Security Target

Juniper SSR Software v6.2.5-5r2 on Juniper SSR120, SSR130, SSR1200, SSR1300, SSR1400 and SSR1500

ST Version: 2.1

Date: August 1, 2024

Prepared for:

Juniper Networks, Inc.

Prepared by:

www.teronlabs.com

Revision History

Version	Date	Author(s)	Description of Change
2.0	July 22, 2024	Teron Labs	Release Version
2.1	August 1, 2024	Teron Labs	Updated TOE version to 6.2.5-5r2

Contents

List of Tables

1 Security Target Introduction

This section is the introduction to the Security Target (ST) identifying and describing the Target of Evaluation (TOE) it defines. The TOE is a suite of Juniper SSR Appliances with Juniper software executing on a Juniper branded platform.

The Security Target Introduction consists of the identification of the ST and the TOE in Sect. 1.1 and of the TOE Overview given in Sect. 1.2.

The TOE and the ST claim conformance to Common Criteria CCv3.1 Revision 5. The TOE claims conformance to the Protection Profile and a Protection Profile Modules in accordance with a Protection Profile Configuration as identified in Table 1. The Terms given are used throughout the Security Target.

Table 1 TOE and ST Conformance Summary

Term	Reference
Base-PP	collaborative Protection Profile for Network Devices Version: 2.2e, Date: 23-March-2020 (CPP_ND_V2.2E)
PP-Module(s)	PP-Module for Stateful Traffic Filter Firewalls, Version 1.4 + Errata 20200625, 25-June-2020 (MOD_CPP_FW_V1.4E)
PP-Configuration	PP-Configuration for Network Device and Stateful Traffic Filter Firewalls, Version 1.4 +Errata20200625, 25 June 2020 (CFG_NDcPP-FW_v1.4e)

1.1 Security Target and TOE Reference

Security Target Title	Security Target Juniper SSR Software v6.2.5-5r2 on Juniper SSR120, SSR130, SSR1200, SSR1300, SSR1400 and SSR1500
Security Target Version	2.1
Security Target Date	August 1, 2024
TOE Identification	Juniper SSR Software v6.2.5-5r2 on Juniper SSR120, SSR130, SSR1200, SSR1300, SSR1400 and SSR1500
TOE Software	Juniper SSR software v6.2.5-5r2 distributed as 128T-6.2.5-5.r2.el7.OTP.v1.x86_64.iso image
TOE Hardware	Juniper SSR120, SSR130, SSR1200, SSR1300, SSR1400 and SSR1500
TOE Developer	Juniper Networks, Inc.
Evaluation Sponsor	Juniper Networks, Inc.

1.2 TOE Overview

TOE Overview summarizes the use and security functions of the TOE and states the physical and logical scope of the TOE.

The TOE Overview commences with an introduction of the use case for the TOE in Sect. 1.2.1. TOE Type in stated in Sect. 1.2.2. The physical and logical scope of the TOE are stated in Sect. 1.2.3 and Sect. 1.2.4, respectively. The

hardware, software and firmware items required by the TOE but which are not parts of the TOE are identified in Sect. 1.2.3. Finally, the functions and parts which are outside the scope of the evaluation are stated in Sect. 1.2.6.

1.2.1 Intended Method of Use

Juniper Session Smart Routing (SSR) is a technology concept for service-centric networking. SSR is ideal for digital businesses, allowing the development of agile, secure, and resilient applications and solutions with Wide Area Network (WAN) connections.

The TOE is a family of Juniper SSR appliances. They consist of software executing on Juniper branded platforms. The TOE is the entire appliance which consists of the platform and the software. The software only SSR solution which the user may deploy on third party platforms is not included in the evaluation. Also, the virtual SSR product is not included in the evaluation. Each variant of the TOE is a bare metal variant.

Each TOE variant includes the same software. Each variant may be provisioned and configured by the user to be a Session Smart Router (in short: Router) or a Session Smart Conductor (in short: Conductor). The TOE configured as a Router implements the data plane and control plane functions of the TOE and performs most functions. The TOE configured as a Conductor implements a centralized management and policy engine allowing provisioning and management of several Routers. A Conductor also acts as an information aggregation repository. SSR appliances may be managed from Juniper MIST cloud, but the cloud-based management is out of scope of the evaluation.

Administrator of the TOE may connect to a TOE configured as a Router locally from console or remotely from a remote management station. The connection between a remote management station and the TOE is protected with SSH. For local administration, the administrator authenticates to the TOE with a username and a password. For remote administration, public key-based and password-based authentication mechanisms are supported.

Each TOE configured as a Router may be administered individually. The Administrator connects to the TOE locally from console or remotely from a remote management station and issues commands to the TOE through a Command Line Interface (CLI).

When a Router is associated to a Conductor, an Administrator may manage the Conductor remotely. The Administrator first establishes a local or remote management connection to the Conductor. The Router and Conductor have established a secure connection between each other using out of band means. The Administrator may issue management commands to the Conductor and those management commands are relayed to the Router over the secure connection.

The security of the network connecting the Routers and Conductors is not claimed by the TOE. The network must be a dedicated out of band network. No other method of interconnecting the Routers and Conductors is allowed.

The TOE implements all security functions of a network device. It also implements a stateful traffic filtering firewall to guard access to the protected network. A typical deployment is illustrated in Figure 1. Instances of the TOE configured as Routers are deployed in various data centres, branches, and other facilities to protect the network connection. The Routers are associated to one or more instances of TOE configured as Conductors for information aggregation, life-cycle management, and configuration management. The Conductor may additionally be connected to other services to utilize the collected information.

The TOE is the entire appliance, including the platform hardware and the TOE software. The platforms are Juniper branded platforms Juniper SSR120, SSR130, SSR1200, SSR1300, SSR1400 and SSR1500. The TOE software is Juniper SSR software v6.2.5-5r2. The TOE software is distributed in an ISO package file 128T-6.2.5-5.r2.el7.OTP.v1.x86_64.iso which includes Oracle Linux 7.9 operating system with kernel version 4.18.0.

TOE software also includes OpenSSL version 1.0.2k and OpenSSH v7.4. The TOE implements SSH on Port 22 for remote administration of a single TOE using OpenSSH v7.4 which uses OpenSSL v1.0.2k. A remote syslog server can

also be connected to the TOE using SSH. Each cryptographic algorithm implemented by the TOE is validated under the Cryptographic Algorithm Validation Program (CAVP).

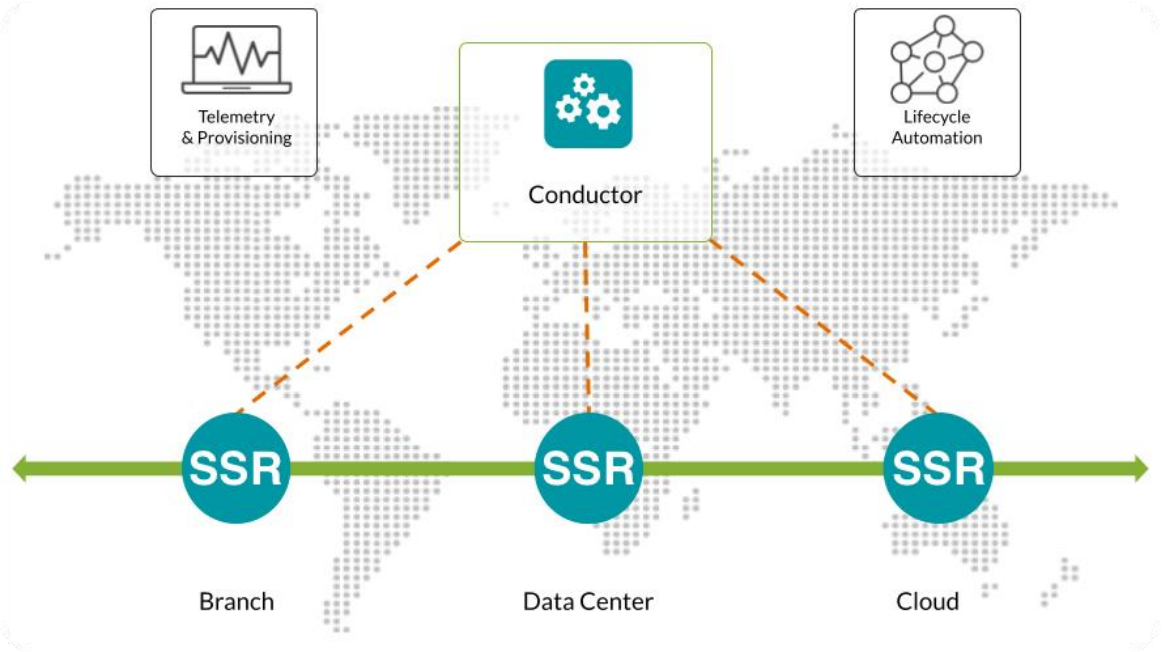

Figure 1 Typical Deployment of the TOE

All software of the TOE is implemented to minimize the attack surface and to only allow the minimum number of connections with the outside users and products. Administration of the TOE is through the CLI. The TOE also implements several security mechanisms to protect itself and the critical data, and to ensure that attempts to tamper with the TOE or the data thereof are detected with a high likelihood.

1.2.2 TOE Type

The TOE is a network appliance implementing the security features required for exact conformance with the Base-PP and the PP-Module. The PP and the PP-Module are used in accordance with the PP-Configuration. The Base-PP, the PP-Module, and the PP-Configuration are identified in Table 1. The TOE is neither a distributed nor a virtual network device.

1.2.3 Physical scope of the TOE

The physical scope of the TOE consists of TOE Hardware, TOE Software and TOE Security Guidance. The physical scope of the TOE is given in Table 2.

Table 2 Physical Scope of the TOE

TOE Hardware	Juniper SSR120, SSR130, SSR1200, SSR1300, SSR1400 and SSR1500
TOE Software	Juniper SSR software v6.2.5-5r2 distributed as 128T-6.2.5-5.r2.el7.OTP.v1.x86_64.iso
TOE Security Guidance	Juniper SSR120, SSR130, SSR1200, SSR1300, SSR 1400 and SSR1500 Common Criteria Installation and User Guide v1.0, June 6, 2024

TOE Hardware is the platform on which the TOE Software is executed. Only the platforms identified in Table 2 are included in the evaluation. The technical characteristics of the hardware platforms are summarized in Table 3.

