ASSURANCE ACTIVITY REPORT

JUNOS OS 17.4R1 FOR SRX1500, SRX4100 AND SRX4200 SERIES

Reference	EFS-T051-AAR	Status	Draft Release
Version	1.0	Release Date	11 July 2018
Author	Dan Pitcher	Customer	Juniper Networks, Inc.
Approved By	X ______________________ 17025 Signatory		

BAE SYSTEMS

INSPIRED WORK

Table of Contents

1 INTRODUCTION

1.1 Overview

This report documents the Common Criteria NDcPP, FWcPP, VPNEP and IPSEP evaluation of the Juniper Networks, Inc. Junos OS 17.4R1 for SRX1500, SRX4100 and SRX4200 Series 17.4R1-S1 (Junos 17.4R1-S1) product.

1.2 Evaluation details

Developer	Juniper Networks, Inc. 1133 Innovation Way, Sunnyvale California 94089 United States
Sponsor	Juniper Networks, Inc. 1133 Innovation Way, Sunnyvale California 94089 United States
Evaluator	BAE Systems Lab - AISEF Level 1, 14 Childers Street, Canberra ACT 2601
Scheme	AISEP
Task ID	EFS-T051

1.3 ST configuration control identifiers

ST Title	Security Target – Junos 17.4R1-S1 for SRX1500, SRX4100 and SRX4200 Series
ST Version/Date	2.2, 19 June 2018

1.4 TOE Configuration

TOE Name	Junos OS 17.4R1 for SRX1500, SRX4100 and SRX4200 Series
TOE Version	17.4R1-S1

1.5 References

1.5.1 Requirements

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, Version 3.1, Revision 5

[2] Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, Version 3.1, Revision 5

[3] Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, version 3.1 Revision 5

[4] Common Methodology for Information Technology Security Evaluation, Evaluation methodology, Version 3.1, Revision 5

[5] collaborative Protection Profile for Network Devices (NDcPP), Version 2.0+Errata 20180314, 14 March 2018

[6] collaborative Protection Profile for Stateful Traffic Filter Firewalls (FWcPP), Version 2.0+Errata20180314, 14 March 2018

[7] Network Device Collaborative Protection Profile/Stateful Traffic Filter Firewall Collaborative Protection Profile (FWcPP) Extended Package VPN Gateway, version 2.1, 8-Mar-17

[8] Network Device Collaborative Protection Profile/Stateful Traffic Filter Firewall Collaborative Protection Profile (FWcPP) Extended Package for Intrusion Prevention Systems, Version 2.11, 8-Mar-17

1.5.2 Evaluation Evidence

[9] Security Target – Junos 17.4R1-S1 for SRX1500, SRX4100 and SRX4200 Series, Version 2.2, 19 June 2018

[10] Junos® OS Common Criteria and FIPS Evaluated Configuration Guide for SRX Series Devices, Release 17.4R1, 18 June 2018

[11] Junos® OS Intrusion Detection and Prevention Feature Guide for Security Devices, 18 June 2018

[12] Junos® OS VPN Feature Guide for Security Devices, 31-Oct-17

[13] Junos® OS CLI User Guide, 19-Sep-17

[14] Junos® OS Installation and Upgrade Guide, 30-Oct-17

[15] Junos® OS Routing Policies, Firewall Filters and Traffic Policers Feature Guide, 22-Aug-17

[16] Seeding of the Kernel RBG in vSRX and SRX TVP Appliances Running Junos 17.4R1-S1, Version 1.3, 15 June 2018

1.6 Copyright statement

2 NDCPP ASSURANCE ACTIVITIES

2.1 Security Audit (FAU)

2.1.1 FAU_GEN.1/ND Network Device Audit Data Generation

TSS	For the administrative task of generating/import of, changing, or deleting of cryptographic keys as defined in FAU_GEN.1.1c, the TSS should identify what information is logged to identify the relevant key.

Junos OS creates and stores audit records for the following events:

- Start-up and shut-down of the audit functions
- Administrative login and logout
- Configuration is committed
- Configuration is changed (includes all management activities of TSF data)
- Generating/import of, changing, or deleting of cryptographic keys
- Resetting passwords
- Starting and stopping services
- All use of the identification and authentication mechanisms
- Unsuccessful login attempts limit is met or exceeded
- Any attempt to initiate a manual update
- Result of the update attempt (success or failure)
- The termination of a local/remote/interactive session by the session locking mechanism
- Initiation/termination/failure of the SSH trusted channel to syslog server
- Initiation/termination/failure of the SSH trusted path with Admin
- Initiation/termination/failure of an IPsec trusted channel, including Session Establishment with peer
- Session establishment with CA
- Application of firewall rules configured with the ‘log’ operation by the stateful traffic filtering function
- Indication of packets dropped due to too much network traffic by the stateful traffic filtering function
- Application of rules configured with the ‘log’ operation by the packet filtering function
- Indication of packets dropped due to too much network traffic by the packet filtering function
- Start-up and shut-down of the IPS functions
- All dissimilar IPS events and reactions
- Totals of similar events and reactions occurring within a specified time period
- Modification of an IPS policy element
- Modification of which IPS policies are active on a TOE interface
- Enabling/disabling a TOE interface with IPS policies applied
- Modification of which mode(s) is/are active on a TOE interface
- Inspected traffic matches a list of known-good or known-bad addresses applied to an IPS policy
- Inspected traffic matches a signature-based IPS policy with logging enabled
- Inspected traffic matches an anomaly-based IPS policy

As a minimum, Junos OS records the following with each log entry:

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

In order to identify the key being operated on, the following details are recorded for all administrative actions relating to cryptographic keys (generating, importing, changing and deleting keys):

- PKID – certificate id will be recorded when generating or deleting a key pair
- IKE SPI – IP address of the initiator and responder recorded, together with the SPI, will be recorded when generating a key pair. The IP address of the initiator and responder will provide the unique link to the key identifier (SPI) of the key that has been destroyed in the session termination
- SSH session keys– key reference provided by process id
- SSH keys created for outbound trusted channel to external syslog server
- SSH keys imported for outbound trusted channel to external syslog server
- SSH key configured for SSH public key authentication –the hash of the public key that is to be used for authentication is recorded in syslog

Guidance	The evaluator shall check the guidance documentation and ensure that it lists all of the auditable events and provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that every audit event type mandated by the cPP is described and that the description of the fields contains the information required in FAU_GEN.1.2, and the additional information specified in the table of audit events.

The list of auditable events provided in Chapter 9 of the Evaluated Configuration Guide covers all of the auditable events listed in the ST.

Table 11 describes each of the fields contained in an audit event log. These fields are:

- Timestamp;
- Hostname;
- Process;
- Process ID;
- TAG;
- Username; and
- Message Text.

Guidance	The evaluator shall also make a determination of the administrative actions related to TSF data related to configuration changes. The evaluator shall examine the guidance documentation and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the cPP. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are related to TSF data related to configuration changes. The evaluator may perform this activity as part of the activities associated with ensuring that the corresponding guidance documentation satisfies the requirements related to it.

The Evaluated Configuration Guide (ECG) provides the CLI commands and configuration examples necessary to place the device into its evaluated configuration and to enforce the requirements specified in the Security Target.

Testing	The evaluator shall test the TOE’s ability to correctly generate audit records by having the TOE generate audit records for the events listed in the table of audit events and administrative actions listed above. This should include all instances of an event: for instance, if there are several different I&A mechanisms for a system, the FIA_UIA_EXT.1 events must be generated for each mechanism. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. If HTTPS is implemented, the test demonstrating the establishment and termination of a TLS session can be combined with the test for an HTTPS session. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the guidance documentation, and that the fields in each audit record have the proper entries.

The evaluators, throughout testing, examined audit logs generated by the TOE. The evaluators confirmed that the TOE generated audit logs for each auditable event and administrative action (including those for each I&A mechanism and trusted channel) specified in this requirement.

2.1.2 FAU_GEN.1/IPS IPS Audit Data Generation

TSS	The evaluator shall verify that the TSS describes how the TOE can be configured to log IPS data associated with applicable policies.

An IDP policy is made up of rule bases, and each rule base contains a set of rules that specify rule parameters, such as traffic match conditions, action, and logging requirements.

TSS	The evaluator shall verify that the TSS describes what (similar) IPS event types the TOE will combine into a single audit record along with the conditions (e.g., thresholds and time periods) for so doing. The TSS shall also describe to what extent (if any) that may be configurable.

Because of the nature of IPS event logs, log generation often happens in bursts and can generate a large volume of messages during an attack. To manage the volume of log messages, Junos supports log suppression, which suppresses multiple instances of the same log occurring from the same or similar sessions over the same period of time.

IPS log suppression is enabled by default and can be customized based on the following configurable attributes:

- Source/destination addresses;
- Number of log occurrences after which log suppression begins;

- Maximum number of logs that log suppression can operate on;
- Time after which suppressed logs are reported.

Suppressed logs are reported as single log entries containing the count of occurrences

TSS	For IPS_SBD_EXT.1, for each field, the evaluator shall verify that the TSS describes how the field is inspected and if logging is not applicable, any other mechanism such as counting that is deployed.

Once stateful firewall processing of packets has been performed by the Information Flow subsystem, if a firewall policy that has been marked for IDP processing is triggered, the packets are processed by the IPS subsystem as follows:

- Fragmentation Processing – IP Fragments are reordered and reassembled. Duplicate, over/undersized, overlapping, incomplete and other invalid fragments are discarded.
- Flow Module SSL Decryption – sessions are checked for existing IP Actions, if none exists, new sessions are created. If a destination is marked for SSL decryption, a copy of the SSL traffic will be sent to the decryption engine. The original packet will be queue until inspection is complete.
- Packet Serialization and TCP Reassembly – packets are ordered and all TCP packets are reassembled into complete application messages.
- Application ID – pattern matching is performed on the traffic to determine what application the traffic is. The traffic is still inspected for Attacks, even if application cannot be determined.
- Protocol Decoding – protocol parsing and decoding is performed. Messages are deconstructed into application "contexts" which identify components of messages. Protocol Anomaly Detection is performed, along with AppDoS (if configured) by thresholds of these contexts.
- Attack Signature Matching – signatures are detected via deterministic finite automaton (DFA) pattern matching.

Guidance	The evaluator shall verify that the operational guidance describes how to configure the TOE to result in applicable IPS data logging.

Per the IPS Feature Guide, logging for signature-based detection is enabled by including the following configuration statement in the IDP policy:

```
set security idp idp-policy base-policy rulebase-ips rule <rule-name> then
notification log-attacks alert
```

Attack screens (such as TCP SYN-FIN or Ping-of-Death) are automatically logged.

Guidance	The evaluator shall verify that the operational guidance provides instructions for any configuration that may be done in regard to logging similar events (e.g., setting thresholds, defining time windows, etc.).

Per the IDP Feature Guide, to set a threshold to begin log suppression, the following command is used:

```
set security idp sensor-configuration log suppression start-log <total
events>
```

To set the threshold at which a log entry is generated, the following command is used:

```
set security idp sensor-configuration log suppression max-time-report
<total events>
```

Testing	The evaluator shall test that the interfaces used to configure the IPS polices yield expected IPS data in association with the IPS policies. A number of IPS policy combination and ordering scenarios need to be configured and tested by attempting to pass both allowed and anomalous network traffic matching configured IPS policies in order to trigger all required IPS events. Note that this activity should have been addressed with a combination of the Test assurance activities for the other IPS requirements.

The evaluators, throughout testing, examined audit logs generated by the TOE. The evaluators confirmed that the TOE generated audit logs for each auditable event and administrative action specified in this requirement.

2.1.3 FAU_GEN.2 User identity association

TSS	N/A

As a minimum, Junos OS records the following with each log entry:

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

Guidance	N/A

N/A

Testing	This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.

The evaluators, throughout testing, examined audit logs generated by the TOE. The evaluators confirmed that the TOE generated audit logs for each auditable event and administrative action specified in these cPPs/EPs.

2.1.4 FAU_STG_EXT.1 Protected Audit Event Storage

TSS	The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided.

Syslog can be configured to store the audit logs, and optionally to send them to one or more syslog log servers via Netconf over SSH.

TSS	The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally; what happens when the local audit data store is full; and how these records are protected against unauthorized access.

The Junos OS defines an active log file and a number of “archive” files (10 by default, but configurable from 1 to 1000). When the active log file reaches its maximum size, the logging utility closes the file, compresses it, and names the compressed archive file ‘logfile.0.gz’. The logging utility then opens and writes to a new active log file. When the new active log file reaches the configured maximum size, ‘logfile.0.gz’ is renamed ‘logfile.1.gz’, and the active log file is closed, compressed, and renamed ‘logfile.0.gz’.

When the maximum number of archive files is reached and when the size of the active file reaches the configured maximum size, the contents of the oldest archived file are deleted so the current active file can be archived

TSS	The evaluator shall examine the TSS to ensure that it details the behaviour of the TOE when the storage space for audit data is full. When the option 'overwrite previous audit record' is selected this description should include an outline of the rule for overwriting audit data. If 'other actions' are chosen such as sending the new audit data to an external IT entity, then the related behaviour of the TOE shall also be detailed in the TSS.

A 1GB syslog file takes approximately 0.25Gb of storage when archived. Syslog files can acquire complete storage allocated to /var filesystem, which is platform specific. However, when the filesystem reaches 92% storage capacity an event is raised to the administrator but the eventd process (being a privileged process) still can continue using the reserved storage blocks. This allows the syslog to continue storing events while the administrator frees the storage.

If the administrator does not free the storage in time and the /var filesystem storage becomes exhausted a final entry is recorded in the log reporting "No space left on device" and logging is terminated. The appliance continues to operate in the event of exhaustion of audit log storage space.

