.

- The Junos® OS Evolved OpenSSL Cryptographic Module, based on the open source OpenSSL library version 3.0.16, provides the rest of the cryptographic algorithms.

The TOE also includes a physical, SP800-90B compliant Entropy Source implemented in the TOE hardware for seeding the DRBG with full entropy. In the evaluated configuration, the DRBG is only seeded by the entropy source claimed in FCS_RBG_EXT.1.

All cryptographic algorithms implemented in the Junos® OS Evolved OpenSSL Cryptographic Module and the Junos® OS Evolved Kernel Cryptographic Module are validated by the Cryptographic Algorithm Validation Program (CAVP). This fulfills the requirements of NIAP Policy Letter #5 [CCEVS-PL05]☝. In addition, the SP800-90B compliant Entropy Source is validated by the Entropy Source Validation (ESV).

1.5.4.3 Identification and Authentication

The TOE ensures that access to the administrative functions is only granted to successfully identified and authenticated users. Illegitimate users are deterred and prevented from gaining access.

The TOE implements password-based authentication to local and remote users. Remote authentication, which is implemented over a trusted path using SSHv2, can be also performed using public-key authentication.

The external syslog server establishes an SSHv2 session under NETCONF with the TOE so the TOE can send audit records. The TOE identifies and authenticates the external server using SSHv2 public-key authentication.

1.5.4.4 Security Management

Authorized administrators may use a Command Line Interface (CLI) for performing a wide range of security management tasks on the TOE. The CLI may be accessed locally from the console or remotely over a SSH connection. There are no alternative methods of administering the TOE.

1.5.4.5 Protection of the TSF

The TOE implements a set of security measures for protecting the functions it implements and the corresponding configuration parameters. The TOE implements integrity tests of the TOE and cryptographic algorithm self-tests at start-up, and takes protective measures if the tests indicate that the TOE software has been tampered or there is a failure in the self-tests.

The TOE protects passwords by hashing their values and not allowing direct access to where they are stored. The TOE also protects cryptographic keys by enforcing access control to the key containers.

TOE access is restricted to authorized administrators and all administrator access goes through a CLI. Administrators have no root access to the underlying Linux operating system.

The TOE also allows upgrading the software in case of vulnerabilities being discovered in the implementation. The integrity of the TOE software is ensured by using a digital signature that is verified before the TOE update.

Finally, the TOE maintains a system clock that is used for generating time stamps used in the enforcement of security functions.

1.5.4.6 TOE Access

The TOE allows the display of a banner before and after a user logs in. The TOE also controls idle remote sessions and terminates the session after a period of time.

1.5.4.7 Trusted Path and Trusted Channels

The TOE implements a secure channel for administrators to manage the TOE remotely. Administrators can connect to the TOE from a remote management station using the SSHv2 protocol. Once successfully identified and authenticated, the administrator has access to the Command Line Interface (CLI).

The TOE also establishes a secure channel using SSHv2 for sending audit records to an external syslog server.

The TOE includes the OpenSSH library version 9.8p1 to implement the SSHv2 protocol. The TOE allows both password-based and public-key-based authentication. The underlying cryptographic algorithms needed for the protocol are provided by the Junos® OS Evolved OpenSSL Cryptographic Module

1.5.5 Excluded TOE Features

The following protocols and services must not be used in association with the TOE:

- Telnet shall not be used. It is not considered secure and violates the trusted path and trusted channel requirements.
- FTP shall not be used. It is not considered secure and violates the trusted path and trusted channel requirements.
- SNMP shall not be used. It is not considered secure and violates the trusted path and trusted channel requirements.
- The TLS protocol is not included in the certification and shall not be used.
- Management of the TOE can only be performed via the Command Line Interface (CLI); other methods like JUNOScript shall not be used.
- Administrators of the TOE shall be created using the super-user class. The Linux root account shall not be used in the evaluated configuration; it can only be used for the initial configuration of the TOE.
- 3rd party applications and tools allowed by the Junos® OS Evolved architecture must not be used.
- The SSHv2 protocol cannot be initiated in the TOE (i.e. using the `ssh` command). The SSHv2 protocol is only allowed as a trusted channel and a trusted path when the communication is initiated at the other endpoint, so the TOE acts as an SSH server.

2 CC Conformance Claim

This Security Target (ST) is CC Part 2 extended and CC Part 3 conformant. Common Criteria [CC] version 3.1 revision 5 is the basis for this conformance claim.

This ST claims exact conformance to the following Protection Profile (PP) and Functional Package:

- [CPP_ND_V3.0E]☝: Collaborative Protection Profile for Network Devices. Version 3.0e as of 2023-12-06 .
- [PKG_SSH_V1.0]☝: Functional Package for SSH. Version 1.0 as of 2021-05-13.

Table 2 below contains the NIAP Technical Decisions (TDs) for the [CPP_ND_V3.0E]☝ protection profile, whereas Table 3 contains the NIAP Technical Decisions (TDs) for the [PKG_SSH_V1.0]☝ functional package. Technical Decisions are those published at the time of the evaluation, and contains a statement of applicability to the evaluation.

Technical Decisions are marked as applicable if any of the documentation or evaluation activities included in this evaluation were updated, even if these evaluation activities were not performed. The applicability of a TD does not imply that the related security functionality is claimed or not claimed by the TOE.

Table 2: NIAP Technical Decisions for [CPP_ND_V3.0E]

TD #	Description	Applicable?	Non-Applicability Rationale
TD0923	NIT Technical Decision: Auditable event for FAU_STG_EXT.1 in FAU_GEN.1.2	Yes	
TD0921	NIT Technical Decision: Addition of FIPS PUB 186-5 and Correction of Assignment	Yes	
TD0900	NIT Technical Decision: Clarification to Local Administrator Access in FIA_UIA_EXT.1.3	Yes	
TD0899	NIT Technical Decision: Correction of Renegotiation Test for TLS 1.2	No	The TLS protocol is not claimed.
TD0886	Clarification to FAU_STG_EXT.1 Test 6	Yes	
TD0880	NIT Decision: Removal of Duplicate Selection in FMT_SMF.1.1	Yes	
TD0879	NIT Decision: Correction of Chapter Headings in CPP_ND_V3.0E	Yes	
TD0868	NIT Technical Decision: Clarification of time frames in FCS_IPSEC_EXT.1.7 and FCS_IPSEC_EXT.1.8	No	The IPsec protocol is not claimed in this ST.
TD0836	NIT Technical Decision: Redundant Requirements in FPT_TST_EXT.1	Yes	

Table 3: NIAP Technical Decisions for [PKG_SSH_V1.0]

TD #	Description	Applicable?	Non-Applicability Rationale
TD0909	Updates to FCS_SSH_EXT.1.1 App Note in SSH FP 1.0	Yes	
TD0777	Clarification to Selections for Auditable Events for FCS_SSH_EXT.1	Yes	
TD0732	FCS_SSHS_EXT.1.3 Test 2 Update	Yes	

TD #	Description	Applicable?	Non-Applicability Rationale
TD0695	Choice of 128 or 256 bit size in AES-CTR in SSH Functional Package	Yes	
TD0682	Addressing Ambiguity in FCS_SSHS_EXT.1 Tests	Yes	

3 Security Problem Definition

3.1 Threat Environment

3.1.1 Threats countered by the TOE

3.1.1.1 Communications with the Network Device

T.UNAUTHORIZED_ADMINISTRATOR_ACCESS

Threat agents may attempt to gain Administrator access to the Network Device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between Network Devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.

T.WEAK_CRYPTOGRAPHY

Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.

T.UNTRUSTED_COMMUNICATION_CHANNELS

Threat agents may attempt to target Network Devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the Network Device itself.

T.WEAK_AUTHENTICATION_ENDPOINTS

Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints, e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the Network Device itself could be compromised.

3.1.1.2 Valid Updates

T.UPDATE_COMPROMISE

Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.

3.1.1.3 Audited Activity

T.UNDETECTED_ACTIVITY

Threat agents may attempt to access, change, and/or modify the security functionality of the Network Device without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the Administrator would have no knowledge that the device has been compromised.