Table 3 TOE Hardware Variants

Platform	CPU	Microprocessor	Networking
SSR120	4-core Intel Atom	Denverton	2 x 1GbE combo RJ45/SFP 4 x 1GbE RJ45
SSR130	8-core Intel Atom	Denverton	2 x 1GbE combo RJ45/SFP 6 x 1GbE RJ45
SSR1200	8-core AMD	Snowy Owl	4 x 1/10 GbE SFP+ 8 x 1 GbE RJ45
SSR1300	16-core Intel Xeon	Cascade Lake	4 x 10 GbE SFP+ 4 x 1/10 GbE SFP+ 5 x 1 GbE RJ45
SSR1400	24-core Intel Xeon	Cascade Lake	4 x 10 GbE SFP+ 4 x 1/10 GbE SFP+ 5 x 1 GbE RJ45
SSR1500	64-core AMD	Milan	12 x 1/10/25 GbE SFP28 5 x 1 GbE RJ45

TOE Software is the software which runs on the TOE Hardware. The TOE Software as a package-based ISO image. The software image is the same for each TOE Hardware variant. TOE Security Guidance is delivered to all users of the TOE and the TOE must be deployed and operated in accordance with it. TOE Security Guidance extends the existing TOE manuals and other product literature.

1.2.4 Logical scope of the TOE

The logical scope of the TOE includes all security functions and mechanisms required by the Base-PP and the PP-Module. The security functions and mechanisms constituting the logical scope of the TOE is summarized in Table 4.

Table 4 Logical Scope of the TOE

Function/Mechanism	Description
Security Audit	The TOE implements an audit function to collect audit records of all critical security operations. The audit records allow the administrators to analyze the state of the TOE and investigate potential security breaches. Each audit record is a log entry which includes all the necessary data about the event to allow detailed analysis of the audit records.

	The audit records are protected against unauthorized modification and may be transferred to an external syslog server for storage and further analysis. The transfer is protected by SSH.
Cryptography	The TOE implements cryptographic functions for secure communication between the TOE and external devices. The TOE implements a random bit generator used for generating cryptographic keys. The TOE also implements key agreement mechanisms as well all public key cryptographic functions, symmetric cryptographic functions, secure hash functions and keyed hash-based MAC functions. Cryptographic keys and Critical Security Parameters (CSP) are destroyed by the TOE when no longer required. All cryptographic algorithms are validated through the Cryptographic Algorithm Validation Program (CAVP) to ensure correct functioning.
SSH	The TOE implements a SSH Server for secure communication between the TOE and external IT devices. The external IT device may be an audit server or a remote management station. The TOE implements public-key based authentication between itself and other IT devices. The public keys are stored in key containers. The TOE does not implement X.509 certificate-based authentication mechanisms. The TOE implements password-based authentication. Multiple authentication mechanisms are not mandatory. Upon successful authentication, the Command Line Interface (CLI) used for administering the TOE may be accessed by the Administrators over SSH. A TOE configured as a Router may be administered remotely through a TOE configured as a Conductor. The administrator first establishes a local or remote connection to the TOE configured as a Conductor. If that connection is remote, the connection is protected with SSH. The TOE configured as a Conductor may then relay the administrative commands issued by the Administrator to the TOEs configured as Routers. Secure communication between the Router and the Conductor(s) is by means not claimed by the TOE. They must be connected through a dedicated out of band network.
Security Management	The TOE implements a CLI for all the management functions. There are no alternative methods for managing the TOE. Administrators may access the CLI and perform all management tasks of the TOE. The CLI may be accessed locally from console or remotely over a SSH connection.
Identification, Authentication, Access Management	The TOE ensures that only legitimate accesses to the management functions are allowed. Each user is identified with a username and password. Upon successful verification of the password the user is assigned to a role defined by the administrators of the TOE. Users are allowed to change their passwords and the TOE enforces that only good quality passwords are accepted. Passwords are stored in a secure file so that they cannot be read by any user. When a password is entered by a user of a remote management station, the characters are not echoed. Each user may terminate their own session. The TOE also maintains an inactivity timer for each user and if the administrator defined limit is reached, the TOE terminates that session. The TOE also maintains a counter for unsuccessful consecutive authentication attempts for remote users (i.e. Administrators accessing the TOE from a remote

	management station over SSH) and takes protective action in case an administrative defined maximum value is exceeded. Each authentication window displays an administrator configurable access banner. The banned informs the users of the sensitive nature of the TOE and of the sanctions resulting from misuse or abuse of the TOE.
Protection of the TOE	The TOE protects itself from tampering and unauthorized access by active and passive means. The active means are the measures the TOE takes to ensure that TOE data and functions are not accessible to unauthorized users. The passive means are the design characteristics of the TOE which minimize the attack surface accessible to threat agents. The minimization of the attack surface is achieved by the TOE software running on a dedicated hardware platform with a minimum set of physical ports and connections, and only implementing the necessary functions for the TOE. No general computing capabilities are available to the users of the TOE. The active measures the TOE takes to protect itself include – Self-tests at bootup and at request by an administrator to assert correct functioning of the cryptographic functions and other critical parts of the TOE; – Implementation of a NTP synchronized clock which the TOE uses for creating reliable timestamps; – Secure storage of passwords, cryptographic keys and CSPs in a manner which prevents them from being read by unauthorized user and process; – Clearing of the previous content the data structures holding network traffic subjected to firewall rule inspection to ensure that the previously processed traffic information does not influence the next filtering decisions; and – Implementing an update mechanism for the developer to issue upgrades to the TOE software, and protection of the upgrades so that the user of the TOE can verify the authenticity of the upgrade prior to installing it.
Firewall	The TOE implements a stateful packet filtering firewall. Administrators may define rules which are applied for filtering all traffic that passes through the TOE from one network to another. The rules may be expressed for IPv4, IPv6, ICMP, TCP and UDP traffic. Only the traffic which is explicitly declared allowed shall be forwarded by the TOE. All other traffic is dropped. Firewall rules may be applied to each network interface separately, and the administrators may define the order of the rules. The TOE traverses the rule base for each network connection and implements the first rule that matches the traffic.

1.2.5 Non-TOE Hardware, Software and Firmware

A typical deployment of the TOE is given in Figure 1. Three instances of the TOE are configured as SSR Router and one as a Conductor. The Routers are connected to the Conductor through an interconnecting network. The instances of the TOE protect the network interfaces of the premises or facility in which they are deployed. None of the networks are part of the TOE but the network connections are required for the TOE to function.

The security of the interconnection network is not claimed by the TOE. The interconnection network must be a dedicated out of band network.

Each TOE may be administered locally or remotely. Neither the local nor the remote management workstation is part of the TOE but is required for the TOE to be administered.

The remote management workstation must connect to the TOE using SSH. The TOE implements a SSH Server allowing the connection, but the management workstation must implement the SSH Client for the connection.

The TOE connects to an audit server for storing audit records for safe keeping and further processing. The audit server is mandatory but is not part of the TOE.

The TOE connects to a NTP server for synchronizing the clock. The NTP server is not part of the TOE but is required for the operation of the TOE.

The user may deploy additional tools for telemetry and provisioning or to the management of the TOE lifecycle. These are optional and the TOE may be administered and operated without them.

1.2.6 Disallowed Protocols and Services

- The following items are not included in the physical and logical scope of the TOE:
- Non-Juniper branded hardware platforms and Juniper branded hardware platforms not explicitly included in the physical scope of the TOE;
- Security of the communication between the instances of TOE configured as Routers and Conductors;
- Juniper SSR Software for virtual platforms;
- HTTPS/TLS, IPSec, SNMP, RADIUS;
- X.509 certificate management, validation or verification;
- Virtual Private Network (VPN) and Intrusion Prevention System (IPS) functions; and
- Graphical User Interface (GUI) and Juniper MIST for the management of the TOE.

2 Conformance Claims

This section states the Conformance Claims for the ST and the TOE. This includes a statement of the Conformance Claims, a statement of the Conformance Claim Rationale, and the Identification of the Technical Decisions applicable to the TOE.

2.1 Statement of Conformance Claims

The ST and the TOE claim conformance to Common Criteria Version 3.1 Revision 5, Part 1 through to Part 3 identified in the following:

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, April 2017, Version 3.1 Revision 5, CCMB-2017-04-001
- Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, April 2017, Version 3.1 Revision 5, CCMB-2017-04-002
- Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, April 2017, Version 3.1 Revision 5, CCMB-2017-04-003

The ST claims CC Part 2 conformance as CC Part 2 Extended.

The ST claims CC Part 3 conformance as CC Part 3 Conformant.

The ST claims conformance to the following Protection Profile, and the Protection Profile Modules:

- collaborative Protection Profile for Network Devices Version: 2.2e, Date: 23-March-2020 (CPP_ND_V2.2E),
- PP-Module for Stateful Traffic Filter Firewalls, Version 1.4 + Errata 20200625, 25-June-2020 (MOD_CPP_FW_V1.4E), and

Conformance to the Base-PP and the PP-Module is claimed in accordance with the PP-Configuration:

- PP-Configuration for Network Device and Stateful Traffic Filter Firewalls, Version 1.4 +Errata20200625, 25 June 2020 (CFG_NDcPP-FW_v1.4e)

The ST claims no conformance to any Evaluation Assurance Level or any other security assurance requirement package. Security assurance requirements applicable to the TOE are those drawn from the Base-PP as required by Sect. 2.2 of the PP-Configuration.

The ST claims conformance to the PP-Configuration for Network Device and Stateful Traffic Filter Firewalls, Version 1.4 +Errata20200625, 25 June 2020 (CFG_NDcPP-FW_v1.4e) as PP-configuration-conformant.

The ST claims exact conformance to the Base-PP, exact conformance to each PP-Module, and exact conformance to the PP-configuration[1].