TSS	The evaluator shall examine the TSS to ensure that it details whether the transmission of audit information to an external IT entity can be done in real-time or periodically.

Syslog can be configured to store the audit logs, and optionally to send them to one or more syslog log servers via Netconf over SSH.

Guidance	The evaluator shall also examine the guidance documentation to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server.

Chapter 5 (Configuring the Remote Syslog Server) details the use of a NETCONF-enabled remote server for syslog transmission. This includes the generation of public/private key pairs, configuration of user accounts for syslog transmission and configuring NETCONF on the server.

Guidance	The evaluator shall also examine the guidance documentation to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server. For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and "cleared" periodically by sending the data to the audit server.

The ECG provides the following information relevant to this requirement:

"*When the device running Junos OS is set up for an external syslog server, the TOE forwards copies of local logs to the external syslog server and retains local copies of all logs when the TOE is configured in event log mode. In stream log mode, all logs except traffic logs are stored locally and can be forwarded to an external syslog server, whereas traffic logs can only be forwarded to an external syslog server*".

Guidance	The evaluator shall also ensure that the guidance documentation describes all possible configuration options for FAU_STG_EXT.1.3 and the resulting behaviour of the TOE for each possible configuration. The description of possible configuration options and resulting behaviour shall correspond to those described in the TSS.

Per the ST, when log storage is full the oldest log entries are overwritten to allow for storage of new events. This option is non-configurable and performed automatically.

Testing	The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing. The evaluator shall verify that the TOE is capable of transferring audit data to an external audit server automatically without administrator intervention.

Evaluators established an SSH NETCONF session between a remote audit server and the TOE, per the Evaluated Configuration Guide.

Evaluators, via Wireshark analysis, confirmed that all audit traffic transmitted between the TOE and audit server was encrypted and was not viewable in plaintext.

The remote audit server was using Syslog v8.15.0-1 (pre-installed within Kali Linux) as its syslog platform.

Evaluators confirmed that the TOE automatically transmitted audit logs upon establishment of the SSH NETCONF session and this did not require any intervention by the administrator.

Testing	The evaluator shall perform operations that generate audit data and verify that this data is stored locally. The evaluator shall perform operations that generate audit data until the local storage space is exceeded and verifies that the TOE complies with the behaviour defined in FAU_STG_EXT.1.3. Depending on the configuration this means that the evaluator has to check the content of the audit data when the audit data is just filled to the maximum and then verifies that: 1. The audit data remains unchanged with every new auditable event that should be tracked but that the audit data is recorded again after the local storage for audit data is cleared (for the option 'drop new audit data' in FAU_STG_EXT.1.3); 2. The existing audit data is overwritten with every new auditable event that should be tracked according to the specified rule (for the option 'overwrite previous audit records' in FAU_STG_EXT.1.3); or 3. The TOE behaves as specified (for the option 'other action' in FAU_STG_EXT.1.3).

The evaluators generated audit data and confirmed that these audit files were stored within the TOE file system.

The evaluators confirmed that, upon exhausting the local storage space, the TOE deleted the oldest log file and created a new file to write to. This behaviour is consistent with FAU_STG_EXT.1.

Testing	If the TOE complies with FAU_STG_EXT.2/LocSpace the evaluator shall verify that the numbers provided by the TOE according to the selection for FAU_STG_EXT.2/LocSpace are correct when performing the tests for FAU_STG_EXT.1.3

The TOE does not claim compliance with FAU_STG_EXT.2/LocSpace.

Testing	For distributed TOEs, Test 1 defined above should be applicable to all TOE components that forward audit data to an external audit server. For the local storage according to FAU_STG_EXT.1.2 and FAU_STG_EXT.1.3 the Test 2 specified above shall be applied to all TOE components that store audit data locally. For all TOE components that store audit data locally and comply with FAU_STG_EXT.2/LocSpace Test 3 specified above shall be applied. The evaluator shall verify that the transfer of audit data to an external audit server is implemented.

The TOE is not distributed and, as such, this test was not applicable.

2.1.5 FAU_STG.1 Protected audit trail storage

TSS	The evaluator shall ensure that the TSS identifies how IPS data is protected from unauthorized modification and deletion.

Local audit log are stored in /var/log/ in the underlying filesystem. Only a Security Administrator can read log files, or delete log and archive files through the CLI interface or through direct access to the filesystem having first authenticated as a Security Administrator.

Guidance	The evaluator shall examine the guidance documentation to determine that it describes any configuration required for protection of the locally stored audit data against unauthorized modification or deletion.

No configuration is required to protect locally stored audit data – this is done automatically by the TOE.

Testing	The evaluator shall access the audit trail without authentication as Security Administrator (either by authentication as a non-administrative user, if supported, or without authentication at all) and attempt to modify and delete the audit records. The evaluator shall verify that these attempts fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to access the audit trail can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

The TOE defines a single role, that of the Security Administrator.

Evaluators confirmed that as no access to TSF data or services is available prior to authentication, no unauthorised access to the audit trail is permitted.

Testing	The evaluator shall access the audit trail as an authorized administrator and attempt to delete the audit records. The evaluator shall verify that these attempts succeed. The evaluator shall verify that only the records authorized for deletion are deleted.

Evaluators accessed the TOE via the CLI as an authorised administrator and requested that the TOE delete a single log file. Evaluators confirmed that the specified log file (and only this single file) was deleted from the TOE file system.

2.2 Cryptographic Support (FCS)

2.2.1 FCS_CKM.1/ND Cryptographic Key Generation

TSS	The evaluator shall ensure that the TSS identifies the key sizes supported by the TOE. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.

The module implements the following key generation methods:

- RSA (2048, 4096); and
- ECDSA (P-256, P-384 and P-521).

Table 8 indicates which keys are used for which protocols.

Guidance	The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key generation scheme(s) and key size(s) for all cryptographic protocols defined in the Security Target.

The Evaluated Configuration Guide (ECG) describes how the administrator can configure SSH and IPsec. As part of these configuration guides, the available cryptographic methods and associated key sizes are indicated with configuration examples for how to set these values appropriately.

Testing	**Key Generation for FIPS PUB 186-4 RSA Schemes** The evaluator shall verify the implementation of RSA Key Generation by the TOE using the Key Generation test. This test verifies the ability of the TSF to correctly produce values for the key components including the public verification exponent e, the private prime factors p and q, the public modulus n and the calculation of the private signature exponent d.

The key generation functionality provided by the TOE was tested via CAVS/the CAVP. The following algorithm certificates have been assigned to this implementation:

- **RSA**: #2838, #2839, #2842 and #2843

Testing	**Key Generation for Elliptic Curve Cryptography (ECC)** FIPS 186-4 ECC Key Generation Test For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall require the implementation under test (IUT) to generate 10 private/public key pairs. The private key shall be generated using an approved random bit generator (RBG). To determine correctness, the evaluator shall submit the generated key pairs to the public key verification (PKV) function of a known good implementation. FIPS 186-4 Public Key Verification (PKV) Test For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall generate 10 private/public key pairs using the key generation function of a known good implementation and modify five of the public key values so that they are incorrect, leaving five values unchanged (i.e., correct). The evaluator shall obtain in response a set of 10 PASS/FAIL values.

The key generation functionality provided by the TOE was tested via CAVS/the CAVP. The following algorithm certificates have been assigned to this implementation:

- **ECDSA**: #1388, #1389, #1392 and #1393

2.2.2 FCS_CKM.1/IKE Cryptographic Key Generation (for IKE Peer Authentication)

TSS	The evaluator shall check to ensure that the TSS describes how the key-pairs are generated.

Asymmetric keys are generated in accordance with NIST SP 800-56A and FIPS PUB 186-3 for IKE with IPSec. The TOE complies with section 5.6 of NIST SP 800-56A regarding asymmetric key pair generation.

TSS	In order to show that the TSF implementation complies with FIPS PUB 186-4, the evaluator shall ensure that the TSS contains the following information: • The TSS shall list all sections of Appendix B to which the TOE complies. • For each applicable section listed in the TSS, for all statements that are not "shall" (that is, "shall not", "should", and "should not"), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as "shall not" or "should not" in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE; • For each applicable section of Appendix B, any omission of functionality related to "shall" or "should" statements shall be described; Any TOE-specific extensions, processing that is not included in the Appendices, or alternative implementations allowed by the Appendices that may impact the security requirements the TOE is to enforce shall be described.

The TOE implements all of the "shall" and "should" requirements and none of the "shall not" or "should not" from FIPS PUB 186-4 Appendix B3 and B4.

Guidance	The evaluator shall check that the operational guidance describes how the key generation functionality is invoked, and describes the inputs and outputs associated with the process for each signature scheme supported. The evaluator shall also check that guidance is provided regarding the format and location of the output of the key generation process.

Per the CLI Guide, keys can be generated using the following command:

```
request security pki generate-key-pair certificate-id <certificate-id-
name> <size (256 | 384 | 1024 | 2048 | 4096)> <type (dsa | ecdsa | rsa)>
```

Testing	The evaluator shall use the key pair generation portions of "The FIPS 186-4 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)" and "The RSA Validation System (RSA2VS)" as a guide in testing the requirement above, depending on the selection performed by the ST author. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.

The key generation functionality provided by the TOE was tested via CAVS/the CAVP. The following algorithm certificates have been assigned to this implementation:

- **RSA**: #2838, #2839, #2842 and #2843
- **ECDSA**: #1388, #1389, #1392 and #1393

2.2.3 FCS_CKM.2 Cryptographic Key Establishment

TSS	The evaluator shall ensure that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme (including whether the TOE acts as a sender, a recipient, or both).

Asymmetric keys are also generated in accordance with FIPS PUB 186-4 Appendix B.3 for RSA Schemes and Appendix B.4 for ECC Schemes for SSH communications.

The TOE acts as both sender and recipient for IPsec and only as the server for SSH in the supported protocols listed in Table 8.

TSS	If Diffie-Hellman group 14 is selected from FCS_CKM.2.1, the TSS shall describe how the implementation meets RFC 3526 Section 3.

The TOE implements Diffie-Hellman group 14, using the modulus and generator specified by Section 3 of RFC3526.

Guidance	The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key establishment scheme(s).

The Evaluated Configuration Guide (ECG) describes how the administrator can configure SSH and IPsec. As part of these configuration guides, the available cryptographic methods and associated key sizes are indicated with configuration examples for how to set these values appropriately.

Testing	The evaluator shall verify a TOE's implementation of SP800-56A key agreement schemes using the following Function and Validity tests. These validation tests for each key agreement scheme verify that a TOE has implemented the components of the key agreement scheme according to the specifications in the Recommendation. These components include the calculation of the DLC primitives (the shared secret value Z) and the calculation of the derived keying material (DKM) via the Key Derivation Function (KDF). If key confirmation is supported, the evaluator shall also verify that the components of key confirmation have been implemented correctly, using the test procedures described below. This includes the parsing of the DKM, the generation of MACdata and the calculation of MACtag.

The key establishment functionality provided by the TOE was tested via CAVS/the CAVP. The following certificates have been assigned to this implementation:

- **CVL**: #1769 and #1770

Testing	The evaluator shall verify the correctness of the TSF's implementation of Diffie-Hellman group 14 by using a known good implementation for each protocol selected in FTP_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and FPT_ITT.1 that uses Diffie-Hellman group 14

The key establishment functionality provided by the TOE was tested via CAVS/the CAVP. The following certificates have been assigned to this implementation:

- **CVL**: #1769 and #1770

2.2.4 FCS_CKM.4 Cryptographic Key Destruction

TSS	The evaluator examines the TSS to ensure it lists all relevant keys (describing the origin and storage location of each), all relevant key destruction situations (e.g. factory reset or device wipe function, disconnection of trusted channels, key change as part of a secure channel protocol), and the destruction method used in each case. For the purpose of this Evaluation Activity the relevant keys are those keys that are relied upon to support any of the SFRs in the Security Target.

Table 9 lists all keys/CSPs applicable to the module and their:

- CSP;
- Description;
- Method of storage;
- Storage location; and
- Zeroization method.

TSS	The evaluator confirms that the description of keys and storage locations is consistent with the functions carried out by the TOE (e.g. that all keys for the TOE-specific secure channels and protocols, or that support FPT_APW.EXT.1 and FPT_SKP_EXT.1, are accounted for). In particular, if a TOE claims not to store plaintext keys in non-volatile memory then the evaluator checks that this is consistent with the operation of the TOE.

Table 9 lists each key/CSP used by the module, its storage location and storage method.

The keys/CSPs listed are consistent with the operation of the TOE.

TSS	The evaluator shall check to ensure the TSS identifies how the TOE destroys keys stored as plaintext in non-volatile memory, and that the description includes identification and description of the interfaces that the TOE uses to destroy keys (e.g., file system APIs, key store APIs).

Table 9 lists each key/CSP used by the module, its storage location and storage method.

The zeroisation method invoked for each key/CSP is described (e.g. “‘clear security IKE securityassociation’ command or reboot the box.”)

TSS	Where the TSS identifies keys that are stored in a non-plaintext form, the evaluator shall check that the TSS identifies the encryption method and the key-encrypting-key used, and that the key-encrypting-key is either itself stored in an encrypted form or that it is destroyed by a method included under FCS_CKM.4.

Table 9 lists each key/CSP used by the module, its storage location and storage method.

The method used to protect non-plaintext keys is described (e.g. “Hashed when stored (sha-256)”.

TSS	The evaluator shall check that the TSS identifies any configurations or circumstances that may not conform to the key destruction requirement. Note that reference may be made to the Guidance Documentation for description of the detail of such cases where destruction may be prevented or delayed.

No configurations or circumstances relevant to this requirement are described.