3.1.1.4 Administrator and Device Credentials and Data

T.SECURITY_FUNCTIONALITY_COMPROMISE

Threat agents may compromise credentials and device data enabling continued access to the Network Device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker. Threat agents may also be able to take advantage of weak administrative passwords to gain privileged access to the device.

3.1.1.5 Device Failure

T.SECURITY_FUNCTIONALITY_FAILURE

An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.

3.2 Assumptions

3.2.1 Intended usage of the TOE

A.PHYSICAL_PROTECTION

The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device. For vNDs, this assumption applies to the physical platform on which the VM runs.

A.LIMITED_FUNCTIONALITY

The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality).

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.

A.NO_THRU_TRAFFIC_PROTECTION

A standard/generic Network Device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the Network Device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the Network Device, destined for another network entity, is not covered by the ND cPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall).

A.TRUSTED_ADMINISTRATOR

The Security Administrator(s) for the Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device.

For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE’s trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification).

A.REGULAR_UPDATES

The Network Device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

A.ADMIN_CREDENTIALS_SECURE

The Administrator’s credentials (private key) used to access the Network Device are protected by the platform on which they reside.

A.RESIDUAL_INFORMATION

The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

3.3 Organizational Security Policies

P.ACCESS_BANNER

The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which Administrators consent by accessing the TOE.

4 Security Objectives

4.1 Objectives for the TOE

This ST does not define security objectives for the TOE.

4.2 Objectives for the Operational Environment

OE.PHYSICAL

Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

OE.NO_GENERAL_PURPOSE

There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. Note: For vNDs the TOE includes only the contents of the its own VM, and does not include other VMs or the VS.

OE.NO_THRU_TRAFFIC_PROTECTION

The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.

OE.TRUSTED_ADMIN

Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality.

For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE’s trust store in case such certificate can no longer be trusted.

OE.UPDATES

The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

OE.ADMIN_CREDENTIALS_SECURE

The Administrator’s credentials (private key) used to access the TOE must be protected on any other platform on which they reside.

OE.RESIDUAL_INFORMATION

The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. For vNDs, this applies when the physical platform on which the VM runs is removed from its operational environment.

4.3 Security Objectives Rationale

The security objectives rationale is defined in the [CPP_ND_V3.0E]☝ protection profile.

5 Extended Components Definition

This Security Target claims exact conformance to [CPP_ND_V3.0E]☝ and [PKG_SSH_V1.0]☝; therefore, it does not extend the security requirements defined by these documents.

6 Security Requirements

6.1 TOE Security Functional Requirements

The table below summarizes the SFRs for the TOE and the operations performed on the components according to CC part 1. Operations in the SFRs use the following convention:

- Iterations (Iter.) are identified by appending a suffix to the original SFR.
- Refinements (Ref.) added to the text are shown in *italic text*, deletions are shown as ~~strikethrough text~~.
- Assignments (Ass.) are shown in **bold text**.
- Selections (Sel.) are shown in **bold text**.

Table 4: SFRs for the TOE

Security Functional Class	Security Functional Requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
FAU - Security audit	FAU_GEN.1 Audit Data Generation	CPP_ND	No	No	No	Yes
	FAU_GEN.2 User Identity Association	CPP_ND	No	No	No	No
	FAU_STG_EXT.1 Protected Audit Event Storage	CPP_ND	No	No	Yes	Yes
FCS - Cryptographic support	FCS_CKM.1 Cryptographic Key Generation	CPP_ND	No	No	Yes	Yes
	FCS_CKM.2 Cryptographic Key Establishment	CPP_ND	No	No	No	Yes
	FCS_CKM.4 Cryptographic Key Destruction	CPP_ND	No	No	No	Yes
	FCS_COP.1/DataEncryption Cryptographic Operation (AES Data Encryption/Decryption)	CPP_ND	No	No	No	Yes
	FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)	CPP_ND	No	No	Yes	Yes
	FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)	CPP_ND	No	No	No	Yes
	FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)	CPP_ND	No	No	Yes	Yes
	FCS_NTP_EXT.1 NTP Protocol	CPP_ND	No	No	No	Yes
	FCS_RBG_EXT.1 Random Bit Generation	CPP_ND	No	No	Yes	Yes
	FCS_SSH_EXT.1 SSH Protocol	PKG_SSH	No	No	Yes	Yes
	FCS_SSHS_EXT.1 SSH Protocol - Server	PKG_SSH	No	No	No	Yes
FIA - Identification and authentication	FIA_AFL.1 Authentication Failure Management	CPP_ND	No	No	Yes	Yes
	FIA_PMG_EXT.1 Password Management	CPP_ND	No	No	Yes	Yes
	FIA_UAU.7 Protected Authentication Feedback	CPP_ND	No	No	No	No
	FIA_UIA_EXT.1 User Identification and Authentication	CPP_ND	No	No	No	Yes
FMT - Security management	FMT_MOF.1/Functions Management of Security Functions Behaviour	CPP_ND	No	No	No	Yes

Security Functional Class	Security Functional Requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
	FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour	CPP_ND	No	No	No	No
	FMT_MOF.1/Services Management of Security Functions Behaviour	CPP_ND	No	No	No	No
	FMT_MTD.1/CoreData Management of TSF Data	CPP_ND	No	No	No	No
	FMT_MTD.1/CryptoKeys Management of TSF Data	CPP_ND	No	No	No	No
	FMT_SMF.1 Specification of Management Functions	CPP_ND	No	No	No	Yes
	FMT_SMR.2 Restrictions on Security Roles	CPP_ND	No	No	No	No
FPT - Protection of the TSF	FPT_APW_EXT.1 Protection of Administrator Passwords	CPP_ND	No	No	No	No
	FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys)	CPP_ND	No	No	No	No
	FPT_STM_EXT.1 Reliable Time Stamps	CPP_ND	No	No	No	Yes
	FPT_TST_EXT.1 TSF testing	CPP_ND	No	No	Yes	Yes
	FPT_TUD_EXT.1 Trusted Update	CPP_ND	No	No	No	Yes
FTA - TOE access	FTA_SSL.3 TSF-initiated Termination	CPP_ND	No	No	No	No
	FTA_SSL.4 User-initiated Termination	CPP_ND	No	No	No	No
	FTA_SSL_EXT.1 TSF-initiated Session Locking	CPP_ND	No	No	No	Yes
	FTA_TAB.1 Default TOE Access Banners	CPP_ND	No	No	No	No
FTP - Trusted path/channels	FTP_ITC.1 Inter-TSF trusted channel	CPP_ND	No	No	Yes	Yes
	FTP_TRP.1/Admin Trusted Path	CPP_ND	No	No	No	Yes

6.1.1 Security audit (FAU)

6.1.1.1 FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1

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

a. Start-up and shut-down of the audit functions;

b. All auditable events for the not specified level of audit; and

c. All administrative actions comprising:

- Administrative login and logout (name of Administrator account shall be logged if individual 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 Administrator account shall be logged)**;

d. Specifically defined auditable events listed in Table 5.

FAU_GEN.1.2

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

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

b. For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, information specified in column three of Table 5.

Table 5: Security Functional Requirements and Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	Configuration of local audit settings.	Identity of account making changes to the audit configuration.
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_NTP_EXT.1	• Configuration of a new time server. • Removal of configured time server.	Identity if new/removed time server.
FCS_RBG_EXT.1	None.	None.
FCS_SSH_EXT.1	**Failure to establish SSH connection**.	**Reason for failure and Non-TOE endpoint of attempted connection (IP Address)**.
	Establishment of SSH connection.	**Non-TOE endpoint of connection (IP Address)**.
	Termination of SSH connection session.	**Non-TOE endpoint of connection (IP Address)**.
	Dropping of packet(s) outside defined size limits.	**Packet size**.
FCS_SSHS_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.

Requirement	Auditable Events	Additional Audit Record Contents
FIA_UAU.7	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanisms.	Origin of the attempt (e.g., IP address).
FMT_MOF.1/Functions	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update.	None.
FMT_MOF.1/Services	None.	None.
FMT_MTD.1/CoreData	None.	None.
FMT_MTD.1/CryptoKeys	None.	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMR.2	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_SKP_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.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_SSL_EXT.1	The termination of a local session by the session lock.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	Initiation of the trusted channel.	None.
	Termination of the trusted channel.	None.
	Failure of the trusted channel functions.	Reason for failure.
FTP_TRP.1/Admin	Initiation of the trusted path.	None.
	Termination of the trusted path.	None.
	Failure of the trusted path functions.	Reason for failure.