2.2 Conformance Claim Rationale

2.2.1 TOE Type Consistency Rationale

The TOE is a non-virtual and non-distributed network appliance. It implements a set of security features required for exact conformance with the Base-PP and with the PP-Module. The PP and the PP-Module are used in accordance with the PP-Configuration. These are exactly the PP, the PP-Module, and the PP-configuration claimed

[1] Exact conformance is defined in *CC and CEM addenda for Exact Conformance, Selection-Based SFRs, and Optional SFRs, CCDB-013-v2.0 Final, 2021-Sep-30.*

in Sect. 2.1. The PP and the PP-Module are exactly as identified in Sect. 1.3 of the PP-Configuration. This ensures that the TOE Type is consistent with the TOE Type in the Base-PP, PP-Modules, and PP-Configuration.

2.2.2 Security Problem Definition Consistency

The statement of the Security Problem Definition in this ST is reproduced exactly from the Base-PP and from the claimed PP-Module. The resulting Security Problem Definition is a union of the Security Problem Definition of the Base-PP and the PP-Module. There are no additional Security Problem Definition elements included in the statement of the Security Problem Definition. This ensures that the statement of the Security Problem Definition is consistent with the PP-Configuration.

2.2.3 Security Objective Consistency

The statement of the Security Objectives in this ST is reproduced exactly from the Base-PP and the PP-Module. The resulting Security Objectives statement is a union of the Security Objectives of the Base-PP and the PP-Module. There are no additional Security Objectives included in the statement of the Security Objectives. This ensures that the statement of the Security Objectives is consistent with the PP-Configuration.

2.2.4 Security Requirements Consistency

The security functional requirements are drawn exactly from the Base-PP and the PP-Module. The statement of the security functional requirements includes all mandatory security requirements and those selection-based security functional requirements applicable to the TOE. The developer claims no optional requirements and does not include additional components in the statement of the security functional requirements. As such, the security functional requirements are consistently drawn from the Base-PP and the PP-Module, and the ST ensures the consistency of the security functional requirements.

The security assurance requirements are drawn from the Base-PP only. This is consistent with Sect. 2.2 of the PP-Configuration. This ensures the consistency of the security assurance requirements.

2.3 Technical Decisions

The Technical Decisions (TD) applicable to the Base-PP are given in Table 5. That is followed by the identification of each TD applicable to the PP-Module. For each TD, the applicability to the ST is stated. For each TD which is not applicable, a brief justification for the exclusion is given.

Table 5 Technical Decisions applicable to the Base-PP

TD	Description	Applicable	Exclusion Rationale (if applicable)
TD 0800	Updated NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	No	The TOE does not claim IPsec
TD 0792	NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR	Yes	
TD 0790	NIT Technical Decision: Clarification Required for testing IPv6	No	The TOE does not claim DTLS or TLS
TD 0738	NIT Technical Decision for Link to Allowed-With List	Yes	

TD 0670	NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	No	The TOE does not claim TLS Client
TD 0639	NIT Technical Decision for Clarification for NTP MAC Keys	Yes	
TD 0638	NIT Technical Decision for Key Pair Generation for Authentication	Yes	
TD 0636	NIT Technical Decision for Clarification of Public Key User Authentication for SSH	No	The TOE does not claim SSH Client.
TD 0635	NIT Technical Decision for TLS Server and Key Agreement Parameters	No	The TOE does not claim TLS Server
TD 0632	NIT Technical Decision for Consistency with Time Data for vNDs	No	The TOE is not a virtual Network Device
TD 0631	NIT Technical Decision for Clarification of public key authentication for SSH Server	Yes	
TD 0592	NIT Technical Decision for Local Storage of Audit Records	Yes	
TD 0591	NIT Technical Decision for Virtual TOEs and hypervisors	No	The TOE is not a virtual TOE
TD 0581	NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	Yes	
TD 0580	NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	Yes	
TD 0572	NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	Yes	
TD 0571	NiT Technical Decision for Guidance on how to handle FIA_AFL.1	Yes	
TD 0570	NiT Technical Decision for Clarification about FIA_AFL.1	Yes	
TD 0569	NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	No	The TOE does not claim DTLS Server
TD 0564	NiT Technical Decision for Vulnerability Analysis Search Criteria	Yes	
TD 0563	NiT Technical Decision for Clarification of audit date information	Yes	
TD 0556	NIT Technical Decision for RFC 5077 question	No	The TOE does not claim TLS Server
TD 0555	NIT Technical Decision for RFC Reference incorrect in TLSS Test	No	The TOE does not claim TLS Server

TD 0547	NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	Yes	
TD 0546	NIT Technical Decision for DTLS - clarification of Application Note 63	No	The TOE does not claim DTLS Client
TD 0537	NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	Yes	
TD 0536	NIT Technical Decision for Update Verification Inconsistency	Yes	
TD 0528	NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	Yes	
TD 0527	Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	No	The TOE does not implement X.509 support.

Table 6 Technical Decisions Applicable to the PP-Module

TD	Description	Applicable	Exclusion Rationale (if applicable)
TD 0827	Aligning MOD_CPP_FW_v1.4E with CPP_ND_V3.0E	Yes	
TD 0551	NIT Technical Decision for Incomplete Mappings of OEs in FW Module v1.4+Errata	Yes	
TD 0545	NIT Technical Decision for Conflicting FW rules cannot be configured (extension of RfI#201837)	Yes	

3 Security Problem Definition

The Security Problem Definition includes a statement of the Threats, Assumptions and OSPs applicable to the TOE. Each is stated in this section.

3.1 Threats

The threats applicable to the TOE are drawn from the Base-PP and from the PP-Modules. There are no additions or omissions, and the wording of each threat statement is taken verbatim. The threats drawn from the Base-PP as applicable to a non-distributed and non-virtual network device are given in Table 7. The threats drawn from the PP-Module are given in the subsequent table.

Table 7 Threats drawn from the Base-PP

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

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

Table 8 Threats drawn from the PP-Module

Threat ID	Threat Statement
T.NETWORK_DISCLOSURE	An attacker may attempt to "map" a subnet to determine the machines that reside on the network, and obtaining the IP addresses of machines, as well as the services (ports) those machines are offering. This information could be used to mount attacks to those machines via the services that are exported.
T.NETWORK_ACCESS	With knowledge of the services that are exported by machines on a subnet, an attacker may attempt to exploit those services by mounting attacks against those services.
T.NETWORK_MISUSE	An attacker may attempt to use services that are exported by machines in a way that is unintended by a site's security policies. For example, an attacker might be able to use a service to "anonymize" the attacker's machine as they mount attacks against others.
T.MALICIOUS_TRAFFIC	An attacker may attempt to send malformed packets to a machine in hopes of causing the network stack or services listening on UDP/TCP ports of the target machine to crash.

3.2 Assumptions

The assumptions applicable to the TOE are drawn from the Base-PP and the applicable PP-Modules. There are no additions or omissions, and the wording of each assumption statement is taken verbatim. The assumptions drawn from the Base-PP as applicable to a non-distributed and non-virtual network device are given in Table 9. There are no additional assumptions defined in the PP-Module.

Table 9 Assumptions Drawn from the Base-PP

Assumption ID	Assumption Statement
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

The Organizational Security Policies (OSP) applicable to the TOE are drawn from the Base-PP. There are no additional OSPs defined int he PP-Module.

Table 10 OSPs Drawn From the Base-PP

OSP ID	OSP Statement
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

The security objectives are stated for the TOE Sect. 4.1 and for the operational environment of the TOE in Sect. 4.2. The security objectives rationale is given in Sect. 4.3.

4.1 Security Objectives for the TOE

The security objectives for the TOE are drawn from the PP-Module. There are no security objectives for the TOE stated on the Base-PP. The security objectives for the TOE are drawn in verbatim from the PP-Module and are stated in Table 11

Table 11 Security Objectives for the TOE Drawn from PP-Module

Security Objective ID	Security Objective Statement
O.,RESIDUAL_ INFORMATION	The TOE shall implement measures to ensure that any previous information content of network packets sent through the TOE is made unavailable either upon deallocation of the memory area containing the network packet or upon allocation of a memory area for a newly arriving network packet or both.
O.STATEFUL_TRAFFIC _FILTERING	The TOE shall perform stateful traffic filtering on network packets that it processess. For this the TOE shall support the definition of stateful traffic filtering rules that allow to permit or drop network packets. The TOE shall support assignment of the stateful traffic filtering rules to each distinct network interface. The TOE shall support the processing of the applicable stateful traffic filtering rules in an administratively defined order. The TOE shall deny the flow of network packets if no matching stateful traffic filtering rule is identified. Depending on the implementation, the TOE might support the stateful traffic filtering of Dynamic Protocols (optional).

4.2 Security Objectives for the Operational Environment

The security objectives for the operational environment are drawn from the Base-PP and the PP-Modules. The security objectives for the operational environment as applicable to a non-virtual and non-distributed network device are drawn in verbatim from the Base-PP and are stated in Table 12. There are no security objectives for the environment stated in the PP-Module.

Table 12 Security Objective for the Operational Environment Drawn from the Base-PP

Security Objective ID	Security Objective Statement
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.RESDUAL_ 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 statement of the security problem definition and the statements of the security objectives are drawn verbatim from the Base-PP and the PP-Module. Therefore, the security objectives rationales given in the Base-PP and in the PP-Module are directly applicable to the ST. They are not repeated here.

5 Security Requirements

This section states the security requirements applicable to the TOE. The statement commences with the extended components definition in Sect. 5.1. The statement of the extended components is followed by the statement of the notations and conventions used in the expression of the security requirements. The security functional requirements are stated in the subsequent subsections on a per functional class basis. The security assurance requirements are only drawn from the Base-PP and are given in Sect. 5.12. The security requirements rationale is given in Sect. 5.13

5.1 Extended Components Definition

The ST references several extended components. Each one is taken verbatim from the Base-PP or the PP-Module. Only the operations allowed in the statement of the extended components are implemented in the ST. There are no additional or modified extended components included in the ST. Therefore, the statement of the extended components is exactly as in the Base-PP and the PP-Module. They are not repeated here.