Guidance	A TOE may be subject to situations that could prevent or delay key destruction in some cases. The evaluator shall check that the guidance documentation identifies configurations or circumstances that may not strictly conform to the key destruction requirement, and that this description is consistent with the relevant parts of the TSS (and any other supporting information used). The evaluator shall check that the guidance documentation provides guidance on situations where key destruction may be delayed at the physical layer. For example, when the TOE does not have full access to the physical memory, it is possible that the storage may be implementing wear-levelling and garbage collection. This may result in additional copies of the key that are logically inaccessible but persist physically. Where available, the TOE might then describe use of the TRIM command3 and garbage collection to destroy these persistent copies upon their deletion (this would be explained in TSS and Operational Guidance).

Chapter 3 of the ECG provides instructions on how to perform zeroisation of the TOE (`request system zeroise`). There are no instances where key destruction may be delayed at the physical layer.

Testing	N/A

N/A

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

TSS	N/A

The TOE utilises AES in CBC, GCM and CTR modes with 128-bit, 192-bit and 256-bit keys.

Guidance	N/A

N/A

Testing	There are four Known Answer Tests (KATs) described for this requirement. In all KATs, the plaintext, ciphertext, and IV values shall be 128-bit blocks. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

The AES implementation used by the TOE was tested via CAVS/the CAVP. The following certificates have been assigned to this implementation:

- **AES**: #5303, #5304, #5307, #5308, #5339 and #5340

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

TSS	N/A

The TOE provides signature generation and verification services using RSA (2048/4096) and ECDSA (P-256, P-384 and P-521).

Guidance	N/A

N/A

Testing	**ECDSA Algorithm Tests** ECDSA FIPS 186-4 Signature Generation Test For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate 10 1024-bit long messages and obtain for each message a public key and the resulting signature values R and S. To determine correctness, the evaluator shall use the signature verification function of a known good implementation.

The ECDSA implementation used by the TOE was tested via CAVS/the CAVP. The following certificates have been assigned to this implementation:

- **ECDSA**: #1388, #1389, #1392 and #1393

Testing	**ECDSA Algorithm Tests** ECDSA FIPS 186-4 Signature Verification Test For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate a set of 10 1024-bit message, public key and signature tuples and modify one of the values (message, public key or signature) in five of the 10 tuples. The evaluator shall obtain in response a set of 10 PASS/FAIL values.

The ECDSA implementation used by the TOE was tested via CAVS/the CAVP. The following certificates have been assigned to this implementation:

- **ECDSA**: #1388, #1389, #1392 and #1393

Testing	**RSA Signature Algorithm Tests** Signature Generation Test The evaluator generates or obtains 10 messages for each modulus size/SHA combination supported by the TOE. The TOE generates and returns the corresponding signatures. The evaluator shall verify the correctness of the TOE's signature using a trusted reference implementation of the signature verification algorithm and the associated public keys to verify the signatures.

The RSA implementation used by the TOE was tested via CAVS/the CAVP. The following certificates have been assigned to this implementation:

- **RSA**: #2838, #2839, #2842 and #2843

Testing	**RSA Signature Algorithm Tests** Signature Verification Test For each modulus size/hash algorithm selected, the evaluator generates a modulus and three associated key pairs, (d, e). Each private key d is used to sign six pseudorandom messages each of 1024 bits using a trusted reference implementation of the signature generation algorithm. Some of the public keys, e, messages, or signatures are altered so that signature verification should fail. For both the set of original messages and the set of altered messages: the modulus, hash algorithm, public key e values, messages, and signatures are forwarded to the TOE, which then attempts to verify the signatures and returns the verification results. The evaluator verifies that the TOE confirms correct signatures on the original messages and detects the errors introduced in the altered messages.

The RSA implementation used by the TOE was tested via CAVS/the CAVP. The following certificates have been assigned to this implementation:

- **RSA**: #2838, #2839, #2842 and #2843

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

TSS	The evaluator shall check that the association of the hash function with other TSF cryptographic functions (for example, the digital signature verification function) is documented in the TSS.

Table 7 provides a mapping between the implemented hash functions and other functions.

For example, the listing for signature verification using RSA and ECDSA indicate which hash functions (SHA-256 or SHA-384) are used with each key length or curve.

Guidance	The evaluator checks the AGD documents to determine that any configuration that is required to configure the required hash sizes is present.

The Evaluated Configuration Guide (ECG) describes how the administrator can configure SSH and IPsec. As part of these configuration guides, the available cryptographic methods and associated hash sizes are indicated with configuration examples for how to set these values appropriately.

Testing	**Short Messages Test - Bit-oriented Mode** The evaluators devise an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

The SHA implementation used by the TOE was tested via CAVS/the CAVP. The following certificates have been assigned to this implementation:

- **SHA**: #4255, #4257, #4261, #4262, #4265, #4266, #4290 and #4291

Testing	**Short Messages Test - Byte-oriented Mode** The evaluators devise an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

The SHA implementation used by the TOE was tested via CAVS/the CAVP. The following certificates have been assigned to this implementation:

- **SHA**: #4255, #4257, #4261, #4262, #4265, #4266, #4290 and #4291

Testing	**Selected Long Messages Test - Bit-oriented Mode** The evaluators devise an input set consisting of m messages, where m is the block length of the hash algorithm (e.g. 512 bits for SHA-256). The length of the ith message is m + 99*i, where 1 ≤ i ≤ m. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

The SHA implementation used by the TOE was tested via CAVS/the CAVP. The following certificates have been assigned to this implementation:

- **SHA**: #4255, #4257, #4261, #4262, #4265, #4266, #4290 and #4291

Testing	**Selected Long Messages Test - Byte-oriented Mode** The evaluators devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm (e.g. 512 bits for SHA-256). The length of the ith message is m + 8*99*i, where 1 ≤ i ≤ m/8. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

The SHA implementation used by the TOE was tested via CAVS/the CAVP. The following certificates have been assigned to this implementation:

- **SHA**: #4255, #4257, #4261, #4262, #4265, #4266, #4290 and #4291

Testing	**Pseudorandomly Generated Messages Test** This test is for byte-oriented implementations only. The evaluators randomly generate a seed that is n bits long, where n is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of [SHAVS]. The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.

The SHA implementation used by the TOE was tested via CAVS/the CAVP. The following certificates have been assigned to this implementation:

- **SHA**: #4255, #4257, #4261, #4262, #4265, #4266, #4290 and #4291

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

TSS	The evaluator shall examine the TSS to ensure that it specifies the following values used by the HMAC function: key length, hash function used, block size, and output MAC length used.

Table 7 lists the supported HMAC functions, lengths, has functions, block sizes and output MACs.

Guidance	N/A

N/A

Testing	For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key and message data using a known good implementation.

The SHA implementation used by the TOE was tested via CAVS/the CAVP. The following certificates have been assigned to this implementation:

- **HMAC**: #3507, #3508, #3511, #3512, #3536 and #3537

2.2.9 FCS_RBG_EXT.1 Random Bit Generation

TSS	The evaluator shall examine the TSS to determine that it specifies the DRBG type, identifies the entropy source(s) seeding the DRBG, and state the assumed or calculated min-entropy supplied either separately by each source or the min-entropy contained in the combined seed value.

Junos OS performs random bit generation in accordance with NIST Special Publication 800-90 using HMAC_DRBG, SHA-256. The RBG for the (virtualised) SRX SRX1500, SRX4100 and SRX4200 Series platforms is seeded by hardware-based noise sources of entropy, namely RANDOM_INTERRUPT, RANDOM_PURE_RDRAND (RDRAND) and RANDOM_NET.

- RANDOM_INTERRUPT: This source of entropy is provided by devices whose hardware interrupts are known to provide some amount of entropy, such as hard drive controllers. The timings are fed into kernel HMAC DRBG (Juniper kernel DRBG) along with a CPU cycle counter;
- RANDOM_PURE_RDRAND: This hardware source of entropy provides 8 bytes of RDRAND40 along with a CPU cycle counter are fed into kernel HMAC DRBG (Juniper kernel DRBG); and
- RANDOM_NET: This source of entropy is associated with network activity. Timings (CPU counter values at the time of the event) together with internal representation of the network packets are used to construct extra entropy that is fed in to kernel HMAC DRBG.

Guidance	The evaluator shall confirm that the guidance documentation contains appropriate instructions for configuring the RNG functionality.

The DRBG utilised by the TOE is non-configurable by the Administrator and is automatically used by the TOE.

Testing	The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator shall perform 15 trials for each configuration. If the RNG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. "generate one block of random bits" means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP800-90A). If the RNG does not have prediction resistance, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call.

The DRBG implementation used by the TOE was tested via CAVS/the CAVP. The following certificates have been assigned to this implementation:

- **DRBG**: #2039, #2041, #2042, #2043, #2046 and #2047

2.2.10 FCS_IPSEC_EXT.1 IPsec Protocol

2.2.10.1 FCS_IPSEC_EXT.1.1

TSS	The evaluator shall examine the TSS and determine that it describes what takes place when a packet is processed by the TOE, e.g., the algorithm used to process the packet.

Each packet is compared against entries in the security policy ruleset in sequential order until one is found that matches the specification in the policy, or until the end of the rule set is reached, in which case the implicit default policy is implemented and the packet is discarded.

TSS	The TSS describes how the SPD is implemented and the rules for processing both inbound and outbound packets in terms of the IPsec policy. The TSS describes the rules that are available and the resulting actions available after matching a rule. The TSS describes how those rules and actions form the SPD in terms of the BYPASS (e.g., no encryption), DISCARD (e.g., drop the packet), and PROTECT (e.g., encrypt the packet) actions defined in RFC 4301.

The security policy rule set is an ordered list of security policy entries, each of which contains the specification of a network flow and an action:

- Source IP address and network mask
- Destination IP address and network mask
- Protocol
- Source port
- Destination port
- Action: permit, deny, drop silently, log

Each packet is compared against entries in the security policy ruleset in sequential order until one is found that matches the specification in the policy, or until the end of the rule set is reached, in which case the implicit default policy is implemented and the packet is discarded.

Guidance	The evaluator shall examine the guidance documentation to verify it instructs the Administrator how to construct entries into the SPD that specify a rule for processing a packet. The description includes all three cases – a rule that ensures packets are encrypted/decrypted, dropped, and flow through the TOE without being encrypted. The evaluator shall determine that the description in the guidance documentation is consistent with the description in the TSS, and that the level of detail in the guidance documentation is sufficient to allow the administrator to set up the SPD in an unambiguous fashion. This includes a discussion of how ordering of rules impacts the processing of an IP packet.

The IPsec VPN chapter of the Evaluated Configuration Guide provides administrators with guidance on how to construct security flow policies with the Bypass (permit), Discard (drop) and Protect (VPN) operations. The provided guidance is sufficiently detailed, with accompanying configuration examples.

Testing	The evaluator shall configure the SPD such that there is a rule for dropping a packet, encrypting a packet, and allowing a packet to flow in plaintext. The selectors used in the construction of the rule shall be different such that the evaluator can generate a packet and send packets to the gateway with the appropriate fields (fields that are used by the rule - e.g., the IP addresses, TCP/UDP ports) in the packet header. The evaluator performs both positive and negative test cases for each type of rule (e.g. a packet that matches the rule and another that does not match the rule). The evaluator observes via the audit trail, and packet captures that the TOE exhibited the expected behaviour: appropriate packets were dropped, allowed to flow without modification, encrypted by the IPsec implementation.

Evaluators confirmed that:

- The TOE permits the creation of security policies with PROTECT, REJECT and BYPASS functionality; and
- The TOE logs traffic as appropriate when each rule is met.

Testing	The evaluator shall devise several tests that cover a variety of scenarios for packet processing. As with Test 1, the evaluator ensures both positive and negative test cases are constructed. These scenarios must exercise the range of possibilities for SPD entries and processing modes as outlined in the TSS and guidance documentation. Potential areas to cover include rules with overlapping ranges and conflicting entries, inbound and outbound packets, and packets that establish SAs as well as packets that belong to established SAs. The evaluator shall verify, via the audit trail and packet captures, for each scenario that the expected behavior is exhibited, and is consistent with both the TSS and the guidance documentation.

Evaluators confirmed that:

- The TOE permits the creation of security policies with PROTECT, REJECT and BYPASS functionality;
- The TOE applies security policies based on the order in which they are entered, regardless of specificity;
- The TOE logs traffic as appropriate when each rule is met.

2.2.10.2 FCS_IPSEC_EXT.1.2

TSS	N/A

By default, the TOE denies all traffic through an SRX Series device. In fact, an implicit default security policy exists that denies all packets. You can change this behavior by configuring a standard security policy that permits certain types of traffic. The implicit default policy can be changed to permit all traffic with the 'set security policies default-policy' command; however, this is not recommended.

Guidance	N/A

N/A

Testing	The evaluator shall configure the SPD such that there is a rule for dropping a packet, encrypting a packet, and allowing a packet to flow in plaintext. The evaluator may use the SPD that was created for verification of FCS_IPSEC_EXT.1.1. The evaluator shall construct a network packet that matches the rule to allow the packet to flow in plaintext and send that packet. The evaluator should observe that the network packet is passed to the proper destination interface with no modification. The evaluator shall then modify a field in the packet header; such that it no longer matches the evaluator-created entries (there may be a "TOE created" final entry that discards packets that do not match any previous entries). The evaluator sends the packet, and observes that the packet was dropped.

Evaluators confirmed that:

- The TOE permits the creation of security policies with PROTECT, REJECT and BYPASS functionality;
- The TOE applies security policies based on the order in which they are entered, regardless of specificity;
- The TOE drops network traffic that does not match any administrator-defined rules; and
- The TOE logs traffic as appropriate when each rule is met.

2.2.10.3 FCS_IPSEC_EXT.1.3

TSS	The evaluator checks the TSS to ensure it states that the VPN can be established to operate in transport mode and/or tunnel mode (as identified in FCS_IPSEC_EXT.1.3).

The TOE supports tunnel mode only.