Applied TDs: *TD0777*

TSS Link: *TSS for FAU_GEN.1*

6.1.1.2 FAU_GEN.2 User Identity Association

FAU_GEN.2.1

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

TSS Link: *TSS for FAU_GEN.2*

6.1.1.3 FAU_STG_EXT.1 Protected Audit Event Storage

FAU_STG_EXT.1.1

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

FAU_STG_EXT.1.2

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

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

.

FAU_STG_EXT.1.3

The TSF shall maintain a **log file**, **a set of circular archived log files** of audit records in the event that an interruption of communication with the remote audit server occurs.

FAU_STG_EXT.1.4

The TSF shall be able to store **persistent** audit records locally with a minimum storage size of **64 KB**.

FAU_STG_EXT.1.5

The TSF shall **overwrite previous audit records according to the following rule: the oldest archived log file is overwritten** when the local storage space for audit data is full.

FAU_STG_EXT.1.6

The TSF shall provide the following mechanisms for administrative access to locally stored audit records **ability to view locally**.

TSS Link: *TSS for FAU_STG_EXT.1*

6.1.2 Cryptographic support (FCS)

6.1.2.1 FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.1.1

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

- **RSA schemes using cryptographic key sizes of 2048, 3072, 4096, 6144 and 8192 bits that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", A.1;**
- **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, or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2, or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.;**

.

Applied TDs: *TD0921*

TSS Link: *TSS for FCS_CKM.1*

6.1.2.2 FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1

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

- **Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";**

.

TSS Link: *TSS for FCS_CKM.2*

6.1.2.3 FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4.1

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

- For plaintext keys in volatile storage, the destruction shall be executed by a **single overwrite consisting of zeroes**, **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 ;**
  - **instructs a part of the TSF to destroy the abstraction that represents the key**

that meets the following: No Standard.

TSS Link: *TSS for FCS_CKM.4*

6.1.2.4 FCS_COP.1/DataEncryption Cryptographic Operation (AES Data Encryption/Decryption)

FCS_COP.1.1/DataEncryption

Copyright © 2025 Juniper Networks, Inc.

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

TSS Link: *TSS for FCS_COP.1/DataEncryption*

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

FCS_COP.1.1/SigGen

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

- **RSA Digital Signature Algorithm**
- **Elliptic Curve Digital Signature Algorithm**

and cryptographic key sizes

- **For RSA: modulus 2048, 3072 and 4096 bits**
- **For ECDSA: 256, 384 and 521 bits**

that meet the following:

- **For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4 using PKCS #1 v2.2 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 implementing P-256, P-384, P-521 curves that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST Recommended" curves; or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 6 and NIST SP 800-186 Section 3.2.1, Implementing Weierstrass curves; or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.**

Applied TDs: *TD0921*

TSS Link: *TSS for FCS_COP.1/SigGen*

6.1.2.6 FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

FCS_COP.1.1/Hash

The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm **SHA-1**, **SHA-256**, **SHA-384**, **SHA-512** and message digest sizes **160**, **256**, **384**, **512** bits that meet the following: ISO/IEC 10118-3:2004.

TSS Link: *TSS for FCS_COP.1/Hash*

6.1.2.7 FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)

FCS_COP.1.1/KeyedHash

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

TSS Link: *TSS for FCS_COP.1/KeyedHash*

6.1.2.8 FCS_NTP_EXT.1 NTP Protocol

FCS_NTP_EXT.1.1

The TSF shall use only the following NTP version(s) **NTP v4 (** [**RFC5905**]☝ **)**.

FCS_NTP_EXT.1.2

The TSF shall update its system time using **Authentication using**

- **SHA1**
- **SHA256**

as the message digest algorithm(s);

FCS_NTP_EXT.1.3

The TSF shall not update NTP timestamp from broadcast and/or multicast addresses.

FCS_NTP_EXT.1.4

The TSF shall support configuration of at least three (3) NTP time sources in the Operational Environment.

TSS Link: *TSS for FCS_NTP_EXT.1*

6.1.2.9 FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1.1

The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using **HMAC_DRBG SHA-512**.

FCS_RBG_EXT.1.2

The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from **one 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.

TSS Link: *TSS for FCS_RBG_EXT.1*

6.1.2.10 FCS_SSH_EXT.1 SSH Protocol

FCS_SSH_EXT.1.1

Copyright © 2025 Juniper Networks, Inc.

The TOE shall implement SSH acting as a **server** in accordance with that complies with RFCs 4251, 4252, 4253, 4254, **4256**, **4344**, **5656**, **6668**, **8308**, **8332** and [no other standard].

FCS_SSH_EXT.1.2

The TSF shall ensure that the SSH protocol implementation supports the following authentication methods:

- **"password" (RFC 4252)**
- **"keyboard-interactive" (RFC 4256)**
- **"publickey" (RFC 4252):**
  - **rsa-sha2-256 (RFC 8332)**
  - **rsa-sha2-512 (RFC 8332)**
  - **ecdsa-sha2-nistp256 (RFC 5656)**
  - **ecdsa-sha2-nistp384 (RFC 5656)**
  - **ecdsa-sha2-nistp521 (RFC 5656)**

and no other methods.

FCS_SSH_EXT.1.3

The TSF shall ensure that, as described in RFC 4253, packets greater than **262144 bytes** in an SSH transport connection are dropped.

FCS_SSH_EXT.1.4

The TSF shall protect data in transit from unauthorised disclosure using the following mechanisms:

- **aes128-ctr (RFC 4344)**
- **aes256-ctr (RFC 4344)**

and no other mechanisms.

FCS_SSH_EXT.1.5

The TSF shall protect data in transit from modification, deletion, and insertion using:

- **hmac-sha2-256 (RFC 6668)**
- **hmac-sha2-512 (RFC 6668)**

and no other mechanisms.

FCS_SSH_EXT.1.6

The TSF shall establish a shared secret with its peer using:

- **ecdh-sha2-nistp256 (RFC 5656)**
- **ecdh-sha2-nistp384 (RFC 5656)**
- **ecdh-sha2-nistp521 (RFC 5656)**

and no other mechanisms.

FCS_SSH_EXT.1.7

The TSF shall use SSH KDF as defined in

Copyright © 2025 Juniper Networks, Inc.

- **RFC 5656 (Section 4)**

to derive the following cryptographic keys from a shared secret: session keys.

FCS_SSH_EXT.1.8

The TSF shall ensure that

- **a rekey of the session keys**

occurs when any of the following thresholds are met:

- one hour connection time
- no more than one gigabyte of transmitted data, or
- no more than one gigabyte of received data.

TSS Link: *TSS for FCS_SSH_EXT.1*

6.1.2.11 FCS_SSHS_EXT.1 SSH Protocol - Server

FCS_SSHS_EXT.1.1

The TSF shall authenticate itself to its peer (SSH Client) using:

- **ssh-rsa (RFC 4253)**
- **rsa-sha2-256 (RFC 8332)**
- **rsa-sha2-512 (RFC 8332)**
- **ecdsa-sha2-nistp256 (RFC 5656)**
- **ecdsa-sha2-nistp384 (RFC 5656)**
- **ecdsa-sha2-nistp521 (RFC 5656)**

.

TSS Link: *TSS for FCS_SSHS_EXT.1*

6.1.3 Identification and authentication (FIA)

6.1.3.1 FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1

The TSF shall detect when an Administrator configurable positive integer within **1 to 10** unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password.

FIA_AFL.1.2

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

TSS Link: *TSS for FIA_AFL.1*

6.1.3.2 FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1.1

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: **"!"**, **"@"**, **"#"**, **"$"**, **"%"**, **"^"**, **"&"**, **"*"**, **"("**, **")"**, **all other standard ASCII, extended ASCII and Unicode characters**;

b) Minimum password length shall be configurable to between **10** and **20** characters.

TSS Link: *TSS for FIA_PMG_EXT.1*

6.1.3.3 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.