5.2 Notation and Conventions

This ST follows the specific conventions in the completion of the operations on the Security Functional Requirements. The following conventions are followed to indicate the operations:

- Unaltered Security Functional Requirements are stated using the notation given in CC Part 2 or in the applicable extended component definition.
- When a refinement made in the ST, the added text is indicated with a **bold font** and any removal of text is indicated with a ~~strikethrough~~.
- When a selection is completed in the ST, the selected values are indicated with underlined text.
    - For example, a selection "[selection: disclosure, modification, loss of use]" in a Security Functional Requirement drawn from the Base-PP or PP-Module might become "[disclosure]" when the selection is performed in the ST.
- Assignment completed in the ST is indicated with *italicized font*.
- Assignment completed within a selection in the ST is indicated with *italicized and underlined font*.
    - For example, an assignment within a selection "[selection: change_default, query, modify, delete, [assignment: other operations]]" in a Security Functional Requirement drawn from the Base-PP or PP-Module might become "[change_default,[*select_tag*]]" when both the selection and the assignment are completed in the ST.
- Iteration is indicated by adding a descriptive string starting with "/" (e.g. "FCS_COP.1/Hash").
- Extended requirements are indicated using the notation given in the Base-PP or PP-Module from which they are drawn. Each extended Security Functional Requirement is indicated with a label "_EXT" in the end of the requirement name (e.g. FCS_RBG_EXT).

When the Base-PP or a PP-Module uses an alternative notation or expression for the statement of a Security Functional Requirements, that notation or expression is followed in the ST - possibly with the addition of the above conventions. This includes, for example,

- The capitalization of the component names is followed in verbatim even if sometimes inconsistent, and
- The PP-Module alternatives for selection operations are given in italic font. The italic font is maintained and additionally also underlined to indicate that the selection is performed from the set of allowed values.

The Security Assurance Requirements are drawn from the Base-PP only for conformance with the PP-Configuration. There are no operations defined for the Security Assurance Requirements. The notation for expressing the Security Assurance Requirements is taken verbatim from the Base-PP.

5.3 Security Audit (FAU)

5.3.1 Security Audit Data Generation (FAU_GEN)

5.3.1.1 FAU_GEN.1 Audit data generation (Refinement)

FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) All administrative actions comprising:

- *Administrative login and logout (name of user account shall be logged if individual user accounts are required for Administrators).*
- *Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).*
- *Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).*
- *Resetting passwords (name of related user account shall be logged).*
- *[no other actions]].*

d) Specifically defined auditable events listed in Table 13.

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

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

Table 13 Security Functional Requirements and Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_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_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	None.	None.
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., an IP address).
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
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_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).

FTA_SSL_EXT.1 (if "terminate the session" is selected)	The termination of a local interactive session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	• Initiation of the trusted channel. • Termination of the trusted channel. • Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	• Initiation of the trusted path. • Termination of the trusted path. • Failure of the trusted path functions.	None.
Additional auditable events and audit record content drawn from the PP-Module		
FDP_RIP.2	None.	None.
FFW_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination address Source and destination ports Transport Layer Protocol TOE Interface
FMT_SMF.1/FFW	All management activities of TSF data (including creation, modification and deletion of firewall rules).	None.

5.3.1.2 FAU_GEN.2 User identity association

FAU_GEN.2 User identity association

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

5.3.2 Security audit event storage (Extended - FAU_STG_EXT)

5.3.2.1 FAU_STG_EXT.12 Protected Audit Event Storage

FAU_STG_EXT.1 Protected Audit Event Storage

FAU_STG_EXT.1.1 The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.1.

FAU_STG_EXT.1.2 The TSF shall be able to store generated audit data on the TOE itself. In addition [

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

FAU_STG_EXT.1.3 The TSF shall [*drop new audit data, overwrite previous audit records according to the following rule: [If the local storage for audit data becomes full, the audit function will be stopped, and the new audit data will be dropped]*] when the local storage space for audit data is full.

5.4 Cryptographic Support (FCS)

5.4.1 Cryptographic Key Management (FCS_CKM)

5.4.1.1 FCS_CKM.1 Cryptographic Key Generation (Refinement)

FCS_CKM.1 Cryptographic Key Generation

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

- *RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;*
- *FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 3526].*

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

5.4.1.2 FCS_CKM.2 Cryptographic Key Establishment (Refinement)

FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1[2] The TSF shall **perform** cryptographic **key establishment** in accordance with a specified cryptographic key **establishment** method: *[*

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

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

[2] As per TD 0581
[3] As per TD 0580

5.4.1.3 FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method

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

that meets the following: *No Standard*.

5.4.2 Cryptographic Operation (FCS_COP)

5.4.2.1 FCS_COP.1 Cryptographic Operation

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

FCS_COP.1.1/DataEncryption The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in [*CTR, GCM*] mode and cryptographic key sizes [*128 bits, 256 bits*] that meet the following: AES as specified in ISO 18033-3, [CTR as specified in ISO 10116, GCM as specified in ISO 19772].

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

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

- *RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048 bit, 4096 bits],*

]

that meet the following: *[*

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

].

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-384, SHA-512*] ~~and cryptographic key sizes [*assignment: cryptographic key sizes*~~] and **message digest sizes [*160, 256, 512*] bits** that meet the following: *ISO/IEC 10118-3:2004*.

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

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

5.4.3 NTP Protocol (Extended - FCS_NTP_EXT)

5.4.3.1 FCS_NTP_EXT.1 NTP Protocol

FCS_NTP_EXT.1 NTP Protocol

FCS_NTP_EXT.1.1 The TSF shall use only the following NTP version(s): [*NTP v4 (RFC 5905)*].

FCS_NTP_EXT.1.2 The TSF shall update its system time using [

- Authentication using [*SHA1*] 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.

5.4.4 Random Bit Generation (Extended - FCS_RBG_EXT)

5.4.4.1 FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [*CTR_DRBG(AES)]*.

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [*[1] software-based noise source, [2] platform-based noise source*] with a minimum of [*256 bits*] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and hashes that it will generate.

5.4.5 Cryptographic Protocols (Extended)

5.4.5.1 FCS_SSHC_EXT & FCS_SSHS_EXT SSH Protocol

FCS_SSHS_EXT.1 SSH Server Protocol

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

FCS_SSHS_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following user authentication methods as described in RFC 4252: public key-based, [*password-based*][4].

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [*262,144*] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [*aes128-ctr, aes256-ctr, aes128-gcm@openssh.com, aes256-gcm@openssh.com*].

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

[4] As per TD 0631.

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

FCS_SSHS_EXT.1.7 The TSF shall ensure that [*diffie-hellman-group14-sha1*] and [*diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, diffie-hellman-group18-sha512*] are the only allowed key exchange methods used for the SSH protocols.

FCS_SSHS_EXT.1.8 The TSF shall ensure that within SSH connections, the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

5.5 User Data Protection (FDP)

5.5.1 Residual Information Protection (FDP_RIP)

5.5.1.1 Full Residual Information Protection (FDP_RIP.2)

FDP_RIP.2 Full Residual Information Protection

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon [allocation of the resource to] all objects.

5.6 Firewall (FFW)

5.6.1 Stateful Traffic Filter Firewall (FFW_RUL_EXT)

5.6.1.1 FFW_RUL_EXT.1 Stateful Traffic Filtering

FFW_RUL_EXT.1 Stateful Traffic Filtering

FFW_RUL_EXT.1.1 The TSF shall perform stateful traffic filtering on network packets processed by the TOE.

FFW_RUL_EXT.1.2 The TSF shall allow the definition of stateful traffic filtering rules using the following network protocols and protocol fields:

- *ICMPv4*
  - *Type*
  - *Code*
- *ICMPv6*
  - *Type*
  - *Code*
- *IPv4*
  - *Source address*
  - *Destination address*
  - *Transport Layer Protocol*
- *IPv6*
  - *Source address*
  - *Destination address*

- *Transport Layer Protocol*
- *[no other field]*

- *TCP*
  - *Source port*
  - *Destination port*
- *UDP*
  - *Source port*
  - *Destination port*

and distinct interface.

FFW_RUL_EXT.1.3 The TSF shall allow the following operations to be associated with stateful traffic filtering rules: permit or drop with the capability to log the operation.

FFW_RUL_EXT.1.4 The TSF shall allow the stateful traffic filtering rules to be assigned to each distinct network interface.

FFW_RUL_EXT.1.5 The TSF shall

a) accept a network packet without further processing of Stateful Traffic Filtering rules if it matches an allowed established session for the following protocols: TCP, UDP, [ICMP] based on the following *network packet attributes*:
   1. *TCP: source and destination addresses, source and destination ports, sequence number, Flags;*
   2. *UDP: source and destination addresses, source and destination ports;*
   3. *[ICMP: source and destination addresses, type, [code, [Echo identifier]]]*.

b) Remove existing traffic flows from the set of established traffic flows based on the following: [*session inactivity timeout, completion of the expected information flow*].

FFW_RUL_EXT.1.6 The TSF shall enforce the following default Stateful Traffic Filtering rules on all network traffic:

a) The TSF shall drop and be capable of [logging] packets which are invalid fragments;
b) The TSF shall drop and be capable of [logging] fragmented packets which cannot be re-assembled completely;
c) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a broadcast network;
d) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a multicast network;
e) The TSF shall drop and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;
f) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address "reserved for future use" (i.e. 240.0.0.0/4) as specified in RFC 5735 for Ipv4;
g) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is defined as an "unspecified address" or an address "reserved for future definition and use" (i.e. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for Ipv6;
h) The TSF shall drop and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and

i) [no other rules].

FFW_RUL_EXT.1.7 The TSF shall be capable of dropping and logging according to the following rules:

a) The TSF shall drop and be capable of logging network packets where the source address of the network packet is equal to the address of the network interface where the network packet was received;
b) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is a link-local address;
c) *The TSF shall drop and be capable of logging network packets where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received.*

FFW_RUL_EXT.1.8 The TSF shall process the applicable stateful traffic filtering rules in an administratively defined order.

FFW_RUL_EXT.1.9 The TSF shall deny packet flow if a matching rule is not identified.

FFW_RUL_EXT.1.10 The TSF shall be capable of limiting an administratively defined number of half-open TCP connections. In the event that the configured limit is reached, new connection attempts shall be dropped and the drop event shall be [logged].