Guidance	The evaluator shall confirm that the guidance documentation contains instructions on how to configure the connection in each mode selected.

The TOE supports tunnel mode for IPsec only – this mode is selected by default and does not require configuration by the Administrator.

Testing	If tunnel mode is selected, the evaluator uses the guidance documentation to configure the TOE to operate in tunnel mode and also configures a VPN peer to operate in tunnel mode. The evaluator configures the TOE and the VPN peer to use any of the allowable cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator shall then initiate a connection from the TOE to connect to the VPN peer. The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using the tunnel mode

The evaluators configured an IPsec connection between the TOE and a peer device.

Evaluators confirmed that the connection was established in tunnel mode.

Both the audit trail and console output confirmed that the connection was established in tunnel mode (example output as follows):

```
junos_vpn: child: 10.0.4.0/24 === 10.0.2.0/24 TUNNEL
junos_vpn[1]: INSTALLED, TUNNEL, ESP SPIs: […]
```

2.2.10.4 FCS_IPSEC_EXT.1.4

TSS	The evaluator shall examine the TSS to verify that the selected algorithms are implemented. In addition, the evaluator ensures that the SHA-based HMAC algorithm conforms to the algorithms specified in FCS_COP.1/KeyedHash Cryptographic Operations (for keyed-hash message authentication).

The TOE supports AES-GCM-128, AES-GCM-192 and AES-GCM-256, and AES-CBC-128, AES-CBC-192 or AES-CBC-256 using HMAC SHA-256 for ESP protection.

Guidance	The evaluator checks the guidance documentation to ensure it provides instructions on how to configure the TOE to use the algorithms selected.

Per the Evaluated Configuration Guide, the administrator can set the Phase 2 (ESP) encryption algorithms using the following command:

```
set security proposal ipsec-proposal1 encryption-algorithm aes-256-
cbc
```

The "aes-256-cbc" can be replaced with the other algorithms specified in the requirement.

Testing	The evaluator shall configure the TOE as indicated in the guidance documentation configuring the TOE to use each of the supported algorithms, attempt to establish a connection using ESP, and verify that the attempt succeeds.

The TOE supports the use of each algorithm specified in this requirement for ESP.

The supported algorithms for ESP are AES-CBC-128, AES-CBC-192, AES-CBC-256, HMAC-SHA-256, AES-GCM-128, AES-GCM-192 and AES-GCM-256.

2.2.10.5 FCS_IPSEC_EXT.1.5

TSS	The evaluator shall examine the TSS to verify that IKEv1 and/or IKEv2 are implemented.

IKEv1 and IKEv2 are implemented. IKEv1 as defined in RFCs 2407, 2408, 2409, RFC 4109 and RFC 4868 for hash functions; IKEv2 as defined in RFCs 5996 (with no support for NAT traversal) and RFC 4868 for hash functions.

TSS	For IKEv1 implementations, the evaluator shall examine the TSS to ensure that, in the description of the IPsec protocol, it states that aggressive mode is not used for IKEv1 Phase 1 exchanges, and that only main mode is used. It may be that this is a configurable option.

IKEv1 aggressive mode is not supported.

Guidance	The evaluator shall check the guidance documentation to ensure it instructs the administrator how to configure the TOE to use IKEv1 and/or IKEv2 (as selected), and how to configure the TOE to perform NAT traversal (if selected).

The TOE uses IKEv1 by default for configured IPsec VPNs. To configure a proposal to use IKEv2, the following command can be used:

```
set security ike gateway <gw-name> version v2-only
```

Guidance	If the IKEv1 Phase 1 mode requires configuration of the TOE prior to its operation, the evaluator shall check the guidance documentation to ensure that instructions for this configuration are contained within that guidance.

The TOE supports Main mode only – this can be configured via the following command:

```
set security ike policy ike-policy1 mode main
```

Testing	If IKEv1 is selected, the evaluator shall configure the TOE as indicated in the guidance documentation, and attempt to establish a connection using an IKEv1 Phase 1 connection in aggressive mode. This attempt should fail. The evaluator should then show that main mode exchanges are supported.

Evaluators attempted to establish an IPsec connection using aggressive mode. These attempts failed.

Attempts to establish connections in main mode were accepted and connections established.

Testing	If NAT traversal is selected within the IKEv2 selection, the evaluator shall configure the TOE so that it will perform NAT traversal processing as described in the TSS and RFC 5996, section 2.23. The evaluator shall initiate an IPsec connection and determine that the NAT is successfully traversed.

The TOE does not support NAT traversal.

2.2.10.6 FCS_IPSEC_EXT.1.6

TSS	The evaluator shall ensure the TSS identifies the algorithms used for encrypting the IKEv1 and/or IKEv2 payload, and that the algorithms chosen in the selection of the requirement are included in the TSS discussion.

The TOE supports AES-CBC-128, AES-CBC-192, AES-CBC-256, AES-GCM-128 and AES-GCM-256 for payload protection in IKEv1 and IKEv2.

Guidance	The evaluator ensures that the guidance documentation describes the configuration of all selected algorithms in the requirement.

Per the Evaluated Configuration Guide, the administrator can set the Phase 1 (IKE) encryption algorithms using the following command:

```
set proposal ike-proposal1 encryption-algorithm aes-256-cbc
```

The “aes-256-cbc” can be replaced with the other algorithms specified in the requirement.

Testing	The evaluator shall configure the TOE to use the ciphersuite under test to encrypt the IKEv1 and/or IKEv2 payload and establish a connection with a peer device, which is configured to only accept the payload encrypted using the indicated ciphersuite. The evaluator will confirm the algorithm was that used in the negotiation.

Evaluators configured the TOE to use each algorithm specified in this requirement in turn. In each case, evaluators confirmed (via audit log and console output) that the selected algorithm was used.

2.2.10.7 FCS_IPSEC_EXT.1.7

TSS	The evaluator shall ensure the TSS identifies the lifetime configuration method used for limiting the IKEv1 Phase 1 SA lifetime and/or the IKEv2 SA lifetime. The evaluator shall verify that the selection made here corresponds to the selection in FCS_IPSEC_EXT.1.5.

In the evaluated configuration, the TOE permits configuration of the IKE and IPsec lifetime exchanges in terms of number of kilobytes (64 to 4294967294 kilobytes) or length of time (180 to 86400 seconds):

Guidance	The evaluator shall verify that the values for SA lifetimes can be configured and that the instructions for doing so are located in the guidance documentation. If time-based limits are supported, the evaluator ensures that the Administrator is able to configure Phase 1 SA values for 24 hours. Currently there are no values mandated for the number of bytes, the evaluator just ensures that this can be configured if selected in the requirement.

Per the VPN Feature Guide, Phase 1 SA lifetimes can be set using the following commands:

```
set security ike proposal proposal-name lifetime-seconds <seconds>
```

Where <seconds> is a value of or between 180 and 86,400.

TOE does not support byte-based lifetimes for IKE Phase 1/SA lifetimes.

Testing	If 'number of bytes' is selected as the SA lifetime measure, the evaluator shall configure a maximum lifetime in terms of the number of bytes allowed following the guidance documentation. The evaluator shall configure a test peer with a byte lifetime that exceeds the lifetime of the TOE. The evaluator shall establish an SA between the TOE and the test peer, and determine that once the allowed number of bytes through this SA is exceeded, a new SA is negotiated. The evaluator shall verify that the TOE initiates a Phase 1 negotiation.

The TOE does not support byte-based measures for Phase 1 lifetimes.

Testing	If 'length of time' is selected as the SA lifetime measure, the evaluator shall configure a maximum lifetime of 24 hours for the Phase 1 SA following the guidance documentation. The evaluator shall configure a test peer with a lifetime that exceeds the lifetime of the TOE. The evaluator shall establish an SA between the TOE and the test peer, maintain the Phase 1 SA for 24 hours, and determine that a new Phase 1 SA is negotiated on or before 24 hours has elapsed. The evaluator shall verify that the TOE initiates a Phase 1 negotiation.

The TOE established an IPsec connection between the TOE (with a configured Phase 1 lifetime of 24 hours) and a peer (with a configured Phase 1 lifetime of 25 hours). Evaluators confirmed that, after 24 hours had expired, the TOE initiated a phase 1 negotiation .

2.2.10.8 FCS_IPSEC_EXT.1.8

TSS	The evaluator shall ensure the TSS identifies the lifetime configuration method used for limiting the IKEv1 Phase 2 SA lifetime and/or the IKEv2 Child SA lifetime. The evaluator shall verify that the selection made here corresponds to the selection in FCS_IPSEC_EXT.1.5.

In the evaluated configuration, the TOE permits configuration of the IKE and IPsec lifetime exchanges in terms of number of kilobytes (64 to 4294967294 kilobytes) or length of time (180 to 86400 seconds):

Guidance	The evaluator shall verify that the values for SA lifetimes can be configured and that the instructions for doing so are located in the guidance documentation. If time-based limits are supported, the evaluator ensures that the Administrator is able to configure Phase 2 SA values for 8 hours. Currently there are no values mandated for the number of bytes, the evaluator just ensures that this can be configured if selected in the requirement.

Per the VPN Feature Guide, Phase 2 SA lifetimes can be set using the following commands:

```
set security ike proposal proposal-name lifetime-seconds <seconds>
```

Where <seconds> is a value of or between 180 and 86,400; or

```
set security ipsec proposal proposal-name lifetime-kilobytes <value>
```

Where <value> is a value of or between 64 and 1,048,576 kilobytes.

Testing	If 'number of bytes' is selected as the SA lifetime measure, the evaluator shall configure a maximum lifetime in terms of the number of bytes allowed following the guidance documentation. The evaluator shall configure a test peer with a byte lifetime that exceeds the lifetime of the TOE. The evaluator shall establish an SA between the TOE and the test peer, and determine that once the allowed number of bytes through this SA is exceeded, a new SA is negotiated. The evaluator shall verify that the TOE initiates a Phase 2 negotiation.

The TOE established an IPsec connection between the TOE (with a configured Phase 2 lifetime of 1MB) and a peer (with a configured Phase 2 lifetime of 2MB). Evaluators confirmed that, after 1MB of traffic had been transmitted, the TOE initiated a phase 2 negotiation.

Testing	If 'length of time' is selected as the SA lifetime measure, the evaluator shall configure a maximum lifetime of 8 hours for the Phase 2 SA following the guidance documentation. The evaluator shall configure a test peer with a lifetime that exceeds the lifetime of the TOE. The evaluator shall establish an SA between the TOE and the test peer, maintain the Phase 1 SA for 8 hours, and determine that once a new Phase 2 SA is negotiated when or before 8 hours has lapsed. The evaluator shall verify that the TOE initiates a Phase 2 negotiation.

The TOE established an IPsec connection between the TOE (with a configured Phase 2 lifetime of 8 hours) and a peer (with a configured Phase 2 lifetime of 10 hours). Evaluators confirmed that, after 8 hours had expired, the TOE initiated a phase 2 negotiation.

2.2.10.9 FCS_IPSEC_EXT.1.9

TSS	The evaluator shall check to ensure that, for each DH group supported, the TSS describes the process for generating "x".

The TOE uses HMAC DRBG with SHA-256 for the generation of DH exponents and nonces in the IKE key exchange protocol of length 224 bits (for DH Group 14), 256 bits (for DH Groups 19 and 24) and 384 bits (for DH Group 20).

TSS	The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of "x" meets the stipulations in the requirement.

The TOE uses HMAC DRBG with SHA-256 for the generation of DH exponents and nonces in the IKE key exchange protocol of length 224 bits (for DH Group 14), 256 bits (for DH Groups 19 and 24) and 384 bits (for DH Group 20).

Guidance	N/A

N/A

Testing	N/A

N/A

2.2.10.10 FCS_IPSEC_EXT.1.10

TSS	If the first selection is chosen, the evaluator shall check to ensure that, for each DH group supported, the TSS describes the process for generating each nonce.

The TOE uses HMAC DRBG with SHA-256 for the generation of DH exponents and nonces in the IKE key exchange protocol of length 224 bits (for DH Group 14), 256 bits (for DH Groups 19 and 24) and 384 bits (for DH Group 20).

TSS	The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of the nonces meet the stipulations in the requirement.

The TOE uses HMAC DRBG with SHA-256 for the generation of DH exponents and nonces in the IKE key exchange.

Guidance	N/A

N/A

Testing	N/A

N/A

2.2.10.11 FCS_IPSEC_EXT.1.11

TSS	The evaluator shall check to ensure that the DH groups specified in the requirement are listed as being supported in the TSS. If there is more than one DH group supported, the evaluator checks to ensure the TSS describes how a particular DH group is specified/negotiated with a peer.

The TOE supports Diffie-Hellman Groups 14, 19, 20, and 24. In the IKEv1 phase 1 and phase 2 exchanges, the TOE and peer will agree on the best DH group both can support.

When the TOE receives an IKE proposal, it will select the first DH group that matches the acceptable DH groups configured in the TOE (one or more of DH Groups 14, 19, 20 or 24) and the negotiation will fail if there is no match.

Similarly, when the peer initiates the IKE protocol, the TOE will select the first match from the IKE proposal sent by the peer and the negotiation fails is no acceptable match is found.

Guidance	The evaluator ensures that the guidance documentation describes the configuration of all algorithms selected in the requirement.

Per the Evaluated Configuration Guide, the TOE supports DH groups 14, 19, 20 and/or 24. These groups can be selected via the following commands:

IKEv1: `set proposal ike-proposal1 dh-group <dh-group>`

IKEv2: `set policy ipsec-policy1 perfect-forward-secrecy keys <dh-group>`

Testing	For each supported DH group, the evaluator shall test to ensure that all supported IKE protocols can be successfully completed using that particular DH group.

The evaluators confirmed an IPsec policy between the TOE and a peer. The evaluators confirmed that each DH group specified in this requirement (14, 19, 20 and 24) was successfully used during the negotiation.