TSS Link: *TSS for FIA_UAU.7*

6.1.3.4 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:

a) Display the warning banner in accordance with FTA_TAB.1;

b) **no other actions**.

FIA_UIA_EXT.1.2

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

FIA_UIA_EXT.1.3

The TSF shall provide the following remote authentication mechanisms **SSH password**, **SSH public key** and **no other mechanism**. The TSF shall provide the following local authentication mechanisms **password-based**.

FIA_UIA_EXT.1.4

The TSF shall authenticate any administrative user's claimed identity according to each authentication mechanism specified in FIA_UIA_EXT.1.3.

Applied TDs: *TD0900*

TSS Link: *TSS for FIA_UIA_EXT.1*

6.1.4 Security management (FMT)

6.1.4.1 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.

TSS Link: *TSS for FMT_MOF.1/Functions*

6.1.4.2 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.

TSS Link: *TSS for FMT_MOF.1/ManualUpdate*

6.1.4.3 FMT_MOF.1/Services Management of Security Functions Behaviour

FMT_MOF.1.1/Services

The TSF shall restrict the ability to start and stop services to Security Administrators.

TSS Link: *TSS for FMT_MOF.1/Services*

6.1.4.4 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.

TSS Link: *TSS for FMT_MTD.1/CoreData*

6.1.4.5 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.

TSS Link: *TSS for FMT_MTD.1/CryptoKeys*

6.1.4.6 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 remotely;
- Ability to configure the access banner;
- Ability to configure the remote session inactivity time before session termination;
- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
- **Ability to start and stop services;**
- **Ability to configure local audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full; changes to local audit storage size)**
- **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 configure NTP;**
- **Ability to administer the TOE locally;**
- **Ability to configure the local session inactivity time before session termination or locking;**
- **Ability to configure the authentication failure parameters for FIA_AFL.1;**
- **Ability to manage the trusted public keys database;**

.

Applied TDs: *TD0880*

TSS Link: *TSS for FMT_SMF.1*

6.1.4.7 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 remotely

are satisfied.

TSS Link: *TSS for FMT_SMR.2*

Copyright © 2025 Juniper Networks, Inc.

6.1.5 Protection of the TSF (FPT)

6.1.5.1 FPT_APW_EXT.1 Protection of Administrator Passwords

FPT_APW_EXT.1.1

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

FPT_APW_EXT.1.2

The TSF shall prevent the reading of plaintext administrative passwords.

TSS Link: *TSS for FPT_APW_EXT.1*

6.1.5.2 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys)

FPT_SKP_EXT.1.1

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

TSS Link: *TSS for FPT_SKP_EXT.1*

6.1.5.3 FPT_STM_EXT.1 Reliable Time Stamps

FPT_STM_EXT.1.1

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

FPT_STM_EXT.1.2

The TSF shall **allow the Security Administrator to set the time**, **synchronise time with an NTP server**.

TSS Link: *TSS for FPT_STM_EXT.1*

6.1.5.4 FPT_TST_EXT.1 TSF testing

FPT_TST_EXT.1.1

The TSF shall run a suite of the following self-tests

a) During initial start-up (on power on) to verify the integrity of the TOE firmware and software;

b) Prior to providing any cryptographic service and **at no other time** to verify correct operation of cryptographic implementation necessary to fulfil the TSF;

c) **no other** self-tests **none**;

to demonstrate the correct operation of the TSF.

FPT_TST_EXT.1.2

The TSF shall respond to **all failures** by **rebooting**, **restarting services that use cryptographic operations provided by Junos® OS Evolved OpenSSL Cryptographic Module**.

Applied TDs: *TD0836*

TSS Link: *TSS for FPT_TST_EXT.1*

6.1.5.5 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1

The TSF shall provide Security Administrators the ability to query the currently executing version of the TOE firmware/software and **the most recently installed version of the TOE firmware/software**.

FPT_TUD_EXT.1.2

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

FPT_TUD_EXT.1.3

The TSF shall provide means to authenticate firmware/software updates to the TOE using a **digital signature** prior to installing those updates.

TSS Link: *TSS for FPT_TUD_EXT.1*

6.1.6 TOE access (FTA)

6.1.6.1 FTA_SSL.3 TSF-initiated Termination

FTA_SSL.3.1

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

TSS Link: *TSS for FTA_SSL.3*

6.1.6.2 FTA_SSL.4 User-initiated Termination

FTA_SSL.4.1

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

TSS Link: *TSS for FTA_SSL.4*

6.1.6.3 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1

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

TSS Link: *TSS for FTA_SSL_EXT.1*

6.1.6.4 FTA_TAB.1 Default TOE Access Banners

Copyright © 2025 Juniper Networks, Inc.

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.

TSS Link: *TSS for FTA_TAB.1*

6.1.7 Trusted path/channels (FTP)

6.1.7.1 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 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 service**.

TSS Link: *TSS for FTP_ITC.1*

6.1.7.2 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.

TSS Link: *TSS for FTP_TRP.1/Admin*

Copyright © 2025 Juniper Networks, Inc.

6.2 Security Functional Requirements Rationale

The SFR rationale is defined in the [CPP_ND_V3.0E]☝ protection profile.

6.3 Security Assurance Requirements

The security assurance requirements (SARs) for the TOE are defined in the CPP_ND protection profile.

The following table shows the SARs, and the operations performed on the components according to CC part 3: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Table 6: SARs

Security Assurance Class	Security Assurance Requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
ADV Development	ADV_FSP.1 Basic functional specification	CPP_ND	No	No	No	No
AGD Guidance documents	AGD_OPE.1 Operational user guidance	CPP_ND	No	No	No	No
	AGD_PRE.1 Preparative procedures	CPP_ND	No	No	No	No
ALC Life-cycle support	ALC_CMC.1 Labelling of the TOE	CPP_ND	No	No	No	No
	ALC_CMS.1 TOE CM coverage	CPP_ND	No	No	No	No
	ALC_FLR.3 Systematic flaw remediation	CPP_ND	No	No	No	No
ATE Tests	ATE_IND.1 Independent testing - conformance	CPP_ND	No	No	No	No
AVA Vulnerability assessment	AVA_VAN.1 Vulnerability survey	CPP_ND	No	No	No	No
ASE Security Target evaluation	ASE_CCL.1 Conformance claims	CPP_ND	No	No	No	No
	ASE_ECD.1 Extended components definition	CPP_ND	No	No	No	No
	ASE_INT.1 ST introduction	CPP_ND	No	No	No	No
	ASE_OBJ.1 Security objectives for the operational environment	CPP_ND	No	No	No	No
	ASE_REQ.1 Stated security requirements	CPP_ND	No	No	No	No
	ASE_SPD.1 Security problem definition	CPP_ND	No	No	No	No
	ASE_TSS.1 TOE summary specification	CPP_ND	No	No	No	No

6.4 Security Assurance Requirements Rationale

The SAR rationale is defined in the [CPP_ND_V3.0E]☝ protection profile.

Copyright © 2025 Juniper Networks, Inc.

7 TOE Summary Specification

7.1 TOE Security Functionality

As per [CPP_ND_V3.0E]☝ and [PKG_SSH_V1.0]☝, the TOE supports the following major security features:

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

7.1.1 Audit Generation

The TOE generates and stores audit records for several events. The audit functionality is implemented using syslog. The detail of the events to be recorded by syslog are determined by the logging level specified through the `set system syslog` CLI command.

The TOE includes the following information with each audit record:

- date and time of the event and/or reaction;
- type of event and/or reaction;
- subject identity (where applicable); and
- the outcome (success or failure) of the event (if applicable).

The subject identity included in the audit records is the username of the human user of the TOE or the IP Address of the peer entity attempting to connect to the TOE.

7.1.1.1 FAU_GEN.1

The list of audit events is specified in Table 5; audit events are grouped by SFR with the corresponding information contained in the audit record.

SSH keys generated are identified in the audit record by the public key filename and fingerprint.

SSH keys imported are identified in the audit record by the hash of the key imported and the username of the user importing the key. The key is bound to the username.

SSH keys used for trusted channels and paths are not deleted by the management daemon when SSH is de-configured. SSH keys used for trusted channels are deleted as part of a factory configuration reset by issuing the `request system zeroize` command. As this command performs the zeroization of the entire appliance, it is not possible to record the event.