5.7 Identification and Authentication (FIA)

5.7.1 Authentication Failure Management (FIA_AFL)

5.7.1.1 FIA_AFL.1 Authentication Failure Management (Refinement)

FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [*1 to 65,535*] unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password.*

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

5.7.2 Password Management (Extended – FIA_PMG_EXT)

5.7.2.1 FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1 Password Management

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: *["!", "@", "#", "$", "%", "^", "&", "*", "(", ")",* [*all other standard ASCII, extended ASCII and Unicode characters*]];
b) Minimum password length shall be configurable to between [*10*] and [*20*] characters.

5.7.3 Protected Authentication Feedback (FIA_UAU)

5.7.3.1 FIA_UAU.7 Protected Authentication Feedback

FIA_UAU.7 Protected Authentication Feedback (Refinement)

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

5.7.4 User Identification and Authentication (Extended - FIA_UIA_EXT)

5.7.4.1 User authentication (FIA_UAU) (Extended – FIA_UAU_EXT)

6.5.4.1 FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2 Password-based Authentication Mechanism

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

5.7.4.2 FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1.1 The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- • [[Response to an ICMP Echo, Establishment of a SSH connection on Port 22 between the TOE and a remote management station]].

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

5.8 Security Management (FMT)

5.8.1 Management of functions in TSF (FMT_MOF)

5.8.1.1 FMT_MOF.1/Functions Management of security functions behaviour

FMT_MOF.1/Functions Management of security functions behaviour

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

5.8.1.2 FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour

FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour

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

5.8.1.3 FMT_MOF.1/Services Management of security functions behaviour

FMT_MOF.1/Services Management of security functions behaviour

FMT_MOF.1.1/Services The TSF shall restrict the ability to **start and stop** ~~the functions~~ **services** to *Security Administrators*.

5.8.2 Management of TSF Data (FMT_MTD)

5.8.2.1 FMT_MTD.1/CoreData Management of TSF Data

FMT_MTD.1/CoreData Management of TSF Data

FMT_MTD.1.1/CoreData The TSF shall restrict the ability to *manage* the *TSF data* to Security Administrators.

5.8.2.2 FMT_MTD.1/CryptoKeys Management of TSF data

FMT_MTD.1/CryptoKeys Management of TSF data

FMT_MTD.1.1/CryptoKeys The TSF shall restrict the ability to manage the cryptographic keys to Security Administrators.

5.8.3 Specification of Management Functions (FMT_SMF)

5.8.3.1 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:

- *Ability to administer the TOE locally and remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the session inactivity time before session termination or locking;*
- *Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1;*
- *[*
    - *Ability to start and stop services;*
    - *Ability to modify the behaviour of the transmission of audit data to an external IT entity;*
    - *Ability to configure thresholds for SSH rekeying;*
    - *Ability to modify the behaviour of the transmission of audit data to an external IT entity;*
    - *Ability to configure NTP;*
    - *Ability to manage the trusted public key databases;[5]].*

FMT_SMF.1/FFW Specification of Management Functions

FMT_SMF.1.1/FFW The TSF shall be capable of performing the following management functions:

- *Ability to configure firewall rules;*

[5] As per TD 0631

5.8.4 Security Management Roles (FMT_SMR)

5.8.4.1 FMT_SMR.2 Restrictions on security roles

FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1 The TSF shall maintain the roles:

- *Security Administrator*.

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions

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

are satisfied.

5.9 Protection of the TSF (FPT)

5.9.1 Protection of Administrator Passwords (Extended – FPT_APW_EXT)

5.9.1.1 FPT_APW_EXT.1 Protection of Administrator Passwords

FPT_APW_EXT.1 Protection of Administrator Passwords

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

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

5.9.2 Protection of the TSF Data (Extended - FPT_SKP_EXT)

5.9.2.1 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)

FPT_SKP_EXT.1 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)

FPT_SKP_EXT.1.1 The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

5.9.3 Time stamps (Extended - FPT_STM_EXT)

5.9.3.1 FPT_STM_EXT.1 Reliable Time Stamps

FPT_STM_EXT.1 Reliable Time Stamps

FPT_STM_EXT.1.1 The TSF shall be able to provide reliable time stamps for its own use.

FPT_STM_EXT.1.2 The TSF shall [*synchronize time with an NTP Server*].

5.9.4 TSF Testing (Extended - FPT_TST_EXT)

5.9.4.1 FPT_TST_EXT.1 TSF Testing (Extended)

FPT_TST_EXT.1 TSF Testing

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [d*uring initial start-up (on power on)*] *at the conditions [key generation, random number generation]* to demonstrate the correct operation of the TSF: [

1. *At start-up:*
    a. *Software integrity and authenticity tests,*
    b. *Known Answer Tests for symmetric cryptographic algorithms,*
    c. *Known Answer Tests for Hash functions and HMAC functions,*
    d. *Known Answer tests for RSA signature computation and verification;*
    e. *Known Answer Tests for DRBGs; and*
    f. *Pair-wise consistency tests for RSA.*
2. *At the key generation:*
    a. *Pair-wise consistency tests for RSA;*
3. *At the random number generation:*
    a. *Continuous RNG tests on all DRBGs.*

]

5.9.5 Trusted Update (FPT_TUD_EXT)

5.9.5.1 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1 The TSF shall provide *Security Administrators* the ability to query the currently executing version of the TOE firmware/software and [*no other TOE firmware/software version*].

FPT_TUD_EXT.1.2 The TSF shall provide *Security Administrators* the ability to manually initiate updates to TOE firmware/software and [*no other update mechanism*].

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a [*digital signature*] prior to installing those updates..

5.10 TOE Access (FTA)

5.10.1 Sesssion Locking and Termination (FTA_SSL)

5.10.1.1 FTA_SSL.3 TSF-initiated Termination (Refinement)

FTA_SSL.3 TSF-initiated Termination

FTA_SSL.3.1: The TSF shall terminate **a remote** interactive session after a *Security Administrator-configurable time interval of session inactivity*.

5.10.1.2 FTA_SSL.4 User-initiated Termination (Refinement)

FTA_SSL.4 User-initiated Termination

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

5.10.2 TSF-initiated Session Locking (Extended – FTA_SSL_EXT)

5.10.2.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions, [

- *terminate the session*]

after a Security Administrator-specified time period of inactivity.

5.10.3 TOE Access Banners (FTA_TAB)

5.10.3.1 FTA_TAB.1 Default TOE Access Banners (Refinement)

FTA_TAB.1 Default TOE Access Banners

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

5.11 Trusted Path/Channels (FTP)

5.11.1 Trusted Channel (FTP_ITC)

5.11.1.1 FTP_ITC.1 Inter-TSF Trusted Channel (Refinement)

FTP_ITC.1 Inter-TSF Trusted Channel

FTP_ITC.1.1 The TSF shall **be capable of using [SSH] to** provide a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [*no other capabiities*]** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from **disclosure and detection of modification of the channel data**.

FTP_ITC.1.2 The TSF shall permit **the TSF or the authorized IT entities** to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [*Communicating with another instance of the TOE acting as an audit server*].

5.11.2 Trusted Path (FTP_TRP)

5.11.2.1 FTP_TRP.1/Admin Trusted Path (Refinement)

FTP_TRP.1/Admin Trusted Path

FTP_TRP.1.1/Admin The TSF shall **be capable of using [*SSH*] to** provide a communication path between itself and **authorized** remote **Administrators** that is logically distinct from other communication paths and

provides assured identification of its end points and protection of the communicated data from **disclosure and provides detection of modification of the channel data**.

FTP_TRP.1.2/Admin The TSF shall permit remote **Administrators** to initiate communication via the trusted path.

FTP_TRP.1.3/Admin The TSF shall require the use of the trusted path for initial Administrator authentication and all remote administration actions.

5.12 Security Assurance Requirements

This section states the Security Assurance Requirements. For conformance with the PP-Configuration, the Security Assurance Requirements are drawn from the Base-PP only. The applicable Security Assurance Requirements are stated in Table 14.

Table 14 Security Assurance Requirements

Security Assurance Class	Security Assurance Components
Security Target (ASE)	Conformance claims (ASE_CCL.1) Extended components definition (ASE_ECD.1) ST Introduction (ASE_INT.1) Security objectives for the operational environment (ASE_OBJ.1) Stated security requirements (ASE_REQ.1) Security Problem Definition (ASE_SPD.1)
Development (ADV)	Basic functional specification (ADV_FSP.1)
Guidance Documents (AGD)	Operational user guidance (AGD_OPE.1) Preparative procedures (AGD_PRE.1)
Life Cycle Support (ALS)	Labelling of the TOE (ALC_CMC.1) TOE CM Coverage (ALC_CMS.1)
Tests (ATE)	Independent testing - conformance (ATE_IND.1)
Vulnerability Assessment (AVA)	Vulnerability survey (AVA_VAN.1)

5.13 Security Requirements Rationale

The Security Functional Requirements are drawn from the Base-PP and PP-Module and not from any other source. The ST claims exact conformance to the Base-PP and to the PP-Module. The Security Functional Requirements include each mandatory requirement and each applicable optional and selection-based requirement. Only the operations allowed in the Base-PP and the PP-Module are implemented. Therefore, the Security Functional Rationales of the Base-PP and the PP-Module are directly applicable to the ST as well. They are not repeated here.

The Security Assurance Requirements are drawn from the Base-PP only as required by the PP-Configuration. None. are added or removed. Therefore, the Security Assurance Requirements Rationale of the Base-PP is directly applicable to the ST as well. It is not repeated here.

6 TOE Summary Specification

The TOE Summary Specification includes the description how the TOE fulfills the security functional requirements, and how the developer and the evaluator fulfill the security assurance requirements. Each is described in this section. Additional details on the cryptographic algorithms and protocols implemented in the TOE are also given.

6.1 Fulfillment of the Security Assurance Requirements

The fulfilment of the Security Assurance Components by the TOE is given in Table 15. Each Security Functional Component applicable to the TOE is listed and the fulfilment of that component described.