2.2.10.12 FCS_IPSEC_EXT.1.12

TSS	The evaluator shall check that the TSS describes the potential strengths (in terms of the number of bits in the symmetric key) of the algorithms that are allowed for the IKE and ESP exchanges.

The TOE ensures that the strength of the symmetric algorithm (128 or 256 bits) negotiated to protect the IKEv1 Phase 1, IKEv2 IKE_SA connection is greater than or equal to the strength of the symmetric algorithm negotiated to protect the IKEv1 Phase 2, IKEv2 CHILD_SA connection.

TSS	The TSS shall also describe the checks that are done when negotiating IKEv1 Phase 2 and/or IKEv2 CHILD_SA suites to ensure that the strength (in terms of the number of bits of key in the symmetric algorithm) of the negotiated algorithm is less than or equal to that of the IKE SA this is protecting the negotiation.

The TOE ensures that the strength of the symmetric algorithm (128, 192 or 256 bits) negotiated to protect the IKEv1 Phase 1, IKEv2 IKE_SA connection is greater than or equal to the strength of the symmetric algorithm negotiated to protect the IKEv1 Phase 2, IKEv2 CHILD_SA connection.

Guidance	N/A

N/A

Testing	This test shall be performed for each version of IKE supported. The evaluator shall successfully negotiate an IPsec connection using each of the supported algorithms and hash functions identified in the requirements.

For both IKEv1 and IKEv2, evaluators successfully established an IPsec connection between the TOE and a peer using each of the encryption algorithms (AES-CBC/AES-GCM and associated lengths) and hashing functions specified in FCS_IPSEC_EXT.1.4 and FCS_IPSEC_EXT.1.6.

Testing	This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish an SA for ESP that selects an encryption algorithm with more strength than that being used for the IKE SA (i.e., symmetric algorithm with a key size larger than that being used for the IKE SA). Such attempts should fail.

For both IKEv1 and IKEv2, evaluators attempted to use an IPsec configuration where the encryption strength for Phase 2 (AES-256-CBC) was greater than the strength configured for Phase 1 (AES-128-CBC).

Evaluators confirmed that the TOE would not permit this configuration to be committed and returned the following error:

```
[edit security ipsec vpn VPN]
  'ike'
    The encryption strength of the weakest IKE proposal must be at
least as strong as the strongest IPSec proposal for this VPN. The
strongest IPSec proposal encryption strength is 256 bits, which
exceeds the weakest IKE proposal strength of 128 bits.
error: configuration check-out failed
```

Testing	This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish an IKE SA using an algorithm that is not one of the supported algorithms and hash functions identified in the requirements. Such an attempt should fail.

For both IKEv1 and IKEv2, the evaluators attempted to connect to the TOE using a peer configured to use an invalid algorithm and hash function for Phase 1 (twofish-192 with md5).

For both IKEv1 and IKEv2, the evaluators confirmed that the TOE rejected these algorithms and did not establish a connection.

Testing	This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish an SA for ESP (assumes the proper parameters where used to establish the IKE SA) that selects an encryption algorithm that is not identified in FCS_IPSEC_EXT.1.4. Such an attempt should fail.

For both IKEv1 and IKEv2, the evaluators attempted to connect to the TOE using a peer configured to use an invalid algorithm for Phase 2 (camella192 with SHA-1).

For both IKEv1 and IKEv2, the evaluators confirmed that the TOE rejected the use of the invalid encryption algorithm and did not establish the connection.

2.2.10.13 FCS_IPSEC_EXT.1.13

TSS	The evaluator ensures that the TSS identifies RSA and/or ECDSA as being used to perform peer authentication. The description must be consistent with the algorithms as specified in FCS_COP.1/SigGen Cryptographic Operations (for cryptographic signature).

The TOE supports both RSA and ECDSA for use with X.509v3 certificates that conform to RFC 4945 and pre-shared Keys for IPsec support.

TSS	If pre-shared keys are chosen in the selection, the evaluator shall check to ensure that the TSS describes how pre-shared keys are established and used in authentication of IPsec connections. The description in the TSS shall also indicate how pre-shared key establishment is accomplished for TOEs that can generate a pre-shared key as well as TOEs that simply use a pre-shared key.

The TOE uses pre-shared keys for IPSec. The TOE accepts ASCII pre-shared or bit-based keys of 1 to 255 characters (and their binary equivalent) that may contain upper and lower case letters, numbers, and special characters (that include: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”.

The TOE accepts pre-shared text keys and converts the text string into an authentication value as per RFC 2409 for IKEv1 or RFC 4306 for IKEv2, using the PRF that is configured as the hash algorithm for the IKE exchanges.

Guidance	The evaluator ensures the guidance documentation describes how to set up the TOE to use certificates with RSA and/or ECDSA signatures and public keys.

Chapter 9 of the Evaluated Configuration Guide and the VPN Feature Guide detail how to configure an IPsec VPN with both RSA and ECDSA signatures – these documents cover:

- Generation of public/private key pairs;
- Generation of CSRs;
- Loading of CA and local certificates; and
- Configuration of Phase 1 and Phase 2 settings to ensure that certificates are used for authentication.

Guidance	The evaluator shall check that the guidance documentation describes how pre-shared keys are to be generated and established. The description in the guidance documentation shall also indicate how pre-shared key establishment is accomplished for TOEs that can generate a pre-shared key as well as TOEs that simply use a pre-shared key.

The TOE does not generate pre-shared keys and, as such, all PSKs used must be entered manually by the TOE administrator. This can be accomplished via the following commands, depending on key format:

```
set ike policy IKE_Policy pre-shared-key ascii-text <key>
set ike policy IKE_Policy pre-shared-key hexadecimal <key>
```

Guidance	The evaluator will ensure that the guidance documentation describes how to configure the TOE to connect to a trusted CA, and ensure a valid certificate for that CA is loaded into the TOE and marked "trusted".

Chapters 17 and 18 of the VPN Feature Guide provides instructions on how to create a CA profile, enrol a CA certificate (via SCEP or local upload) and configure the CA for revocation checks (CRL).

By creating a CA profile on the device, the CA is implicitly marked as trusted.

On a per-IPsec policy basis, a CA can be set as trusted and as a preferred CA using the following command:

```
set security ike policy policy-name certificate <CA ID OR use-all>
```

Testing	The evaluator shall configure the TOE to use a private key and associated certificate signed by a trusted CA and shall establish an IPsec connection with the peer.

The evaluators configured the TOE and a peer to use X509 certificates signed by a trusted intermediate CA and confirmed that an IPsec connection could be established using those certificates.

Testing	If pre-shared keys are selected, the evaluator shall generate a pre-shared key off-TOE and use it, as indicated in the guidance documentation, to establish an IPsec connection with the peer.

The evaluators configured the TOE and a peer to use an ASCII pre-shared key for authentication and confirmed that authentication was successfully completed and an IPsec connection established using this pre-shared key.

2.2.10.14 FCS_IPSEC_EXT.1.14

TSS	The evaluator shall ensure that the TSS describes how the TOE compares the peer's presented identifier to the reference identifier. This description shall include which field(s) of the certificate are used as the presented identifier (DN, Common Name, or SAN).

The TOE requires that the configured IKE identity of the local and remote endpoints to match the contents of the certificate associated with a SA endpoint. The TOE permits the identity to be expressed as distinguished name, email address, fully qualified domain name or IP address. If either certificate does not validate, or the contents do not match the configured identity, then the SA will not be established.

TSS	If the ST author assigned an additional identifier type, the TSS description shall also include a description of that type and the method by which that type is compared to the peer's presented certificate.

No other identifiers specified in FCS_IPSEC_EXT.1.14.

Guidance	The evaluator shall ensure that the operational guidance includes the configuration of the reference identifier(s) for the peer.

Per the ECG and VPN Feature Guide, the administrator may set a peer reference identifier using the following command:

```
set security ike gateway <gateway name> remote-identity <value>
```

Where <value> is one of the following:

- distinguished-name container <container-string>
- hostname <hostname>
- inet <ip-address>
- inet6 <ipv6-address>
- user-at-hostname <e-mail-address>

Testing	For each field of the certificate supported for comparison, the evaluator shall configure the peer’s reference identifier on the TOE (per the administrative guidance) to match the field in the peer’s presented certificate and shall verify that the IKE authentication succeeds.

The evaluators configured the TOE to use each of the supported fields (DN, hostname, etc.) for a peer’s reference identifier to match those of the X509 certificate in use.

For each configured field, the evaluators confirmed that the TOE performed a reference identifier check and that authentication successfully completed.

Testing	For each field of the certificate support for comparison, the evaluator shall configure the peer’s reference identifier on the TOE (per the administrative guidance) to not match the field in the peer’s presented certificate and shall verify that the IKE authentication fails.

For each supported reference identifier type (DN, hostname, etc.), the evaluators configured the reference identifier to not match the field in the X509 certificate in use.

Evaluators confirmed that for each misconfigured reference identifier, authentication failed and the TOE rejected the connection attempt.

Testing	(conditional) If the TOE supports DN identifier types, the evaluator shall configure the peer’s reference identifier on the TOE (per the administrative guidance) to match the subject DN in the peer’s presented certificate and shall verify that the IKE authentication succeeds. To demonstrate a bit-wise comparison of the DN, the evaluator shall change a single bit in the DN (preferably, in an Object Identifier (OID) in the DN) and verify that the IKE authentication fails.

The evaluators configured the TOE to use the Distinguished Name container for peer authentication. The evaluators confirmed that authentication could successfully complete with the DN in use.

The evaluators altered the configured DN reference identifier by a single bit and re-attempted authentication. Evaluators confirmed that the comparison between the reference identifier and the DN in the certificate failed and the connection was not established.

2.2.11 FCS_SSHS_EXT.1 SSH Server Protocol

2.2.11.1 FCS_SSHS_EXT.1.1

TSS	N/A

Junos OS SSH server and client are implemented in accordance with RFCs 4251, 4252, 4253, 4254, 5656 and 6668.

Guidance	N/A

N/A

Testing	N/A

N/A

2.2.11.2 FCS_SSHS_EXT.1.2

TSS	The evaluator shall check to ensure that the TSS contains a description of the public key algorithms that are acceptable for use for authentication, that this list conforms to FCS_SSHS_EXT.1.5, and ensure that password-based authentication methods are also allowed

Junos OS supports password-based authentication for SSH connections.

"[…] uses keys generated in accordance with "ssh-rsa", "ecdsa-sha2- nistp256", "ecdsa-sha2-nistp384" or "ecdsa-sha2-nistp521" to perform public-key based device authentication".

Guidance	N/A

N/A

Testing	Using the guidance documentation, the evaluator shall configure the TOE to accept password-based authentication, and demonstrate that a user can be successfully authenticated to the TOE over SSH using a password as an authenticator.

The evaluators configured the TOE to accept only password-based authentication for SSH connections. Evaluators then connected to the TOE from a client device and confirmed that password-based authentication could be successfully completed.

Testing	The evaluator shall use an SSH client, enter an incorrect password to attempt to authenticate to the TOE, and demonstrate that the authentication fails.

Evaluators attempted to connect to the TOE via SSH and, when prompted, entered an incorrect password. Evaluators confirmed that authentication failed and the TOE did not permit access.

2.2.11.3 FCS_SSHS_EXT.1.3

TSS	The evaluator shall check that the TSS describes how "large packets" in terms of RFC 4253 are detected and handled.

Packets greater than 256Kbytes in an SSH transport connection are dropped and the connection is terminated by Junos OS.

Guidance	N/A

N/A

Testing	The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.

The evaluators established an SSH between a client and the TOE. The evaluators then sent a packet of just over 400KB in size.

Evaluators confirmed that the TOE dropped this large packet and dropped the connection.

2.2.11.4 FCS_SSHS_EXT.1.4

TSS	The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the encryption algorithms supported are specified as well.

The TOE offers the following for encryption of SSH sessions: aes128-cbc and aes256-cbc, aes128-ctr, aes256-ctr.

TSS	The evaluator shall check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component.

The listed algorithms match those specified in FCS_SSHS_EXT.1.4.

Guidance	The evaluator shall also check the guidance documentation to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements).

Chapter 4 of the Evaluated Configuration Guide provides configuration statements for SSH – these cover:

- Setting the permissible host-key algorithms to be used;
- Setting the key-exchange methods to be used;
- Setting the message authentication code to be used;
- Setting the encryption algorithm to be used; and
- Setting the maximum permitted failed authentication attempts.

Testing	The evaluator must ensure that only claimed ciphers and cryptographic primitives are used to establish a SSH connection. To verify this, the evaluator shall start session establishment for a SSH connection from a remote client (referred to as 'remote endpoint' below). The evaluator shall capture the traffic exchanged between the TOE and the remote endpoint during protocol negotiation (e.g. using a packet capture tool or information provided by the endpoint, respectively). The evaluator shall verify from the captured traffic that the TOE offers all the ciphers defined in the TSS for the TOE for SSH sessions, but no additional ones compared to the definition in the TSS. The evaluator shall perform one successful negotiation of an SSH session to verify that the TOE behaves as expected. It is sufficient to observe the successful negotiation of the session to satisfy the intent of the test. If the evaluator detects that not all ciphers defined in the TSS for SSH are supported by the TOE and/or the TOE supports one or more additional ciphers not defined in the TSS for SSH, the test shall be regarded as failed.

Per the guidance documentation, the evaluators configured the TOE to only offer those algorithms and cryptographic primitives specified in this requirement. The evaluators then commenced session establishment between a remote client and the TOE while monitoring network traffic between the two.

Evaluators confirmed that the server KEXINIT packet contained only those algorithms specified in this requirement.

2.2.11.5 FCS_SSHS_EXT.1.5

TSS	The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the public key algorithms supported are specified as well.