7.1.1.2 FAU_GEN.2

The TSS requirements are already covered by the TSS requirements for FAU_GEN.1.

7.1.1.3 FAU_STG_EXT.1

The TOE is a non-distributed TOE that stores persistent audit records locally, and can be configured to send audit records to one or more external audit servers. In this case, audit events are sent in real time via NETCONF over SSH.

Local audit logging consists of an active audit log file and a set of circular archive files, all stored under the `/var/log/` directory in the filesystem of the TOE. The TOE writes audit records in the log file until the maximum size for the log file is reached. The log file is then compressed and moved to the set of circular archive files, removing the oldest archive log file in case the set is full.

For example, assuming that audit is configured to have a maximum size of 1 GB. in the active log and a set of 10 circular archived audit logs, when the active log file (e.g. `logfile`) surpasses the maximum size, the TOE first renames each existing filename in the archive set from `filename.(i).gz` to `filename.(i+1).gz` so the active log can be closed, compressed and renamed as the latest archived file (e.g. `logfile.0.gz`). The oldest file in the circular set is removed in case the set is full (e.g. `logfile.9.gz`). The TOE then creates an empty, new active log file (e.g. `logfile`).

The audit log filename, the maximum size of the active audit log, and the maximum number of files that comprise the set of archived files can be configured by the Security Administrator using the CLI. The following table shows the minimum, default and maximum values supported by the TOE.

Table 7: Local Audit Configuration

Configuration	Number of Circular Files	Size per File	Total Storage
Minimum	1	64 KB	64 KB
Default	10	1 MB	10 MB
Maximum	1000	1 GB	1 TB

The TOE protects audit records from unauthorized modification and deletion by using the filesystem access control of the underlying operating system. Only an Administrator can read, delete, or archive log files. Managing the log files is through the CLI interface or through direct access to the filesystem. Local audit logs are accessible via the `show log` CLI command.

A 1 GB syslog file requires approximately 0.25 GB of storage when archived. The total size of files in Syslog may reach the complete storage capacity allocated to the `/var` filesystem. The complete storage capacity is platform specific. When the filesystem size reaches 92% of the storage capacity, an event is generated but the event daemon 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, the `/var` filesystem becomes exhausted. In that case, a final log entry “No space left on device” is generated and the logging is terminated. Other functions of the TOE shall continue when the audit log storage space is exhausted.

7.1.2 Cryptographic Support

The TOE uses the following cryptographic libraries:

- Junos® OS Evolved OpenSSL Cryptographic Module, included as part of the TOE software;
- Junos® OS Evolved Kernel Cryptographic Module, included as part of the TOE software.

The Junos® OS Evolved OpenSSL Cryptographic Module is based on the open source OpenSSL library, version 3.0.16. The table below shows the cryptographic services used by the TOE, describing the algorithms, their supported key sizes, applicable standard and purpose. The table also includes the certificates obtained from the Cryptographic Algorithm Validation Program (CAVP) in the evaluated configuration for each of the cryptographic algorithms.

Table 8: Cryptographic Algorithms Implemented by the Junos® OS Evolved OpenSSL Cryptographic Module

Cryptographic Service	Algorithm	Key Sizes	Standard	Purpose	CAVP Cert.
FCS_CKM.1 - Cryptographic Key Generation	RSA	2048, 3072, 4096, 6144 and 8192 bits	[FIPS186-5]	Key generation of the SSH host (TOE) key pair (used for TOE authentication by the SSH peer).	A7389
	Elliptic Curve Cryptography (ECC)	P-256, P-384, P-521 (256, 384 and 521 bits)	[FIPS186-5]	Key generation of the SSH host (TOE) key pair (used for TOE authentication by the SSH peer). Ephemeral asymmetric key generation for SSHv2 key exchange.	A7389
FCS_CKM.2 - Cryptographic Key Establishment	Elliptic Curve Cryptography (KAS-ECC-SSC)	P-256, P-384, P-521 (256, 384 and 521 bits)	[SP800-56A-Rev3]	SSHv2 key exchange.	A7389
FCS_COP.1/ DataEncryption - Cryptographic Operation (AES Data Encryption/ Decryption)	AES in CTR mode	128, 256 bits	[SP800-38A]	Data encryption and decryption in the SSHv2 protocol.	A7387
FCS_COP.1/SigGen - Cryptographic Operation (Signature Generation and Verification)	RSA with SHA-1	2048, 3072 and 4096 bits	[FIPS186-5]	Server authentication in the SSHv2 protocol.	CCTL tested
	RSA with SHA2-256, SHA2-512			Server authentication in the SSHv2 protocol. Public-key authentication in the SSHv2 protocol. Digital Signature Verification for Trusted Updates of the TOE.	A7389
	ECDSA with SHA2-256, SHA2-384, SHA2-512	P-256, P-384, P-521 (256, 384 and 521 bits)	[FIPS186-5]	Server authentication in the SSHv2 protocol. Public-key authentication in the SSHv2 protocol.	A7389
FCS_COP.1/Hash - Cryptographic Operation (Hash Algorithm)	SHA-1, SHA2-256, SHA2-384, SHA2-512	N/A	[FIPS180-4]	HMAC algorithm. Authentication in NTP protocol (SHA-1, SHA2-256). Digital Signature Generation and Verification. Pseudorandom function (PRF) for the SSHv2 protocol.	A7389

Cryptographic Service	Algorithm	Key Sizes	Standard	Purpose	CAVP Cert.
FCS_COP.1/ KeyedHash - Cryptographic Operation (Keyed Hash Algorithm)	HMAC with SHA2-256, SHA2-512	256, 512 bits	[FIPS198-1]☝	Data Integrity in the SSHv2 protocol. Integrity test of software components when the TOE starts up.	A7389

Table 9: Cryptographic Algorithms Implemented by the Junos® OS Evolved Kernel Cryptographic Module

Cryptographic Service	Algorithm	Key Sizes	Standard	Purpose	CAVP Cert.
FCS_COP.1/Hash - Cryptographic Operation (Hash Algorithm)	SHA2-256	N/A	[FIPS180-4]☝	Protect passwords. Verify TOE integrity via the Linux Integrity Mechanism Architecture (IMA).	A7046
	SHA2-512			Protect passwords.	A6983
FCS_RBG_EXT.1 - Random Bit Generation	DRBG (HMAC_DRBG with SHA2-512)	256 bits	[SP800-90A-Rev1]☝	Asymmetric Key Generation. Client and server random secrets in the SSHv2 protocol.	A6983

7.1.2.1 FCS_CKM.1

See Table 8.

7.1.2.2 FCS_CKM.2

See Table 8.

7.1.2.3 FCS_CKM.4

The Junos® OS Evolved OpenSSL Cryptographic Module and Junos® OS Evolved Kernel Cryptographic Module that are part of the TOE implement functions for the secure destruction of Sensitive Security Parameters (SSP), like cryptographic keys. SSPs stored in volatile memory are first zeroized (set with zeroes using the `memset()` function), and both the memory area used for the SSP and its handler used to reference it are released (using the `free()` function) at the termination of a session. The Linux kernel does not implement garbage collection functionality per se; instead it uses manual memory management techniques like reference counting for shared data structures and has some automated cleanup mechanisms, which act similarly to a garbage collector. The TOE decides when a given memory area is no longer used, and request to the operating system its release so it can be used by other processes.

SSPs stored in non-volatile memory are erased when the administrator decommissions the TOE by executing the `request system zeroize` command. The command first zeroizes the content of the files where SSPs are stored, and then removes the file from the filesystem.

The following table shows the SSPs used by the TOE, their location (including the filename when it's non-volatile memory), storage format and zeroization method.