Table 15 Fulfillment of the Security Assurance Components

Security Assurance Component	Fulfillment
ASE_CCL.1 ASE_ECD.1 ASE_INT.1 ASE_OBJ.1 ASE_REQ.1 ASE_SPD.1 ASE_TSS.1	The developer authors a well formed, complete Security Target (i.e. this document) which addresses all applicable requirements from the Assurance Class ASE: Security Target Evaluation as well as all requirements for the content of the TOE Summary Specification stated in the Supporting Documents of the claimed Protection Profiles. The Security Targets consists of the ST Introduction (ASE_INT.1), a statement of conformance claims (ASE_CCL.1), a statement of the Security Problem Definition (ASE_SPD.1), a statement of Security Objectives for the TOE and for the environment (ASE_OBJ.1), a statement of the Security Requirements (ASE_REQ.1) and a TOE Summary Specification (ASE_TSS.1). Each assurance component is defined in [CCPart3] with additional requirements originating from the relevant Protection Profiles and the Supporting Documents associated to them. The Security Target is evaluated independently of the TOE by the evaluation facility to ensure that it is a complete, correct and accurate representation of the security problem the TOE solves.
ADV_FSP.1	The developer includes a Functional Specification in the TOE Summary Specification. The Functional specification details the external interfaces of the TOE to the level of detail required by the requirements stated for the TOE Summary Specification for each relevant Security Functional Requirement in the Supporting Documents for the applicable Protection Profiles and under ADV_FSP.1.
AGD_OPE.1 AGD_PRE.1	The developer produces a separate Common Criteria Guidance Supplement. The Guidance Supplement is made available to all users of the TOE and the users are required to set up the TOE and the environment thereof and at all times to operate the TOE in accordance with the instructions given in the Guidance Supplement. The Guidance supplement meets all the requirements stated for the Guidance for each applicable SFR in the Supporting Documents of the applicable Protection Profiles and under the requirements for operative guidance (AGD_OPE.1) and preparatory guidance (AGD_PRE.1).

ALC_CMC.1 ALC_CMS.1	The TOE is labelled with a unique reference. The unique reference allows the user to verify that they are using the exactly correct version of the TOE. The unique reference is given in the TOE Identification. The TOE is developed using a Configuration Management System (CMS) into which the developer checks the TOE itself as well all the evaluation evidence produced throughout the development of the TOE. The CMS must meet the capabilities stated in ALC_CMC.1 and include in the scope configuration items stated in ALC_CMS.1.
ATE_IND.1 AVA_VAN.1	The developer shall provide the TOE for testing by the evaluation facility. The TOE shall be provided for testing in a configuration in which it is possible for the evaluation facility to test it for functional correctness (ATE_IND.1) and also to carry out the exact tests stated for each applicable Security Functional Requirement in the Supporting Documents of the applicable Protection Profiles (AVA_VAN.1). Specifically, the developer shall associate to the TOE a document identifying all software and hardware parts which constitute the TOE. This ensures that the developer meets ATE_IND.1 for independent testing of the TOE and that the evaluation facility is able to perform each required test.

6.2 Fulfillment of the Security Functional Requirements

The fulfillment of the Security Functional Components is described in XXX.

Security Functional Component	Fulfillment
FAU_GEN.1 FAU_GEN.2	The TOE implements an audit function using syslog. Audit records are generated and stored for the following events and for each event related to specific SFRs as enumerated in Table 13: – Start-up and shut-down of the audit functions; – All administrative actions comprising of administrative login and logout, including the user account; – Changes to TSF data related to configuration changes, including the information that a change occurred and what was changed; – Generating/import of, changing, or deleting of cryptographic keys, the event itself and a unique key name or key reference of the affected keys; – Resetting passwords, including the identification of the user account, For each audit log entry, the TOE stores the date and time of the event and/or reaction, the type of the event and/or reaction, identity of the subject if applicable, the outcome of the event if applicable, and all the additional SFR-specific data enumerated in Table 13. All cryptographic keys are obscured when stored in audit logs to ensure that they shall not be disclosed. For the ephemeral SSH session keys the PID is used as the key reference to relate the audit events on key generation and key destruction. Key destruction is recorded as a session termination event.

	The TOE implements a clock which is used for a time source for time stamps. The clock is synchronized using NTP. Each audit record includes a time stamp which states the exact time on which the auditable event occurred.
FAU_STG_EXT.1	Each instance of a TOE is a standalone appliance which generates and stores the audit log entries and stores them locally. The log files are not protected by cryptographic means but there are no CLI functions or other means for modifying them. Audit records on the TOE are stored locally as syslog entries. The TOE may also accept a SSH connection from an external audit server and forward the syslog entries to the audit server over the SSH connection. The TOE stores the syslog entries locally in a disc partition of a fixed size. The size is defined at the provisioning of the TOE and may not be modified by the Administrator. If that partition becomes full, the TOE shall stop the auditing service. The stopping of the service shall generate an audit record which shall be stored in the log file. Each subsequent new audit record shall be dropped.
FCS_CKM.1 FCS_CKM.2	The TOE only uses RSA as an asymmetric algorithm for SSH authentication keys and key establishment keys. The key size used in 2048 or 4096 bits. SSH host keys used for host authentication are generated at the first start-up of the TOE (if not already configured) using the RSA scheme as defined in FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3. SSH key establishment keys are generated and exchanged in accordance with Diffie-Hellman on group 14, specifically using diffie-hellman-group14-sha1 as defined in RFC 4253 and diffie-hellman-group14-sha256 in RFC 8268. Diffie-hellman-group16-sha512 and diffie-hellman-group18-sha512 are also used as defined in RFC 8268. Diffie-Hellman Groups are defined in RFC 3526. The exact method key generation and exchange is defined in NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography".
FCS_CKM.4	Each cryptographic key and CSP the TOE uses is identified in Table 16. The table also describes for each cryptographic key and CSP the purpose of the key or CSP, and how the key or CSP is stored by the TOE. For each cryptographic key and CSP that is destroyed throughout the life-cycle of the TOE, the table states how the key or CSP is destroyed.
FCS_COP.1/DataEncryption	The TOE implements AES for the symmetric encryption and decryption of data on SSH connections. AES is used in CTR and GCM modes. The CTR mode is in accordance with ISO 10116 and the GCM mode is in accordance with ISO 19772. Key sizes supported are 128 bits and 256 bits. Each key size is used for each mode.
FCS_COP.1/Hash	Hash functions are used by the TOE for NTP time stamp authentication, protection of the passwords, SSH, and for the verification of the authenticity of TOE software upgrades. For NTP the TOE uses SHA-1. User passwords are protected with SHA-512 when stored on the TOE. For SSH the TOE uses SHA-256 and SHA-512. For the

<table>
<tr><td></td><td>verification of the TOE software upgrades using a digital signature computed by the developer of the TOE, the TOE uses SHA-256.</td></tr>
<tr><td>FCS_COP.1/KeyedHash</td><td>The TOE implements two HMAC functions for use with SSH, HMAC-SHA-256 and HMAC-SHA-256. The details of the algorithms are given in the following:
<table>
<tr><td></td><td>HMAC-SHA-256</td><td>HMAC-SHA-512</td></tr>
<tr><td>Key Length</td><td>256 bits</td><td>512 bits</td></tr>
<tr><td>Hash function</td><td>SHA-256</td><td>SHA-512</td></tr>
<tr><td>Block Size</td><td>512 bits</td><td>1024 bits</td></tr>
<tr><td>Output MAC length</td><td>256 bits</td><td>512 bits</td></tr>
</table></td></tr>
<tr><td>FCS_COP.1/SigGen</td><td>The TOE only uses RSA for the generation of digital signatures. The digital signatures are generated with a 2048 or 40496 bit key (modulus).</td></tr>
<tr><td>FCS_NTP.1</td><td>The TOE implements NTPv4 as defined in RFC5905 for synchronizing the clock of the TOE with a NTP server. Time synchronization messages are protected with SHA-1 which the TOE verifies prior to accepting the time. No timestamps are accepted from multicast or broadcast addressed. The TOE is capable of connecting to three or more NTP Servers as configured by the Administrator</td></tr>
<tr><td>FCS_RBG_EXT.1</td><td>The TOE generates random bits in accordance with NIST Special Publication 800-90 using CTR_DRBG (AES). The RBG does not require any configuration and is seeded from two platform-based entropy sources and one software-based entropy source.

The platform-based entropy sources are used by the Linux kernel to seed the internal DRBG and produce random bits which are written to the /dev/urandom. The entropy sources are the Intel CPU RDRAND instruction (on those variations of the TOE where available) and the network packet jitter are used at the boot time to seed the Linux random number generator which then produces randomness and writes it to /dev/urandom. The Linux kernel may also write randomness to other files but only that written to /dev/urandom is used by the TOE.

The randomness from /dev/urandom is read and used as a seed by the OpenSSL library DRBG. 256 bits of entropy is read from /dev/urandom and used for seeeding the OpenSSL DRBG. The OpenSSL DRBG is then used by the TOE for the run-time generation of random bits as required.</td></tr>
<tr><td>FCS_SSHS_EXT.1</td><td>The TOE implements a SSH Server which may be used for SSH connections between the TOE and the remote management station, and between the TOE and an audit server.

The TOE uses a 2048-bit RSA Host Key for SSH v2. The key is generated randomly at the initial setup of the TOE and is with an overwhelming probability unique to each host. The key cannot be managed using the CLI.