Junos OS supports password-based authentication for SSH connections.

"[…] uses keys generated in accordance with "ssh-rsa", "ecdsa-sha2- nistp256", "ecdsa-sha2-nistp384" or "ecdsa-sha2-nistp521" to perform public-key based device authentication".

TSS	The evaluator shall check the TSS to ensure that the public key algorithms specified are identical to those listed for this component.

The listed algorithms match those specified in FCS_SSHS_EXT.1.5.

Guidance	The evaluator shall also check the guidance documentation to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements).

Chapter 4 of the Evaluated Configuration Guide provides configuration statements for SSH – these cover:

- Setting the permissible host-key algorithms to be used;
- Setting the key-exchange methods to be used;
- Setting the message authentication code to be used;
- Setting the encryption algorithm to be used; and
- Setting the maximum permitted failed authentication attempts.

Testing	The evaluator shall establish a SSH connection using each of the public key algorithms specified by the requirement to authenticate the TOE to an SSH client. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.

The evaluators connected to the TOE from an SSH client using each of the public key algorithms specified in this requirement (SSH-RSA, ECDSA-256 and ECDSA-384). Via Wireshark analysis, the evaluators were able to confirm successful authentication, negotiation and establishment of an SSH session.

Testing	The evaluator shall choose one public key algorithm supported by the TOE. The evaluator shall generate a new key pair for that algorithm without configuring the TOE to recognize the public key for authentication. The evaluator shall use an SSH client to attempt to connect to the TOE with the new key pair and demonstrate that authentication fails.

The evaluators attempted to authenticate to the TOE via SSH using a private key (SSH-RSA) whose corresponding public key was not configured on the TOE for authentication. The evaluators confirmed that the TOE rejected the provided key and did not permit access to TSF data or services.

Testing	The evaluator shall configure an SSH client to only allow the a public key algorithm that is not included in the ST selection. The evaluator shall attempt to establish an SSH connection from the SSH client to the TOE and observe that the connection is rejected.

The evaluators created a 1024-bit DSA key pair for use in public key authentication. Attempts to load this key onto the TOE for use in SSH public-key authentication were met with an error (as the TOE only permits RSA and ECDSA keys of sizes specified in FCS_SSHS_EXT.1).

2.2.11.6 FCS_SSHS_EXT.1.6

TSS	The evaluator shall check the TSS to ensure that it lists the supported data integrity algorithms, and that that list corresponds to the list in this component.

The TOE permits negotiation of HMAC-SHA1 in each direction for SSH transport.

Both the recommended and optional algorithms hmac-sha2-256 and hmac-sha2-512 (respectively) are implemented for SSH transport.

Guidance	The evaluator shall also check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed data integrity algorithms are used in SSH connections with the TOE (specifically, that the "none" MAC algorithm is not allowed).

Chapter 4 of the Evaluated Configuration Guide provides configuration statements for SSH – these cover:

- Setting the permissible host-key algorithms to be used;
- Setting the key-exchange methods to be used;
- Setting the message authentication code to be used;
- Setting the encryption algorithm to be used; and
- Setting the maximum permitted failed authentication attempts.

Testing	The evaluator shall establish a SSH connection using each of the integrity algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.

The evaluators configured an SSH client to use each of the integrity algorithms specified (hmac-sha1, hmac-sha2-256 and hmac-sha2-512) in turn. Wireshark analysis confirmed that the TOE accepted each of the specified algorithms and established an SSH session.

Testing	The evaluator shall configure an SSH client to only allow a MAC algorithm that is not included in the ST selection. The evaluator shall attempt to connect from the SSH client to the TOE and observe that the attempt fails.

The evaluators configured an SSH client to only use hmac-sha1-96 as its integrity algorithm. Evaluators then attempted to establish an SSH session and confirmed that the TOE rejected the attempt due to an invalid integrity algorithm.

2.2.11.7 FCS_SSHS_EXT.1.7

TSS	The evaluator shall check the TSS to ensure that it lists the supported key exchange algorithms, and that that list corresponds to the list in this component.

Key exchange is performed only using one of the supported key exchange algorithms, which are ordered as follows: ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521 (all specified in RFC 5656), diffie-hellmangroup14- sha1 (specified in RFC 4253).

Guidance	The evaluator shall also check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed key exchange algorithms are used in SSH connections with the TOE.

Chapter 4 of the Evaluated Configuration Guide provides configuration statements for SSH – these cover:

- Setting the permissible host-key algorithms to be used;
- Setting the key-exchange methods to be used;
- Setting the message authentication code to be used;
- Setting the encryption algorithm to be used; and
- Setting the maximum permitted failed authentication attempts.

Testing	The evaluator shall configure an SSH client to only allow the diffie-hellman-group1-sha1 key exchange. The evaluator shall attempt to connect from the SSH client to the TOE and observe that the attempt fails.

The evaluators configured an SSH client to only use DH Group 1 (w/ SHA-1) for key exchange and attempted to connect to the TOE. The evaluators confirmed that the TOE rejected this authentication attempt.

Testing	For each allowed key exchange method, the evaluator shall configure an SSH client to only allow that method for key exchange, attempt to connect from the client to the TOE, and observe that the attempt succeeds.

The evaluators configured an SSH client to use each of the specified key exchange methods (dh-group14-sha1, ecdh-sha2-nistp521, ecdh-sha2-nistp256 and ecdh-sha2-nistp384) in turn.

Evaluators confirmed that, for each specified key exchange method, the TOE permitted the connection and successfully established an SSH session.

2.2.11.8 FCS_SSHS_EXT.1.8

TSS	The evaluator shall check that the TSS specifies the following: • Both thresholds are checked by the TOE. • Rekeying is performed upon reaching the threshold that is hit first.

For ciphers whose blocksize >= 16, the TOE rekeys every (2^32-1) bytes.

The client may explicitly request a rekeying event as a valid SSHv2message at any time and the TOE will honour this request.

Re-keying of SSH session keys can be configured using the sshd_config knob. The data-limit must be between 51200 and 4294967295 (2^32-1) bytes and in the evaluated deployment, the time-limit must be set within 1 and 60 minutes.

Guidance	If one or more thresholds that are checked by the TOE to fulfil the SFR are configurable, then the evaluator shall check that the guidance documentation describes how to configure those thresholds. Either the allowed values are specified in the guidance documentation and must not exceed the limits specified in the SFR (one hour of session time, one gigabyte of transmitted traffic) or the TOE must not accept values beyond the limits specified in the SFR. The evaluator shall check that the guidance documentation describes that the TOE reacts to the first threshold reached.

Rekey lifetimes can be set using the following commands:

```
set system services ssh rekey time-limit 60
set system services ssh rekey data-limit 1073741824
```

The TOE will perform a re-key based on whichever threshold is reached first.

Testing	For testing of the time-based threshold the evaluator shall use an SSH client to connect to the TOE and keep the session open until the threshold is reached. The evaluator shall verify that the SSH session has been active longer than the threshold value and a corresponding audit event has been generated by the TOE.

Evaluators configured the TOE to have an SSH session rekey time of 60 minutes. The evaluators established a session from an SSH client and ensured that the session was kept alive for longer than 60 minutes.

The evaluators confirmed that a) the TOE initiated an SSH rekey upon reaching the 60-minute threshold; and b) an audit log was generated to indicate that the rekey event took place.

Testing	For testing of the traffic-based threshold the evaluator shall use an SSH client to connect to the TOE, and shall transmit data from and to the TOE within the active SSH session until the threshold for transmitted traffic is reached. The transmitted traffic is the total traffic comprising incoming and outgoing traffic. The evaluator shall verify that more data has been transmitted within the SSH session than the threshold allows and a corresponding audit event has been generated by the TOE.

Evaluators configured the TOE to have an SSH session rekey data limit of 1 gigabyte. The evaluators established a session from an SSH client and began to transmit traffic to exceed this threshold (via transfer of a large file).

The evaluators confirmed that a) the TOE initiated an SSH rekey upon reaching the 1 gigabyte threshold; and b) an audit log was generated to indicate that the rekey event took place.

Testing	If one or more thresholds that are checked by the TOE to fulfil the SFR are configurable, the evaluator needs to verify that the threshold(s) can be configured as described in the guidance documentation and the evaluator needs to test that modification of the thresholds is restricted to Security Administrators (as required by FMT_MOF.1/Functions).

The evaluators confirmed that both byte-based and time-based thresholds for SSH rekey are configurable (as specified in the guidance documentation) and are only accessible to authorised administrators.

2.3 Identification and Authentication (FIA)

2.3.1 FIA_AFL.1 Authentication Failure Management

TSS	The evaluator shall examine the TSS to determine that it contains a description, for each supported method for remote administrative actions, of how successive unsuccessful authentication attempts are detected and tracked.

The retry-options can be configured to specify the action to be taken if the administrator fails to enter valid username/password credentials for password authentication. The retry-options are applied following the first failed login attempt for a given username.

The length of delay (5-10 seconds) after each failed attempt is specified by the backoff-factor, and the increase of the delay for each subsequent failed attempt is specified by the backoff-threshold (1-3). The tries-before-disconnect sets the maximum number of times (1-10) the administrator is allowed to enter a password to attempt to log in to the device through SSH before the connection is disconnected.

TSS	The TSS shall also describe the method by which the remote administrator is prevented from successfully logging on to the TOE, and the actions necessary to restore this ability.

The length of delay (5-10 seconds) after each failed attempt is specified by the backoff-factor, and the increase of the delay for each subsequent failed attempt is specified by the backoff-threshold (1-3).

The tries-before-disconnect sets the maximum number of times (1-10) the administrator is allowed to enter a password to attempt to log in to the device through SSH before the connection is disconnected.

The lockout-period sets the amount of time in minutes before the administrator can attempt to log in to the device after being locked out due to the number of failed login attempts (1-43,200 minutes).

It is also possible for another administrator to "unlock" the account of administrator whose account has been locked for a period of time following failed authentication attempts. In this way, the Security Administrator is not permanently blocked from being able to authenticate as the maximum timeout period is 24 hours.

TSS	The evaluator shall examine the TSS to confirm that the TOE ensures that authentication failures by remote administrators cannot lead to a situation where no administrator access is available, either permanently or temporarily (e.g. by providing local logon which is not subject to blocking).

Even when an account is blocked for remote access to the TOE, an administrator is always able to login locally through the serial console and the administrator can attempt authentication via remote access after the maximum timeout period of 24 hours.

Guidance	The evaluator shall examine the guidance documentation to ensure that instructions for configuring the number of successive unsuccessful authentication attempts and time period (if implemented) are provided, and that the process of allowing the remote administrator to once again successfully log on is described for each "action" specified (if that option is chosen). If different actions or mechanisms are implemented depending on the secure protocol employed (e.g., TLS vs. SSH), all must be described.

The subsection 'Limiting the Number of User Login Attempts for SSH Sessions' within Chapter 2 of the Evaluated Configuration Guide provides guidance on how to configure the maximum login attempts for SSH connections. This includes:

- Setting the total permitted tries before disconnect;

- Setting the backoff threshold (number of failed login attempts before the user experiences a delay in being able to enter a password again); and
- Setting the backoff factor (length of time, in seconds, before a user can attempt to log in after a failed attempt).

The administrator may use the `set system login retry-options lockout-period <value>` to set a value (in minutes) that users will be locked out.

An account can either be allowed to unlock 'naturally' (i.e. wait for the configured lockout period to expire) or via the CLI command `clear system login lockout <account name>`.

Guidance	The evaluator shall examine the guidance documentation to confirm that it describes, and identifies the importance of, any actions that are required in order to ensure that administrator access will always be maintained, even if remote administration is made permanently or temporarily unavailable due to blocking of accounts as a result of FIA_AFL.1.

Per the ECG and CLI User Guide, user accounts cannot be locked out from local (console) access and, as such, access is always available to users of the TOE.

Testing	The evaluator shall use the operational guidance to configure the number of successive unsuccessful authentication attempts allowed by the TOE (and, if the time period selection in FIA_AFL.1.2 is included in the ST, then the evaluator shall also use the operational guidance to configure the time period after which access is re-enabled). The evaluator shall test that once the authentication attempts limit is reached, authentication attempts with valid credentials are no longer successful.

The evaluators configured the TOE to permit up to three unsuccessful authentication attempts. The evaluators purposefully failed the authentication process the specified number of times and confirmed that any further authentication attempts were rejected and met with error.

Testing	After reaching the limit for unsuccessful authentication attempts as in Test 1 above, the evaluator shall proceed as follows. If the administrator action selection in FIA_AFL.1.2 is included in the ST then the evaluator shall confirm by testing that following the operational guidance and performing each action specified in the ST to re-enable the remote administrator's access results in successful access (when using valid credentials for that administrator). If the time period selection in FIA_AFL.1.2 is included in the ST then the evaluator shall wait for just less than the time period configured in Test 1 and show that an authorisation attempt using valid credentials does not result in successful access. The evaluator shall then wait until just after the time period configured in Test 1 and show that an authorisation attempt using valid credentials results in successful access.

Administrator action

Evaluators repeated the actions in the previous test to lock out a remote administrator account. Evaluators then accessed the TOE via the local console and, using the commands provided in the guidance documentation, re-enabled remote access for the locked account. Evaluators then confirmed that remote access had been successfully restored for the previously locked account.

Time-based

Evaluators repeated the actions in the previous test to lock out a remote administrator account for ten (10) minutes.

At the nine-minute mark, evaluators confirmed that remote access was still unavailable to the administrator.

At the ten minute and thirty second mark, evaluators confirmed that the TOE had restored remote access to the administrator and were able to access TSF data and services.

2.3.2 FIA_PMG_EXT.1 Password Management

TSS	N/A

Authentication data for fixed password authentication is a case-sensitive, alphanumeric value.