Table 10: Storage and Zeroization of Sensitive Security Parameters (SSP)

SSP	Storage Location	Storage Format	Zeroization Method
SSH Host Key (RSA with 3072-bit modulus)	/etc/ssh/ssh_host_rsa_key and /etc/ssh/ssh_host_rsa_key.pub	Plaintext	When the TOE is decommissioned, the existing SSH host keys are zeroized and erased.
SSH Host Key (ECDSA with NIST P-256)	/etc/ssh/ssh_host_ecdsa_key and /etc/ssh/ssh_host_ecdsa_key.pub	Plaintext	When the TOE is decommissioned, the existing SSH host keys are zeroized and erased.
SSH Host Key (ECDSA with NIST P-384)	/etc/ssh/ssh_host_ec_p384_key and /etc/ssh/ssh_host_ec_p384_key.pub	Plaintext	When the TOE is decommissioned, the existing SSH host keys are zeroized and erased.
SSH Host Key (ECDSA with NIST P-521)	/etc/ssh/ssh_host_ec_p521_key and /etc/ssh/ssh_host_ec_p521_key.pub	Plaintext	When the TOE is decommissioned, the existing SSH host keys are zeroized and erased.
SSH User (client) Public Key	<user directory>/.ssh/authorized_keys	Plaintext	When the TOE is decommissioned, existing keys are zeroized and erased.
SSH Session Keys	Volatile memory	Plaintext	Zeroized when SSHv2 session terminates.
DRBG Internal State	Volatile memory	Plaintext	Overwritten with zeroes when the TOE is shutdown or reboot.
User Password	/etc/shadow file	Hashed with SHA2-256 or SHA2-512	When the TOE is decommissioned, all user information is erased.
NTP authentication key	/etc/ntp.keys	Plaintext	When the TOE is decommissioned, the existing keys are zeroized and erased.
Trusted Update verification certificate chain	Built into the Junos® OS Evolved Operating System	Plaintext	Not zeroized, but not necessary as it is not sensitive.

7.1.2.4 FCS_COP.1/DataEncryption

See Table 8.

7.1.2.5 FCS_COP.1/SigGen

See Table 8.

7.1.2.6 FCS_COP.1/Hash

See Table 8 and Table 9.

7.1.2.7 FCS_COP.1/KeyedHash

The following table shows the parameter sizes used by the keyed hash algorithms implemented in the TOE.

Table 11: HMAC Parameters

Algorithm	Key Length	Hash Function	Block Size	Output Size
HMAC-SHA2-256	256 bits	SHA2-256	512 bits	256 bits
HMAC-SHA2-512	512 bits	SHA2-512	1024 bits	512 bits

7.1.2.8 FCS_NTP_EXT.1

The TOE allows the synchronization of the system time using the Network Time Protocol (NTP). The TOE supports NTP version 4, compliant with [RFC5905]☝.

NTP is implemented in the Linux Operating System by a daemon (ntpd) running in user space. When the system clock is being synchronized by ntpd, the kernel will in turn update the real time clock (RTC) every 11 minutes automatically.

The TOE implements the symmetric key authentication scheme as defined in [RFC5905]☝ using SHA-1 or SHA2-256 as the hashing algorithm. The encryption key comprises two parts: a key number, which is an integer from 1 to 65534 and a key value. The TOE computes a hash value based on a combination of the key value and the NTP message and appends the key number and the hash value to the outgoing request. The time servers compute a hash code in the same way: the key number in the received message is used to recover a private copy of the key string, which is then combined with the incoming NTP message, and the hash value of the combination is computed. This computed value is compared with the value appended to the message by the sender. If the comparison agrees, and if the source address of the request agrees with the value associated with the key number, the server constructs a reply message, adds the key value and computes the hash of the combination. The computed hash code is appended to the NTP message and the combination is sent back to the TOE.

Distribution of the symmetric key used for NTP authentication must be performed by an Administrator using secure means.

7.1.2.9 FCS_RBG_EXT.1

The TOE uses the Deterministic Random Bit Generator (DRBG) implemented in the Junos® OS Evolved Kernel Cryptographic Module. The DRBG is SP800-90A compliant, and is configured to use HMAC_DRBG with SHA2-512 and without prediction resistance. This configuration cannot be changed.

The TOE obtains entropy to seed the DRBG from the Junos OS Physical Entropy Source, an SP800-90B compliant, physical entropy source validated under the CMVP with ESV certificate E121. The noise source is provided by the Intel Denverton Atom C3508 and Intel Denverton Atom C3758R processors, which are the CPU for the ACX7024 and ACX7024X routers, respectively. The entropy source provides an output of 64 bits through the RDSEED instruction, with an entropy rate of one bit of entropy per bit (i.e. full entropy).

The DRBG is seeded with 384 bits of entropy during initialization and reseeding. This ensures that the DRBG provides a security strength of 256 bits.

7.1.2.10 FCS_SSH_EXT.1

The TOE implements an SSH server for two purposes: a trusted channel between the TOE and a remote audit server, and trusted paths between administrators connected remotely to the TOE. The SSHv2 protocol ensures that the communication over trusted channels and trusted paths is protected against unauthorized disclosure or modification.

Secure connection to a secure, remote server is achieved by setting up an event trace monitor that sends event log messages by using NETCONF over SSH to the remote system event logging server. The remote audit server initiates the connection.

Remote user sessions to the TOE are initiated by administrators from an SSH client. Identification and authentication of remote users are provided by the password-based and public key-based authentication methods specified in the SSHv2 protocol.

The TOE supports password and public-key authentication in the SSHv2 protocol ([RFC4252]☝). For password authentication and keyboard-interactive authentication ([RFC4256]☝), the TOE uses the same user authentication mechanism described in Section 7.1.3.4.

For public-key authentication, the TOE uses the following mechanisms:

- RSASSA-PKCS1-v1_5 using SHA2-256 (rsa-sha2-256) ([RFC8332]☝)
- RSASSA-PKCS1-v1_5 using SHA2-512 (rsa-sha2-512) ([RFC8332]☝)
- ECDSA using SHA2-256 (ecdsa-sha2-nistp256) ([RFC5656]☝)
- ECDSA using SHA2-384 (ecdsa-sha2-nistp384) ([RFC5656]☝)
- ECDSA using SHA2-512 (ecdsa-sha2-nistp521) ([RFC5656]☝)

The TOE drops packets with size greater than 262144 bytes and terminates the connection ([RFC4253]☝).

The TOE allows the following methods for the encryption of SSH sessions: AES in CTR mode with 128-bit (aes128-ctr) and 256-bit keys (aes256-ctr). The TOE permits negotiation of encryption algorithms in each direction. The TOE does not allow the “none” algorithm for encryption ([RFC4253]☝).

The TOE ensures the integrity of the data transmitted by using HMAC with SHA2-256 (hmac-sha2-256) and SHA2-512 (hmac-sha2-512)([RFC4253]☝, [RFC6668]☝).

The TOE supports the following key exchange methods: ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521 ([RFC5656]☝).

The TOE derives session keys from a shared secret obtained during the key exchange ([RFC4253]☝). Hashing algorithms used are SHA2-256 and SHA2-512 ([RFC5656]☝).

The TOE enforces the rekey of session keys. Re-keying of SSH session keys can be configured using the `set system services ssh rekey` with the `data-limit` or `time-limit` keywords. In the evaluated configuration the time-limit must be set within 1 and 60 minutes and the data-limit up to 1 gigabyte ([RFC4251]☝).

7.1.2.11 FCS_SSHS_EXT.1

There are no TSS requirements for this SFR.

7.1.3 Identification and Authentication

7.1.3.1 FIA_AFL.1

The TOE implements password-based authentication for local users and for remote users. The authentication is implemented using the hardened Linux that is part of the TOE software. The TOE software implements the login() using the Pluggable Authentication Modules (PAM) Library calls. The password entered by the user is hashed, and the digest value is compared to the stored reference value. The success or failure of the comparison is returned to login(). PAM is used for authentication management, account management, session management, and password management. The login() primarily uses the session management and password management functions of PAM.

The TOE allows three unsuccessful login attempts before the device disconnects the user. The administrator can modify the default threshold for failed login attempts.

Copyright © 2025 Juniper Networks, Inc.

The administrator can configure the amount of time before the administrator may attempt to log in again after being locked out due to the number of failed login attempts.

7.1.3.2 FIA_PMG_EXT.1

The TOE allows passwords to include any combination of:

- upper-case letters;
- lower-case letters;
- numbers;
- the special characters: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”; and
- other standard ASCII, extended ASCII and Unicode characters.