The client presents the TOE with its public key which the TOE matches against its authorized_keys list of keys. When a client connects to the TOE, the client will be able to determine if the same host key was used in previous connections, or if the key is different.</td></tr>
</table>

	The TOE implements all mandatory cryptographic algorithms and methods and accepts public-key based authentication and password-based authentication. Multiple authentication mechanisms for users is not required. Port forwarding and sessions to clients are allowed. X11 forwarding is prohibited. The TOE does not accept the "none" cipher and implements aes128-ctr, aes256-ctr, aes128-gcm@openssh.com and aes256-gcm@openssh.com for the protection of data on administrative sessions. The TOE rekeys every 1Gb bytes data or after a session life-time reaches sixty minutes, whichever occurs first. The client may request a rekeying event as a valid SSHv2 message at any time and the TOE will honor this request. Re-keying of session keys can be configured using the sshd_config knob. When a connection is brought down, the TOE does not attempt to re-establish it. Key exchange is performed only using the supported key exchange algorithms ordered as follows: diffie-hellman-group14-sha1, diffie-hellman-group14-sha256, diffie-hellman-group16-sha512 and diffie-hellman-group18-sha512. The TOE sshd server does not support debug messages via the CLI. The TOE implements a timeout period for authentication of the SSHv2 protocol and enforces a limit of three failed authentication attempts before sending a disconnect to the client. The TOE does not accept authentication if the requested service does not exist. Authentication requests for non-existent usernames will not succeed. The TOE returns a disconnect message as it would for failed authentications. This prevents attackers from enumerating valid usernames. Authentication method "none" is not allowed. The TOE responds to it with a list of permitted authentication methods. The TOE implements public key authentication for SSHv2 session authentication using ssh-rsa, rsa-sha2-256 and rsa-sha2-512. Public key authentication method implicit is also supported. Authentication of administrative sessions succeeds if a correct administrative password is presented to the TOE. The TOE does not require multiple authentications (public key and password) for users. The TOE does not support the configuration of host-based authentication methods. The TOE reads the packet payload size in the TCP packet to determine the packet length. Packets greater than 262,144 bytes are dropped and the connection is terminated. Negotiation of HMAC-SHA1 in each direction for SSH transport is not allowed. Diffie-hellman-group14-sha1 is supported. Key re-exchange is performed when SSH_MSG_KEXINIT is received. The TOE implements aes128-ctr, aes256-ctr, aes128-gcm@openssh.com and aes256-gcm@openssh.com for encryption. The TOE does not implement the recommended modes AES192-ctr or 3des-ctr. None of the optional modes are implemented. The recommended and optional algorithms hmac-sha2-256 and hmac-sha2-512 are implemented for SSH transport.

FDP_RIP.2	If a session exists, the TOE associates each packet to the session. The forwarding decision is cached and applied to each packet of a session. In case of stateless protocols (i.e. UDP), a timer is set and the forwarding decisions are cached until the timer expires. When a stateful session is terminated or a stateless session timer expires, the TOE erases the cached decision to ensure that a fresh decision is made on a new session based on the filtering rules not influenced by the previous decisions.
FFW_RUL_EXT.1	The TOE implements CLI commands for the Administrator to configure the stateful packet filtering rules. The rules are constructed based on header fields in IPv4, IPv6, ICMPv4, ICMPv6, TCP and UDP headers, and applied in the sequence in which they are stored in the rule base to all network traffic processed by the TOE. The TOE is configured to associate network interfaces to IP subnets and source IP addresses are associated with network interfaces. When the TOE boots up, it executes a suite of self-tests. In order for the boot sequence to proceed, each self-test must pass. Network interfaces of the TOE are only activated when all functions required for processing the datagrams are verified and loaded. This ensures that the only when the TOE is fully operational and all rules enforced before receiving any traffic through the physical interfaces. Datagram processing is controlled by a flow daemon. If the flow daemon fails, packet processing will stop and no traffic will be forwarded. This ensures that a failure in other daemons will not prevent the flow daemon from enforcing the TOE security policies. Also, stopping of the packet forwarding in case of a failure in the flow daemon ensures that the default deny policy is enforced by the TOE. The rules in the rule base are examined in sequence. If a rule is found to match a datagram, the action of that rule is performed. Ergo, only if an explicit rule exists to allow traffic, the traffic shall be forwarded. If no rule in the rule bases allows specific traffic, datagrams are dropped and not logged. Sessions are not created for denied traffic, only for traffic which is allowed. The following as packets that will be automatically dropped and are counted but not logged: – Packets which are invalid fragments, i.e. any fragment that deviates from the followed standard, – Fragments that cannot be completely re-assembled, – Packets where the source address is defined as being on a broadcast or multicast network, – Packets where the source address is defined as being a loopback address, – Packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address “reserved for future use” (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4, – Packets where the source or destination address of the network packet is defined as an “unspecified address” or an address “reserved for future definition and use” (i.e. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for IPv6, and

	– Packets with the IP options Loose Source Routing, Strict Source Routing, or Record Route specified. Additionally, the following traffic is dropped and counted: – Packets where the source address is equal to the address of the network interface where the network packet was received, – Packets where the source or destination address of the network packet is a link-local address, and – Packets where the source address does not belong to the networks associated with the network interface where the network packet was received. For stateful traffic, the TOE maintains each session, and associates packets belonging to that session throughout the existence of a session. A cached forwarding decision shall be applied to each packet. For UDP traffic, the TOE maintains a decision threshold timer which is used for determining whether the packets belong to the same flow.
FIA_AFL.1	The Administrator always identifies and authenticates to the TOE using a username and password. The authentication may be locally from a console or remotely from a remote management station but the same method of dealing with authentication failures applies. For each username, the TOE starts a counter for the failed, consecutive authentication attempts. If the authentication attempt fails, the counter value is incremented. If the counter reaches the Administrator-configured maximum value for authentication failures, the offending account is locked for a period of time set by the Administrator. While locked, no authentication attempts are allowed on that account. When an account is locked, other Administrator accounts will remain active, and the locked account shall be unlocked once the locking period expires.
FIA_PMG_EXT.1	Authentication data for fixed password authentication is a case-sensitive, alphanumeric value. The password has a minimum length of 8 characters and maximum length of 15 characters. Each password must contain characters from at least two different character sets (upper, lower, numeric, punctuation). Any standard ASCII, extended ASCII and Unicode characters can be selected when choosing a password. Specifically, the following special characters are allowed: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(”, “)”.
FIA_UAU.7	When authenticating from a local management station (i.e. the console), the TOE does not echo the characters entered by the user. This prevents unauthorized users from learning the passwords or any password information (e.g. the number of characters) by monitoring the display of the remote management station.
FIA_UAU_EXT.2 FIA_UIA_EXT.1	Each user is associated with a username and password. When a user authenticates, the user identifies himself/herself with a username and enters a password for authentication. For each user, the TOE stores a SHA-512 digest of a reference password created when the user selects the password. The entered password is hashed, and the two hashes are compared. If they math, the authentication is considered successful, and the user is granted access to the

	TOE. If the user has been assigned to a role with sufficient credentials, the CLI shall be made available to the user. Predominantly, functions of the TOE are only made available to successfully authenticated users assigned to a role with sufficient credentials to access the function. However, there are the following exceptions: As part of the authentication exchange, the TOE displays an access banner to all users. This is part of the authentication window and is displayed to each user even if not yet authenticated. The TOE does respond to ICMP Echo messages to allow basic diagnostics even if there is no authenticated user session active. The TOE allows establishment of SSH connections between itself and remote IT products. The remote IT product may be an audit server or a remote management station. The connection must be requested by the remote IT product. Users can connect to the TOE by two means: 1. From console, when the user authenticates to the TOE from a console directly connected to the TOE. In this case, the authentication exchange is carried out directly between the TOE and the user. 2. From a remote management workstation, in which case the user authenticates to the TOE from a remote (i.e. connected over a network) management station. The management station first establishes a SSH connection between itself and the TOE and upon establishment of the connection, carries out the authentication exchange with the user. Authentication of the remote IT products is through SSH and pre-distributed public keys. The TOE does not implement X.509 certificate-based authentication methods.
FMT_MOF.1/Functions	The TOE allows the administrator to modify the SSH Server behavior to control how the syslog server may connect to the TOE. The syslog behaviour is configurable under “configure authority router router system syslog”.
FMT_MOF.1/ManualUpdate	The TOE notifies the Administrators accessing the TOE through the CLI automatically of the availability of software updates. Once an update is available, an Administrator (i.e. a user with sufficient credentials to gain access to the CLI) may use the CLI to manually update the TOE software. Users with no Administrative privileges are not granted access to the CLI and can, therefore, not update the TOE software. The updating of the TOE software is done as described under FPT_TUD_EXT.1.
FMT_MOF.1/Services	The Administrator may turn the auditing service on and off. That is done through the CLI by setting the audit administration enabled flag to be of value true. Each turning on or off of the auditing service generates a log entry into the audit logs.
FMT_MTD.1/CoreData	The TOE only allows access to the management functions of the TOE to the users assigned to an Administrator role. Non-management functions may be made available to the successfully identified and authenticated users assigned to

	the user roles. The only TOE functions available to the users prior to a successful identification and authentication are the following: Displaying the access banner. The administrators may configure an access banned which is displayed to the users when attempting to use the TOE locally or from a remote management station. The access banned is only displayed and does not allow any means of entering data or manipulating the TOE. Responding to ICMP Echo. The Echo protocol is a simple IP layer Request-Reply protocol for other IT devices to query the status of the TOE. ICMP echo datagrams do not require session establishment and do not carry payload. They cannot be used for accessing the TSF data or TOE functions other than responding to an ICMP Echo request. SSH connection between the TOE and a remote management station. SSH over Port 22 will make available to the remote user an authentication window in which the remote user may authenticate as a legitimate Administrator. Access to the CLI which is the only method of accessing the TOE shall only be made available upon successful identification and authentication. SSH itself cannot be used for issuing CLI or other commands to the TOE.
FMT_MTD.1/CryptoKeys	Most functions on the cryptographic keys are implemented as part of the cryptographic protocols of the TOE. The protocols are executed by the TOE software without Administrator intervention. Nevertheless, the Administrator may configure the cryptographic keys used by the cryptographic Protocols by two means: The Administrator may modify the file authorized_keys located in the .ssh folder in the home folder of the account used. The file contains the public keys of Clients allowed to connect to the TOE over SSH. The Administrator may configure the rekeying thresholds for SSH through the sshd_config knob.
FMT_SMF.1 FMT_SMF.1.1/FFW	The TOE implements a CLI where a command exists for each management and configuration function of the TOE. There are no other methods of management of the TOE. The CLI commands may be issued by three different means: 1. From a local console where the Administrator is in the same physical space than the TOE and uses a management workstation connected directly to the console port of the TOE. 2. From a remote management station where the Administrator is not in the immediate proximity of the TOE and uses a management station to connect to the TOE over a TCP/IP network. The connection between the TOE and the remote management station is protected by SSH over Port 22. 3. From another instance of a TOE configured into a Conductor. The Administrator connects to the TOE locally or from a remote management workstation, and issues life-cycle and configuration management commands to a range of instances of TOE configured as Routers. The communication between the two instances of the TOE