The password has a minimum length of 10 characters and maximum length of 20 characters, and must contain characters from at least two different character sets (upper, lower, numeric, punctuation), and can be up to 20 ASCII characters in length (control characters are not recommended). Any standard ASCII, extended ASCII and Unicode characters can be selected when choosing a password.

Guidance	The evaluator shall examine the guidance documentation to determine that it: a) identifies the characters that may be used in passwords and provides guidance to security administrators on the composition of strong passwords, and b) provides instructions on setting the minimum password length and describes the valid minimum password lengths supported.

Chapter 2 of the Evaluated Configuration Guide allows administrators to set the password policy via the following commands:

```
set system login password minimum-length 10
set system login password change-type character-sets
set system login password minimum-changes 2
set system login password format sha256
```

The ECG also provides guidance on the composition of passwords, including supported characters ("*Include both alphanumeric and punctuation characters, composed of any combination of upper and lowercase letters, numbers, and special characters such as, "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")". There should be at least a change in one case, one or more digits, and one or more punctuation marks.*") and steps to ensure that a chosen password is not easily guessed ("*Permutations on any of the above. For example, a dictionary word with vowels replaced with digits (for example f00t) or with digits added to the end.*").

Testing	The evaluator shall compose passwords that either meet the requirements, or fail to meet the requirements, in some way. For each password, the evaluator shall verify that the TOE supports the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that all characters, and a minimum length listed in the requirement are supported, and justify the subset of those characters chosen for testing.

The evaluators devised a list of passwords to exercise the password management functionality of the TOE. These passwords were:

- Below the minimum length required by the password policy;
- Above the minimum length but containing one or more disallowed characters; or
- Above the minimum length and containing all permitted characters.

Evaluators ensured that each permitted character was used in at least one password.

Evaluators confirmed that all valid passwords were accepted by the TOE and all invalid passwords were rejected.

2.3.3 FIA_UIA_EXT.1 User Identification and Authentication

TSS	The evaluator shall examine the TSS to determine that it describes the logon process for each logon method (local, remote (HTTPS, SSH, etc.)) supported for the product. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a "successful logon".

The internal architecture supporting Authentication includes an active process, associated linked libraries and supporting configuration data. The Authentication process and library are

- login()
- PAM Library module

Following TOE initialization, the login() process is listening for a connection at the local console. This 'login' process can be accessed through either direct connection to the local console or following successful establishment of a remote management connection over SSH, when a login prompt is displayed.

This login process identifies and authenticates the user using PAM operations. The login process does two things; it first establishes that the requesting user is whom they claim to be and second provides them with an interactive Junos Command interactive command line interface (CLI).

The SSH daemon supports public key authentication by looking up a public key in an authorized keys file located in the directory '.ssh' in the user's home directory (i.e. '~/.ssh/') and this authentication method will be attempted before any other if the client has a key available.

login() uses PAM Library calls for the actual verification of this data. The password is hashed and compared to the stored value, and success/failure is indicated to login().

TSS	The evaluator shall examine the TSS to determine that it describes which actions are allowed before user identification and authentication. The description shall cover authentication and identification for local and remote TOE administration.

Prior to authentication, the only Junos OS managed responses provided to the administrator are:

- Negotiation of SSH session
- Display of the access banner
- ICMP echo responses.

Guidance	The evaluator shall examine the guidance documentation to determine that any necessary preparatory steps (e.g., establishing credential material such as preshared keys, tunnels, certificates, etc.) to logging in are described. For each supported the login method, the evaluator shall ensure the guidance documentation provides clear instructions for successfully logging on. If configuration is necessary to ensure the services provided before login are limited, the evaluator shall determine that the guidance documentation provides sufficient instruction on limiting the allowed services.

The Evaluated Configuration Guide provides:

- Guidance on configuring administrator accounts and passwords (Chapter 2); and
- Guidance on configuring the TOE for SSH (Chapter 4).

An administrator successfully authenticates to the TOE by providing a username and password combination matching the stored credentials (for both console and SSH).

There is no configuration required to limit services available prior to login.

Testing	The evaluator shall use the guidance documentation to configure the appropriate credential supported for the login method. For that credential/login method, the evaluator shall show that providing correct I&A information results in the ability to access the system, while providing incorrect information results in denial of access.

The evaluators configured password-based authentication for both local and remote administrator access and public-key based authentication for remote administrator access.

For each local and remote login method, evaluators confirmed that providing incorrect information (incorrect password or invalid private key) caused the TOE to deny access. Providing the correct password or private key successfully completed the I&A process and provided access to the TOE.

Testing	The evaluator shall configure the services allowed (if any) according to the guidance documentation, and then determine the services available to an external remote entity. The evaluator shall determine that the list of services available is limited to those specified in the requirement.

Per the ST, the only services permitted to non-authenticated entities is the viewing of the access banner and ICMP echo.

Evaluators confirmed that, in the evaluated configuration, the TOE will respond to ICMP echo requests and will displayed the configured access banner to any remote entity connecting to the TOE via SSH.

Testing	For local access, the evaluator shall determine what services are available to a local administrator prior to logging in, and make sure this list is consistent with the requirement.

Evaluators confirmed that the TOE presents local entities with the configured access banner. No other services are provided prior to local authentication.

2.3.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

TSS	N/A

The TOE requires users to provide unique identification and authentication data (passwords/key) before any access to the system is granted.

Guidance	N/A

N/A

Testing	N/A

N/A

2.3.5 FIA_UAU.7 Protected Authentication Feedback

TSS	N/A

The username entered by the administrator at the username prompt is reflected to the screen, but no feedback to screen is provided while the entry made by the administrator at the password prompt until the Enter key is pressed.

Guidance	N/A

N/A

Testing	The evaluator shall locally authenticate to the TOE. While making this attempt, the evaluator shall verify that at most obscured feedback is provided while entering the authentication information.

Evaluators confirmed that, while authenticating locally to the TOE, no feedback (visible or otherwise) is provided while entering authentication information.

2.3.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation

TSS	N/A

The TOE uses X.509 certificates as defined in RFC 5280.

To validate certificates, the TOE extracts the subject, issuer, subjects public key, signature, basicConstraints and validity period fields. If any of those fields is not present, the validation fails. The issuer is looked up in the PKI database. If the issuer is not present, or if the issuer certificate does not have the CA:true flag in the basicConstraints section, the validation fails. The TOE verifies the validity of the signature. If the signature is not valid, the validation fails. It then confirms that the current date and time is within the valid time period specified in the certificate. The TOE also extracts the extendedKeyUsage field and verifies the value represents that for the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3).

If the TOE has been configured to perform a revocation check using CRL (as specified in RFC 5280 Section 6.3). If the CRL fails to download, the certificate is considered to have failed validation, unless the option to skip CRL checking on download failure has been enabled.

The TOE validates a certificate path by building a chain of (at least 3) certificates based upon issuer and subject linkage, validating each according the certificate validation procedure described above. If any certificate in the chain fails validation, the validation fails as a whole. A self-signed certificate is not required to be at the root of the certificate chain.

The TOE determines if a certificate is a CA certificate by requiring the CA:true flag to be present in the basicConstraints section.

Guidance	N/A

N/A

Testing	The evaluator shall present the TOE with a valid chain of certificates (terminating in a trusted CA certificate) as needed to validate the certificate to be used in the function, and shall use this chain to demonstrate that the function succeeds. The evaluator shall then delete one of the certificates in the presented chain (i.e. the root CA certificate or other intermediate certificate, but not the end-entity certificate), and show that an attempt to validate an incomplete chain fails.

Evaluators loaded a chain of certificates (Root CA -> Intermediate CA -> TOE and Peer certificates) on to the TOE and configured an IPsec connection to use the certificates for authentication. Evaluators confirmed that a complete verification of the certificate chain was performed and authentication completed successfully.

Evaluators then deleted the Intermediate CA certificate and re-attempted authentication. Evaluators confirmed that, due to the absence of one certificate in the chain, authentication did not complete successfully.

Testing	The evaluator shall demonstrate that validating an expired certificate results in the function failing.

Evaluators attempted to perform IPsec authentication from a peer using an X509 certificate that had expired. Evaluators confirmed that the TOE rejected the expired certificate and authentication failed.

Testing	**The TOE supports CRL only** The evaluator shall test revocation of the peer certificate and revocation of the peer intermediate CA certificate i.e. the intermediate CA certificate should be revoked by the root CA. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails.

The evaluators configured an IPsec connection between the TOE and a peer using X509 certificates for authentication and CRL for revocation checks.

Evaluators confirmed that, when both the Peer and Intermediate CA certificates are marked as valid in the CRL, authentication completes successfully.

Evaluators revoked both the Peer and Intermediate CA certificate in turn, regenerating the CRL as appropriate for each test. Evaluators confirmed that when the revocation check for the Peer or Intermediate CA certificate was performed, the TOE confirmed via the CRL that the relevant certificate was not valid and did not permit authentication to complete successfully.

Testing	The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.)

The evaluators modified the first eight bytes of a certificate contained within an update file. Evaluators attempted to upload this update file and confirmed that validation of the certificate failed.

Testing	The evaluator shall modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)

The evaluators modified the last eight bytes of a certificate contained within an update file. Evaluators attempted to upload this update file and confirmed that validation of the certificate failed.

Testing	The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The hash of the certificate will not validate.

The evaluators modified the public key of a certificate contained within an update file. Evaluators attempted to upload this update file and confirmed that validation of the certificate failed.

Testing	The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate does not contain the basicConstraints extension. The validation of the certificate path fails.

The evaluators attempted to configure a certificate chain on the TOE where the Intermediate CA certificate did not contain the basicConstraints extension. Attempts to validate the certificate path were met with an error.

Testing	The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate has the cA flag in the basicConstraints extension set to FALSE. The validation of the certificate path fails.

The evaluators attempted to configure a certificate chain on the TOE where the Intermediate CA certificate had its cA flag in the basicConstraints section set to FALSE. Attempts to validate the certificate path were met with an error.

Testing	The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate has the cA flag in the basicConstraints extension set to TRUE. The validation of the certificate path succeeds.

The evaluators configured a certificate chain on the TOE where the Intermediate CA certificate had its cA flag in the basicConstraints section set to TRUE. Attempts to validate the certificate path succeeded and the TOE was able to validate the entire certificate chain.

2.3.7 FIA_X509_EXT.2 X.509 Certificate Authentication

TSS	N/A

For public key-based authentication of IPsec connections, Junos OS validates the X.509 certificates by extracting the subject, issuer, signature, basicConstraints and validity period fields. If any of those fields is not present, the validation fails. The issuer is looked up in the PKI database.

If the issuer CA is not present, or if the issuer certificate does not have the CA:true flag in the basicConstraints section, the validation fails.

Junos OS verifies the validity of the signature. If the signature is not valid, the validation fails. It then confirms that the current date and time is within the valid time period specified in the certificate.

Guidance	N/A

N/A

Testing	The evaluator shall demonstrate that using a valid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate, and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the guidance documentation to determine that all supported administrator-configurable options behave in their documented manner.

The evaluators configured the TOE to perform revocation checks of certificates via CRL but did not provide a CRL to be used during these checks. Per the guidance, evaluators configured the TOE to bypass the revocation check when the CRL was unavailable.

Evaluators confirmed that, in the absence of the CRL, the TOE bypassed the revocation check and permitted the certificate validation to continue.

2.3.8 FIA_X509_EXT.3 X.509 Certificate Requests

TSS	N/A

To generate a Certificate Request, the administrator uses the CLI command

- request security pki generate-certificate-request

and supplies the following values:

- Certificate-id – The internal identifier string for this certificate
- Domain-name
- Email address
- IP address
- Subject (DC=<Domain component>,CN=<Common-Name>,OU=<Organizational-Unitname>, O=<Organization-name>,SN=<Serial-Number>,L=<Locality>,ST=<state>,C=<Country>)
- Filename – The local file in which to store the certificate signing request

Guidance	The evaluator shall check to ensure that the guidance documentation contains instructions on requesting certificates from a CA, including generation of a Certificate Request Message. If the ST author selects "Common Name", "Organization", "Organizational Unit", or "Country", the evaluator shall ensure that this guidance includes instructions for establishing these fields before creating the certificate request message.

Per Chapter 18 of the VPN Feature Guide, a certificate subject may contain the following:

"A subject name is associated with the local certificate request in the form of a common name (CN), organizational unit (OU), organization (O), locality (L), state (ST), country (C), and domain component (DC). Additionally, a subject alternative name is associated in the following form:

- *IP address;*
- *E-mail address; and*
- *Fully qualified domain name (FQDN)."*

TOE administrators can generate CSRs via the following command:

```
request security pki generate-certificate-request certificate-id
<certificate-id> subject <subject>
```

An example, taken from the VPN Feature Guide, is as follows:

```
request security pki generate-certificate-request certificate-id ms-
cert subject "CN=john doe,CN=10.1.1.2,OU=sales,O=example,
L=Sunnyvale,ST=CA,C=US" email user@example.net filename ms-cert-req
```

The guidance provides administrators with guidance on submitting certificate requests via SCEP or via manual download and submission to a CA.

Testing	The evaluator shall use the guidance documentation to cause the TOE to generate a certificate request message. The evaluator shall capture the generated message and ensure that it conforms to the format specified. The evaluator shall confirm that the certificate request provides the public key and other required information, including any necessary user-input information.

The evaluators generated a certificate request message on the TOE and exported it to an external CA. The evaluators examined the certificate request and confirmed that it contained all of the information specified in this requirement (public key, CN, O, OU, etc.).

Testing	The evaluator shall demonstrate that validating a certificate response message without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates as trusted CAs needed to validate the certificate response message, and demonstrate that the function succeeds.