The administrator can configure a password policy consisting of the following parameters:

- Minimum length (`minimum-length`): between 10 to 20 characters.
- Maximum length (`maximum-length`): between 20 to 128 characters.
- Minimum number of lower-case letters (`minimum-lower-cases`): between 1 to 128.
- Minimum number of upper-case letters (`minimum-upper-cases`): between 1 to 128.
- Minimum number of numeric numbers (`minimum-numeric`): between 1 to 128.
- Minimum number of punctuation numbers (`minimum-punctuations`): between 1 to 128.

In the evaluated configuration, the administrator must configure the minimum length of the password to be between 10 and 20 characters.

7.1.3.3 FIA_UAU.7

There are no TSS requirements for this SFR.

7.1.3.4 FIA_UIA_EXT.1

The TOE supports identification and authentication of local and remote users. Local users connect to the TOE via a management station connected through a serial port, whereas remote users connect to the TOE via a management station connected through a network port using a trusted path using the SSHv2 protocol.

Identification and authentication of local users is performed by using a username and password. The identification and authentication mechanism is described in Section 7.1.3.1.

Identification and authentication of remote users can be performed by either the password-based or public-key authentication methods implemented in the SSHv2 protocol and described in [RFC4252]☝. The TOE performs identification and authentication once the handshaking between the TOE and the SSH peer using the SSHv2 protocol is successfully established. The TOE performs password-based authentication by using the mechanism described in Section 7.1.3.1. For public-key authentication, the TOE identifies and authenticate the user by using its SSH public key stored in the TOE with the mechanism described in [RFC4252]☝.

7.1.4 Security Management

7.1.4.1 FMT_MOF.1/Functions

The administrator can configure the TOE to transmit audit records to an external syslog server. The transmission is over a secure SSHv2 connection which is initiated by the syslog server. If configured, the audit records are sent to the syslog server in real time.

The administrator can also configure the handling of audit data as detailed in Section 7.1.1.3.

The TOE does not allow administrators to modify the behavior when the Local Audit Storage Space is full.

7.1.4.2 FMT_MOF.1/ManualUpdate

There are no specific requirements for non-distributed TOEs.

7.1.4.3 FMT_MOF.1/Services

The administrator can start and stop the following services:

- NETCONF service: together with SSH, this service allows the TOE to transmit audit data to the external syslog server. The administrator uses the `set system services netconf ssh` CLI command to start the service, and `deactivate system services netconf` to stop it.

7.1.4.4 FMT_MTD.1/CoreData

All security management functions are implemented through the Command Line Interface (CLI), which is available once the administrator has successfully identified and authenticated. The TOE does not provide any administrative function accessible prior to the successful authentication of the administrator.

In addition, the TOE enforces access control for the security management functions by defining roles (i.e. login classes), which have different permissions granted. Users other than administrators, once authenticated, do not have permission to execute any security management function.

7.1.4.5 FMT_MTD.1/CryptoKeys

The TOE administrator can perform management functions on cryptographic keys as follows.

Table 12: Cryptographic Key Management

Cryptographic Key	Usage	Generate	Import	Modify	Delete
SSH host (server) key pair	TOE authentication by the SSH peer (external audit server or remote management station)	✓			✓
SSH user (client) public key	User authentication by the TOE (administrator or external audit server)		✓	✓	✓
NTP authentication key	Authentication of NTP server		✓	✓	✓

7.1.4.6 FMT_SMF.1

The TOE implements a Command Line Interface (CLI) which allows the administrators to manage the TOE. The CLI may be accessed locally from console or from a remote management station over a SSHv2 connection. The entire CLI is accessible to all administrators, whether accessing the TOE locally or remotely. The TOE prevents access to the CLI by unauthenticated and unauthorized users.

The TOE provides the following Security Management functions related with the security claims made in this ST:

- Configuring the access banner.
- Configuring the session inactivity time before session termination.
- Updating the TOE software manually, and verifying the update using digital signature capability prior to installation.

- Starting and stopping services.
- Configuring the local audit behaviour.
- Modifying the behavior of transmission of the audit data to the external syslog server.
- Managing the cryptographic keys.
- Configuring cryptographic functionality.
- Configuring the thresholds for SSH rekeying.
- Re-enabling a locked Administrator account.
- Setting the date and time used for time stamps.
- Configuring connection to NTP servers.
- Configuring the local session inactivity time before session termination.
- Configuring the authentication failure parameters.
- Managing the SSH key databases.

For a description of the logging implementation, please see Section 7.1.1.3.

7.1.4.7 FMT_SMR.2

The TOE maintains the Security Administrator role, assigned to users with the right to administer the TOE. When a user is created, it must be assigned to a login class, which can restrict access to the management functionality. Administrators of the TOE are assigned to a fully permissioned login class, which provides full control of the TOE.

7.1.5 Protection of the TSF

7.1.5.1 FPT_APW_EXT.1

User passwords are hashed when stored in the local password file. The TOE can be configured to use SHA2-256 or SHA2-512 as the hashing method.

The CLI does not implement any functionality for accessing the passwords directly. SHA2-256 and SHA2-512 are cryptographically secure. Even if gaining access to the hashed passwords, there is no practical means of recovering the password from the hash value.

7.1.5.2 FPT_SKP_EXT.1

The CLI implemented by the TOE does not include commands for viewing the cryptographic keys. The TOE enforces kernel-level file access rights to the key containers. The access rights granted by the TOE limit access to the contents of cryptographic key containers only to the processes with cryptographic rights and to the shell users with root permission.

7.1.5.3 FPT_STM_EXT.1

The TOE relies on the Linux Operating System, which is part of the TOE, for providing the system time. The Linux OS maintains a system clock that is set at boot time using the real-time clock (RTC) that is implemented in the hardware BIOS. The RTC is a battery-powered clock that keeps track of time even when the system is shut down or the hardware lose power. Then, the system clock keeps the time using the Time Stamp Counter (TSC), a CPU register which counts the number of cycles since it was last reset. The TSC is very fast, has a high resolution, and there are no interrupts. The Linux OS keeps the RTC synchronized with the system clock.

The system clock can be either set by the administrator or synchronized with an NTP Server.

Time obtained from the system clock is used by the TOE for the following security functions:

Copyright © 2025 Juniper Networks, Inc.

- Generate a time stamp on each audit record.
- Implement a timer to track the idle time of a user's session and terminate the session when it reaches the idle-timeout limit.
- Implement a timer to deny a user for re-authenticating after the user is locked for a given number of failed authentication attempts.
- Implement timers for triggering re-keying or termination of a protocol session.

7.1.5.4 FPT_TST_EXT.1

The TOE runs the following set of self-tests when powered on to verify the correct operation of the TOE software:

1. File integrity tests to verify each mounted signed package and to assert that system files have not been tampered. To test the integrity of the software, the TOE uses two mechanisms:
    - The Linux Integrity Mechanism Architecture (IMA), implemented in the Linux kernel, validates the signatures stored in the extended attribute of the each filesystem under the `/soft` directory when it is mounted. These signatures contains the signed hash of file. The mechanism uses RSA with SHA2-256 as the signing algorithm.
    - Calculation of an HMAC value and comparison against a good value stored previously. This mechanism uses HMAC-SHA2-256 and verifies the integrity of the initial RAM disk (initrd), the Junos® OS Evolved Kernel Cryptographic Module and the Junos® OS Evolved OpenSSL Cryptographic Module.

The Junos® OS Evolved Kernel Cryptographic Module and the Junos® OS Evolved OpenSSL Cryptographic Module, which are part of the TOE, execute the following self-tests before providing any cryptographic service:

- Cryptographic Algorithm Self-Tests (CAST), consisting on executing known answer tests of the implemented cryptographic algorithms.

In case of a failure in the integrity test or any of the CASTs in the Junos® OS Evolved Kernel Cryptographic Module, the TOE will panic. The event will be logged, the TOE will cease processing network traffic and CLI commands, and restart. When the TOE restarts, the boot process shall not succeed without passing each self-test. This constitutes the automatic recovery and self-test behavior of the TOE.

In case of a failure in the integrity test or any of the CASTs in the Junos® OS Evolved OpenSSL Cryptographic Module, the module will abort the process and generate a core dump. Core details can be fetched from the output of the `show system core-dumps` CLI. The TOE will detect the failure of the services that aborted and restart them.