	(one configured as a Router and the other configured as a Conductor) is protected by out of band means. The CLI is made available to the successfully authenticated administrators in entirety. There are no CLI subsets made available to roles other than Administrator.
FMT_SMR.2	The TOE implements a role Security Administrator. Security Administrator is the only role to which the CLI is made available, i.e. which is authorized to administer the TOE. If the identification and authentication of a user is successful and the user is authorized to administer the TOE, the user is assigned a role Security Administrator. Users assigned to non-administrative roles are not granted access to the administration functions of the TOE. The role assignment remains until the session is terminated. The TOE does not implement a role hierarchy. Each user successfully authenticated and assigned to the role Security Administrator is granted access to the entire CLI. When accessing the TOE from console, human users are identified and authenticated with a username and password. The administrator may also configure the TOE to accept SSH public-key based authentication for the users accessing the TOE from the remote management station.
FPT_APW_EXT.1	The TOE protects authentication data by two means: 1. Reference passwords are not stored in plain text but as SHA-512 digests of the reference passwords. When a password entered by a user is compared to a reference password, a SHA-512 digest is computed from the entered password and the two digests are compared. 2. The TOE is only administered or otherwise accessed through the CLI. The CLI does not implement any functions for reading or exporting the passwords.
FPT_SKP_EXT.1	The TOE stores the cryptographic keys and CSPs as described in Table 16. Cryptographic keys are only accessed by authorized processes. There are no methods available for users to execute unauthorized processes on the TOE and the CLI does not implement any means of reading or exporting the private or symmetric cryptographic keys.
FPT_STM_EXT.1	The TOE implements a real time system clock which may be used for time stamps and clock cycles when reliable time is required. There is no CLI command for setting the time but the TOE clock is synchronized with a NTP Server. The time is used by the TOE in the following functions: 1. Time stamps for the audit records. Each entry in the audit logs is stamped with a time stamp stating the date and time of the event. 2. Inactivity timers for the active user sessions. The TOE maintains an inactivity timer for each user session and if the idle time of the session reaches the set threshold, the TOE shall terminate the session. 3. Lockdown timers. If the number of consecutive failed authentication attempts on any used account exceeds the Administrator-defined

	threshold, the account shall be locked for 1800 seconds. The TOE sets a timer and once the timer expires, the account is unlocked. 4. SSH host authentication request expiration timer. 5. Rekeying timer for SSH connections on Port 22. SSH shall rekey when either an administrator-defined maximum amount of data per connection is reached or an administrator-defined maximum lifetime of the connection is reached. The default values are 1Gb of data or one hour connection lifetime.
FPT_TST_EXT.1	The TOE is to be used with the FIPS mode enabled. This ensures that the suite of self-tests on the TOE is executed at the start-up and when generating cryptographic keys of random numbers. The self-tests at the start-up commence with the TOE software integrity and authenticity test. The integrity and authenticity of the software is verified against a digital signature of the software computed at the development environment with RSA and SHA2-256. 2048-bit or 4096-bit RSA key is used. If the signature verification succeeds, the TOE software is considered unaltered and authentic, and the TOE shall proceed with the boot sequence. The next step in the boot sequence are the power-up self-tests. The power-up self-tests consist of the algorithm-specific Pairwise Consistency and Known Answer tests. If any component of the power-up self-test fails, an internal global error flag is set to prevent subsequent invocation of any cryptographic function calls. Any such power-up self-test failure is a hard error that can only be recovered by reinstalling the module. The power-up self-tests may be performed at any time by reloading the module. Additional pair-wise consistency tests are executed for asymmetric ciphers when generating cryptographic keys are generated, and the continuous RNG tests when random numbers are generated. The RNG test are in accordance with the requirements of SP800-90A. No operator intervention is required during the running of the self-tests.
FPT_TUD_EXT.1	The currently active version of the TOE can be queried by a legitimate Administrator by the CLI command show system version. Availability of software upgrades can be queried with the show assets software command which displays all available software upgrades. The downloading of the upgrade is done manually as is the commencement of the actual upgrade. TOE software is distributed as a package-based ISO image file. The package-based ISO format allows upgrading of the packages constituting the ISO file when distributed in RPM (Red Hat Package Manager) format. Upgrades to the TOE software are distributed in RPM format. The RPM format allows embedding of a digital signature into the package. Into each upgrade package is embedded a GPG digital signature produced using RSA with 2048 or 4096 bit key and SHA-256. The Package Manager of the Linux operating system which is part of the TOE automatically verifies the digital signature on the downloaded package. Only if the signature verification is successful shall the package manager install the upgrade.

FTA_SSL.3 FTA_SSL4. FTA_SSL_EXT.1	The TOE implements command quit which terminates the current session. An Administrator may issue that at any time to terminate the administrative session. For each administrative session the TOE maintains an inactivity timer which tracks the time the administrator is idle, i.e. not issuing any commands. If the inactivity timer reaches a configured maximum time, the TOE shall terminate the administrative session.
FTA_TAB.1	The TOE only implements password-based authentication of Administrators. The authentication exchange may take place on console or on a remote management station. In both cases, the same authentication exchange is performed. The TOE displays on each authentication exchange the Administrator-configurable access banner containing consent warning messages and any other warnings or information the Administrator may choose to display on each authentication exchange.
FTP_ITC.1	The TOE implements a SSH Server to protect confidentiality and integrity of communication between itself and a remote syslog server and between itself and a remote management station. When the TOE and a syslog server establish a SSH connection, the connection is initiated by the syslog server. When the TOE and a remote management station establish a SSH connection, the connection is initiated by the remote management station.
FTP_TRP.1/Admin	The TOE implements a SSH Server which allows the administrator to connect to the TOE using SSH from a remote management station. The remote administration uses Port 22. The administrator is authenticated using a username and password, and if the authentication is successful the TOE establishes a SSH connection with the management workstation. Once successfully established, the administrator accesses the CLI of the TOE from the remote management workstation and each command and response thereof is protected by SSH. The SSH connection is rekeyed if the rekeying threshold is reached, and the session is terminated if the idle timer maximum value is reached. Otherwise, the SSH connection remains until the remote administrator chooses to terminate the session.

6.3 Cryptographic Details and CAVP References

This section provides additional details on the cryptographic algorithms and protocols implemented by the TOE.

6.3.1 Zeroization of Cryptographic Keys and Critical Security Parameters

The timing and method of the zeroization of the cryptographic keys and critical security parameters (CSP) used by the TOE is given in Table 16.

Table 16 Zeroization of cryptographic keys and Critical Security Parameters

Key/CSP	Purpose	Storage Location	Method of Destruction
AES Keys	Encrypt/Decrypt operations Used to generate and verify MACs with AES as part of the CMAC algorithm.	Plaintext in RAM	Power cycle cleanse()
RSA Public Key	RSA public keys used to verify data. RSA public keys used to verify firmware upgrades.	Plaintext in RAM when loaded in the memory for use	Power cycle cleanse()
RSA Public key	SSH public key	Plaintext in a file in the ~/.ssh/ directory	Overwritten when the TOE is re-initialized and a new key generated
RSA Private Key	RSA private keys used to sign data.	Plaintext in RAM when loaded in the memory for use	Power cycle cleanse()
RSA Private Key	SSH private key	Plaintext in a file in the ~/.ssh/ directory	Overwritten when the TOE is re-initialized and a new key generated
HMAC Key	HMAC keys used to generate and verify MACs on data.	Plaintext in RAM	Power cycle cleanse()
DRBG Internal state	V and key are used as part of HMAC and CTR DRBG process. V and C are used as part of HASH DRBG process.	Plaintext in RAM	Power cycle cleanse()
DRBG Entropy	Entropy input strings used as part of the DRBG process.	Plaintext in RAM	Power cycle cleanse()

6.3.2 CAVP Certificate References

The TOE implements a rich set of cryptographic functions used to protect communications and the integrity of the security functions. Each cryptographic function of the TOE is CAVP validated. The CAVP certificate references are given in Table 17.

Table 17 CAVP Certificate References

Algorithm	Description	Modes Supported	Certificate
AES-128 AES-256	Encryption and decryption of data for SSH.	CTR, GCM	A4871

RSA-2048 RSA-4096	Digital signature computation and verification for SSH and TOE Software verification.	N/A	A4859
SHA-1	Message digest computation for NTP message authentication. Diffie-Hellman key exchange.	N/A	A4871
SHA-256	Message digest computation for digital signature computation and verification. Diffie-Hellman key exchange.	N/A	A4871
SHA-256	Verification of TOE Software Verification of the TOE Software upgrades.	N/A	A4859
SHA-512	Message digest computation for digital signature computation and verification. Protection of passwords stored by the TOE. Diffie-Hellman key exchange.	N/A	A4871
HMAC-SHA-256, HMAC-SHA-512	Message Authentication Code (MAC) computation for SSH.	N/A	A4871
DRBG	Random Bit Generation	CTR_DRBG	A4871

7 Acronyms

AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	Application Programming Interface
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFP	C Form-factor Pluggable
cPP	collaborative Protection profile
CSP	Critical Security Parameter
DH	Diffie-Hellman
EAL	Evaluation Assurance Level
ECC	Elliptic Curve Cryptosystem
ECDSA	Elliptic Curve Digital Signature
EP	Extended Package
ESP	Encapsulating Security Payload
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standard
HMAC	Keyed-Hash Message Authentication Code
IA	Identification and Authentication
ID	Identity
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
ISO	International Organization for Standardization
IT	Information Technology
Junos	Juniper Operating System
MAC	Message Authentication Code or Media Access Code
MIC	Modular Interface Card
MPC	Modular Port Concentrator
MS-MPC	MultiServices Modular Port Concentrator
NAT	Network Address Translation
NTP	Network Time Protocol

OSI	Open Systems Interconnect
PAM	Pluggable Authentication module
PFE	Packet Forwarding Engine
PIC/PIM	Physical Interface Card / Physical Interface Module
PKI	Public Key Infrastructure
PoE	Power over Ethernet
PRNG	Pseudo Random Number Generator
RADIUS	Remote Authentication Dial-In User Service
RE	Routing Engine
RFC	Request For Comments
RNG	Random Number Generator
RPM	Red Had Package Manager
RSA	Rivest-Shamir-Adleman
SA	Security Association
SFP	Small Form-factor Pluggable
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SSL	Secure Socket Layer
SSR	Session Smart Routing
TCP	Transport Control Protocol
UDP	User Datagram Protocol
WAN	Wide Area Network