7.1.5.5 FPT_TUD_EXT.1

Administrators of the TOE may query the current version of the TOE software using the CLI command `show version`. If a new version of the TOE software is available, the administrators may initiate an update of the TOE software that will happen once the TOE is rebooted. When the software has been loaded onto the device, the CLI command `show system software list` will indicate the current software image as well as the image that is to be loaded into on boot.

The TOE does not allow partial updates. The administrator must upgrade to the entire new release. Updates are downloaded and applied manually. The TOE does not implement automatic updates.

The installable software package containing an update to the TOE software has a digital signature attached. The digital signature is computed using an RSA certificate with a 3072-bit modulus and SHA2-256 in the development environment of the TOE. The TOE verifies the digital signature using a certificate chain embedded into the Junos® OS Evolved Operating System when the image is loaded during boot time. If verification succeeds, then the boot continues and the new image is loaded. If verification fails, the original image of the TOE will be restored and the TOE will boot into it instead.

7.1.6 TOE Access

7.1.6.1 FTA_SSL.3

The TOE terminates remote idle login sessions after a period of inactivity. An idle login session is one in which the CLI displays the operational mode or configuration mode prompt but there is no input from the keyboard.

The Administrator can configure an `idle-timeout` value between 1 and 60 minutes. The CLI starts displaying warning messages (e.g. "Session will be closed in 5 minutes if there is no activity") 5 minutes before the idle timeout limit is reached. In case the parameter is set to a value less than 5 minutes, the message is displayed 1 minute before the idle timeout limit is reached. When the idle session time limit expires, the TOE terminates the login session as well as the SSHv2 session.

7.1.6.2 FTA_SSL.4

User sessions, whether local or remote, can be terminated by the user. When the user issues the `exit` command, the TOE terminates the session. In the case of remote sessions, the SSHv2 session is also terminated.

No user activity can take place until a successful re-authentication.

7.1.6.3 FTA_SSL_EXT.1

The TOE terminates local idle login sessions after a period of inactivity. The TOE treats local sessions in the same way as remote sessions as described in Section 7.1.6.1.

7.1.6.4 FTA_TAB.1

The TOE allows the display of access banners before and after a user logs in. The access banners are globally applied to all methods of access, that is:

- local, through the console port;
- remote, via an SSHv2 session.

The login message and the login announcement are defined at system level by an Administrator.

7.1.7 Trusted Path/Channels

The TOE implements trusted channels and trusted paths using the SSHv2 protocol.

7.1.7.1 FTP_ITC.1

The TOE provides a trusted communication channel using the SSHv2 protocol to send audit records securely to a remote syslog server. The TOE acts as an SSH server using NETCONF. The syslog server runs as an SSH client with NETCONF support configured to receive the streamed syslog messages.

The TOE uses SSHv2 authentication to assure the identification of the external syslog server.

7.1.7.2 FTP_TRP.1/Admin

The TOE provides a trusted path using the SSHv2 protocol for remote administration of the TOE. The administrator uses an SSH client from a remote management station to connect to the TOE.

8 Abbreviations, Terminology, and References

8.1 Abbreviations

AES
Advanced Encryption System

API
Application Programming Interface

CAVP
Cryptographic Algorithm Validation Program

CBC
Cipher Block Chaining

CTR
Counter Mode

CC
Common Criteria

DRBG
Deterministic Random Bit Generator

ECC
Elliptic Curve Cryptography

ECDSA
Elliptic Curve Digital Signature Algorithm

FIPS
Federal Information Processing Standard

HMAC
Keyed-hash Message Authentication Code

HTTP
Hypertext Transfer Protocol

HTTPS
Hypertext Transfer Protocol Secure

IMA
Integrity Mechanism Architecture

NIAP
National Information Assurance Partnership

NTP
Network Time Protocol

OSP
Organizational Security Policies

PCL
Product Compliant List

PP
Protection Profile

PRF
Pseudorandom Function

PSP
Public Security Parameter

RFC
Request for Comments

RSA
Rivest-Shamir-Adleman

SSP
Sensitive Security Parameter

ST
Security Target

TCP
Transmission Control Protocol

UDP
User Datagram Protocol

TOE
Target of Evaluation

TSF
TOE Security Functionality

TSS
TOE Security Summary

8.2 References

CC

Common Criteria for Information Technology Security Evaluation
Version 3.1R5
Date April 2017
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.pdf
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R5.pdf
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R5.pdf

CCEVS-PL05

Applicability and Relationship of NIST Cryptographic Algorithm Validation Program (CAVP) and Cryptographic Module Validation Program (CMVP) to NIAP's Common Criteria Evaluation and Validation Scheme (CCEVS) - Includes addendums #1, #2 and #3
Date 2019-12-06
Location https://www.niap-ccevs.org/Documents_and_Guidance/ccevs/policy-ltr-5-update4.pdf

CCGUIDE

Common Criteria Evaluated Configuration Guide for ACX7024 and ACX7024X Devices
Date 2025-07-19

CPP_ND_V3.0E

collaborative Protection Profile for Network Devices Version 3.0e
Date 2023-12-06

Location https://www.niap-ccevs.org/protectionprofiles/482

FIPS180-4 **Secure Hash Standard (SHS)**
Date 2015-08-04
Location https://csrc.nist.gov/pubs/fips/180-4/upd1/final

FIPS186-5 **Digital Signature Standard (DSS)**
Date 2023-02-03
Location https://csrc.nist.gov/pubs/fips/186-5/final

FIPS198-1 **The Keyed-Hash Message Authentication Code (HMAC)**
Date 2008-07-16
Location https://csrc.nist.gov/pubs/fips/198-1/final

PKG_SSH_V1.0 **Functional Package for SSH Version 1.0**
Date 2021-05-13
Location https://www.niap-ccevs.org/protectionprofiles/459

RFC4251 **The Secure Shell (SSH) Protocol Architecture**
Author(s) T. Ylonen, C. Lonvick
Date 2006-01-01
Location http://www.ietf.org/rfc/rfc4251.txt

RFC4252 **The Secure Shell (SSH) Authentication Protocol**
Author(s) T. Ylonen, C. Lonvick
Date 2006-01-01
Location http://www.ietf.org/rfc/rfc4252.txt

RFC4253 **The Secure Shell (SSH) Transport Layer Protocol**
Author(s) T. Ylonen, C. Lonvick
Date 2006-01-01
Location http://www.ietf.org/rfc/rfc4253.txt

RFC4256 **Generic Message Exchange Authentication for the Secure Shell Protocol (SSH)**
Author(s) F. Cusack, M. Forssen
Date 2006-01-01
Location http://www.ietf.org/rfc/rfc4256.txt

RFC5656 **Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer**
Author(s) D. Stebila, J. Green
Date 2009-12-01
Location http://www.ietf.org/rfc/rfc5656.txt

RFC5905 **Network Time Protocol Version 4: Protocol and Algorithms Specification**
Author(s) D. Mills, J. Martin, J. Burbank, W. Kasch
Date 2010-06-01
Location http://www.ietf.org/rfc/rfc5905.txt

RFC6668 **SHA-2 Data Integrity Verification for the Secure Shell (SSH) Transport Layer Protocol**
Author(s) D. Bider, M. Baushke

Date 2012-07-01
Location http://www.ietf.org/rfc/rfc6668.txt

RFC8332 **Use of RSA Keys with SHA-256 and SHA-512 in the Secure Shell (SSH) Protocol**
Author(s) D. Bider
Date 2018-03-01
Location http://www.ietf.org/rfc/rfc8332.txt

SP800-38A **Recommendation for Block Cipher Modes of Operation: Methods and Techniques**
Date 2001-12-01
Location https://csrc.nist.gov/pubs/sp/800/38/a/final

SP800-56A-Rev3 **Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography**
Date 2018-04-16
Location https://csrc.nist.gov/pubs/sp/800/56/a/r3/final

SP800-90A-Rev1 **Recommendation for Random Number Generation Using Deterministic Random Bit Generators**
Date 2015-06-24
Location https://csrc.nist.gov/pubs/sp/800/90/a/r1/final