Evaluators attempted to validate a certificate response without having the Root and Intermediate CA certificates within the TOE certificate store to allow for validation of the certificate chain. Evaluators confirmed that this validation failed.

Evaluators loaded the two CA certificates into the TOE certificate store to complete the certificate chain. Evaluators confirmed that the validation of the response was successful.

2.3.9 FIA_X509_EXT.4 X.509 Certificate Identity

TSS	N/A

The TOE requires that the configured IKE identity of the local and remote endpoints to match the contents of the certificate associated with a SA endpoint. The TOE permits the identity to be expressed as distinguished name, email address, fully qualified domain name or IP address.

If either certificate does not validate, or the contents do not match the configured identity, then the SA will not be established.

Guidance	N/A

N/A

Testing	N/A

N/A

2.3.10 FIA_PSK_EXT.1 Pre-Shared Keys

TSS	N/A

The TOE uses pre-shared keys for IPSec. The TOE accepts ASCII pre-shared or bit-based keys of 1 to 255 characters (and their binary equivalent) that may contain upper and lower case letters, numbers, and special characters (that include: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")".

The TOE accepts pre-shared text keys and converts the text string into an authentication value as per RFC 2409 for IKEv1 or RFC 4306 for IKEv2, using the PRF that is configured as the hash algorithm for the IKE exchanges.

Guidance	The evaluator shall examine the operational guidance to determine that it provides guidance to administrators on the composition of strong text-based pre-shared keys, and (if the selection indicates keys of various lengths can be entered) that it provides information on the merits of shorter or longer pre-shared keys. The guidance must specify the allowable characters for pre-shared keys, and that list must be a super-set of the list contained in FIA_PSK_EXT.1.2.

Chapter 9 of the Evaluated Configuration Guide provides the following regarding pre-shared key composition:

"*A device running Junos OS uses preshared keys for IPsec (no other protocols). TOE accepts ASCII preshared or bit-based keys up to 255 characters (and their binary equivalents) that contain uppercase and lowercase letters, numbers, and special characters such as !, @, #, $, %, ^, &, *, (, and).*

Note that Junos does not impose minimum complexity requirements for preshared keys. Thus, users are advised to carefully choose long preshared keys of sufficient complexity."

Testing	The evaluator shall compose a pre-shared key of 22 characters that contains a combination of the allowed characters in accordance with the operational guidance, and demonstrates that a successful protocol negotiation can be performed with the key.

Evaluators configured an IPsec connection between the TOE and a peer using a pre-shared key for authentication. Evaluators composed a 22-character pre-shared key using a combination of some of the allowed letters, numbers and symbols defined in the SFR. Evaluators confirmed that mutual authentication and was successful using this key.

Testing	If the TOE supports pre-shared keys of multiple lengths, the evaluator shall repeat Test 1 using the minimum length; the maximum length; and an invalid length. The minimum and maximum length tests should be successful, and the invalid length must be rejected by the TOE.

The evaluators repeated the previous test using pre-shared keys of 1, 22 and 255 characters and confirmed that the TOE permitted the use of the PSKs.

Evaluators attempted to use a PSK that was 256 characters in length. Evaluators confirmed that the TOE did not accept this pre-shared key for use.

Testing	If the TOE does not generate bit-based pre-shared keys, the evaluator shall obtain a bit-based pre-shared key of the appropriate length and enter it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.

The evaluators provided a bit-based pre shared key (based on the previously used 22-character key). The evaluators configured the TOE and peer to use this bit-based PSK for authentication and attempted to establish an IPsec connection.

The evaluators confirmed that the TOE was able to use the bit-based PSK for authentication and successfully established an IPsec tunnel.

2.4 Security Management (FMT)

2.4.1 FMT_MOF.1/ManualUpdate Management of security functions behaviour

TSS	N/A

Updates are downloaded and applied manually (there is no automatic updating of the Junos OS).

Guidance	The evaluator shall examine the guidance documentation to determine that any necessary steps to perform manual update are described. The guidance documentation shall also provide warnings regarding functions that may cease to operate during the update (if applicable).

Per Part 2 of the Installation and Upgrade Guide, system software can be updated via the following commands:

```
request system software add <filename>
request system reboot
```

Testing	The evaluator shall try to perform the update using a legitimate update image without prior authentication as security administrator (either by authentication as a user with no administrator privileges or without user authentication at all – depending on the configuration of the TOE). The attempt to update the TOE shall fail.

The TOE does not permit any access to functionality without prior authentication as a Security Administrator. As such, there is no way to initiate the update process without authentication as a Security Administrator.

Testing	The evaluator shall try to perform the update with prior authentication as security administrator using a legitimate update image. This attempt should be successful. This test case should be covered by the tests for FPT_TUD_EXT.1 already.

The evaluators authenticated to the TOE as a Security Administrator and, per the guidance, requested the TOE perform a firmware upgrade. The evaluators confirmed that the firmware upgrade was applied and the process completed as expected.

2.4.2 FMT_MOF.1/Services Management of security functions behaviour

TSS	N/A

The Security Administrator has the capability to:

- Manage Functions:
  - Transmission of audit data to an external IT entity, including Start/stop and modify the behaviour of the trusted communication channel to external syslog server (netconf over SSH) and the trusted path for remote Administrative sessions (SSH)
  - Handling of audit data, including setting limits of log file size

Guidance	N/A

N/A

Testing	The evaluator shall try to enable and disable at least one of the services as defined in the Application Notes for FAU_GEN.1.1 (whichever is supported by the TOE) without prior authentication as security administrator (either by authenticating as a user with no administrator privileges, if possible, or without prior authentication at all). The attempt to enable/disable this service/these services should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to enable/disable this service/these services can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

No access to TOE services and TSF data is permitted prior to authentication as a Security Administrator. As a non-authenticated user, access extends as far as the login prompt and the user must successfully authenticate before any further access is granted.

Testing	The evaluator shall try to enable and disable at least one of the services as defined in the Application Notes for FAU_GEN.1.1 (whichever is supported by the TOE) with prior authentication as security administrator. The attempt to enable/disable this service/these services should be successful.

Evaluators, as an authenticated Security Administrator, confirmed that the TOE provides the capability to enable/disable SSH and IPsec services.

2.4.3 FMT_MOF.1/Functions Management of security functions behaviour

TSS	N/A

The Security Administrator has the capability to:

- Manage Functions:
  - Transmission of audit data to an external IT entity, including Start/stop and modify the behaviour of the trusted communication channel to external syslog server (netconf over SSH) and the trusted path for remote Administrative sessions (SSH)
  - Handling of audit data, including setting limits of log file size

Guidance	N/A

N/A

Testing	The evaluator shall try to modify all security related parameters for configuration of the transmission protocol for transmission of audit data to an external IT entity without prior authentication as security administrator (by authentication as a user with no administrator privileges or without user authentication at all). Attempts to modify parameters without prior authentication should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to modify the security related parameters can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

No access to TOE services and TSF data is permitted prior to authentication as a Security Administrator. As a non-authenticated user, access extends as far as the login prompt and the user must successfully authenticate before any further access is granted.

Testing	The evaluator shall try to modify all security related parameters for configuration of the transmission protocol for transmission of audit data to an external IT entity with prior authentication as security administrator. The effects of the modifications should be confirmed.

Evaluators authenticated to the TOE as a Security Administrator and confirmed that, when in configuration mode, the functionality was provided to alter security-related parameters (e.g. cipher suites, authentication methods) for transmission of audit logs to an external entity.

Evaluators confirmed that, upon committing the configuration and establishing the secure tunnel for audit log transmission, the revised configuration was used.

Testing	The evaluator shall try to modify all security related parameters for configuration of the handling of audit data without prior authentication as security administrator (by authentication as a user with no administrator privileges or without user authentication at all). Attempts to modify parameters without prior authentication should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator. The term 'handling of audit data' refers to the different options for selection and assignments in SFRs FAU_STG_EXT.1.2, FAU_STG_EXT.1.3 and FAU_STG_EXT.2/LocSpace.

No access to TOE services and TSF data is permitted prior to authentication as a Security Administrator. As a non-authenticated user, access extends as far as the login prompt and the user must successfully authenticate before any further access is granted.

Testing	The evaluator shall try to modify all security related parameters for configuration of the handling of audit data with prior authentication as security administrator. The effects of the modifications should be confirmed. The term 'handling of audit data' refers to the different options for selection and assignments in SFRs FAU_STG_EXT.1.2, FAU_STG_EXT.1.3 and FAU_STG_EXT.2/LocSpace.

Evaluators authenticated to the TOE as a Security Administrator and confirmed that, when in configuration mode, the functionality was provided to alter security-related parameters (such as audit log size, the number of audit log files to be stored on the device, etc.) related to audit log storage.

Evaluators confirmed that, once these parameters had been adjusted, the TOE took action (e.g. creating new log files, deleting older log files, etc.) as expected.

2.4.4 FMT_MTD.1/CoreData Management of TSF Data

TSS	The evaluator shall examine the TSS to determine that, for each administrative function identified in the guidance documentation; those that are accessible through an interface prior to administrator log-in are identified. For each of these functions, the evaluator shall also confirm that the TSS details how the ability to manipulate the TSF data through these interfaces is disallowed for non-administrative users.

The Security Administrator has the capability to:

- Manage TSF data (FMT_MTD,1/CoreData)
    - Create, modify, delete administrator accounts, including configuration of authentication failure parameters
    - Reset administrator passwords
    - Re-enable an Administrator account

No functionality is provided prior to login (with the exception of ICMP response).

Guidance	The evaluator shall review the guidance documentation to determine that each of the TSF-data-manipulating functions implemented in response to the requirements of the cPP is identified, and that configuration information is provided to ensure that only administrators have access to the functions.

The documentation groups functionality into specific sections and/or chapters (IPsec, SSH, firewall rules, etc.), which allows for simple identification of which functions are applicable to the requirements of the cPP/EPs.

The TOE implements a single role, that of the authorised administrator. As such, no configuration is required to restrict access to TOE functions and TSF data.

Testing	N/A

N/A

2.4.5 FMT_MTD.1/CryptoKeys Management of TSF data

TSS	N/A

The Security Administrator has the capability to:

- Manage crypto keys (FMT_MTD.1/CryptoKeys):
    - SSH key generation (ecdsa, ssh-rsa)

Guidance	N/A

N/A

Testing	The evaluator shall try to perform at least one of the related actions (modify, delete, generate/import) without prior authentication as security administrator (either by authentication as a non-administrative user, if supported, or without authentication at all). Attempts to perform related actions without prior authentication should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to manage cryptographic keys can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

No access to TOE services and TSF data is permitted prior to authentication as a Security Administrator. As a non-authenticated user, access extends as far as the login prompt and the user must successfully authenticate before any further access is granted.

Testing	The evaluator shall try to perform at least one of the related actions with prior authentication as security administrator. This attempt should be successful.

The evaluators confirmed that, once authenticated as a Security Administrator, key generation and deletion operations could successfully be executed.

2.4.6 FMT_SMF.1/ND Specification of Management Functions for ND

TSS	The evaluator shall examine the TSS, Guidance Documentation and the TOE as observed during all other testing and shall confirm that the management functions specified in FMT_SMF.1 are provided by the TOE.

The Security Administrator has the capability to:

- Perform management functions:
    - Configure the access banner
    - Configure the session inactivity time before session termination or locking, including termination of session when serial console cable is disconnected
    - Manage cryptographic functionality, including:

        - ssh ciphers
        - hostkey algorithm
        - key exchange algorithm
        - hashed message authentication code
        - thresholds for SSH rekeying
    - Set the system time
    - Ability to configure Firewall rules
    - Ability to configure the VPN-associated cryptographic functionality
    - Ability to configure the IPsec functionality, including configuration of IKE lifetime-seconds (within range 180 to 8640074, with default value of 180 seconds), IPsec lifetime-seconds (within range 180 to 86400, with default value of 28800 seconds75), and Lifetime-kilobytes (within range 64 to 4294967294 kilobytes) and ability to configure the reference identifier for the peer;
    - Ability to enable, disable, determine and modify behavior, and configure all other VPN-associated security functions of the TOE identified in [VPN_EP]

TSS	The evaluator shall confirm that the TSS details which security management functions are available through which interface(s) (local administration interface, remote administration interface).

The TOE provides user access either through the system console or remotely over the Trusted Path using the SSHv2 protocol.

TSS	The evaluator shall verify that the TSS describes how the traffic filter rules for VPN traffic can be configured. Note that this activity can be addressed in parallel with the TSS assurance activities for FPF_RUL_EXT.1.

The Security Administrator has the capability to:

- Perform management functions:
    - Ability to enable, disable, determine and modify behavior, and configure all other VPN-associated security functions of the TOE identified in [VPN_EP]

Guidance	The evaluator shall examine the TSS, Guidance Documentation and the TOE as observed during all other testing and shall confirm that the management functions specified in FMT_SMF.1 are provided by the TOE.

The documentation groups functionality into specific sections and/or chapters (IPsec, SSH, firewall rules, etc.), which allows for simple identification of the functions specified in FMT_SMF.1

Testing	N/A

N/A

2.4.7 FMT_SMF.1/IPS Specification of Management Functions for IPS

TSS	The evaluator shall verify that the TSS describes how the IPS data analysis and reactions can be configured.

The Security Administrator has the capability to:

- Perform management functions:
    - Enable, disable signatures applied to sensor interfaces, and determine the behaviour of IPS functionality
    - Modify these parameters that define the network traffic to be collected and analysed

    - Source IP addresses (host address and network address);
    - Destination IP addresses (host address and network address);
    - Source port (TCP and UDP);
    - Destination port (TCP and UDP);
    - Protocol (IPv4 and IPv6)
    - ICMP type and code
- Update (import) IPS signatures
- Create custom IPS signatures
- Configure anomaly detection
- Enable and disable actions to be taken when signature or anomaly matches are detected
- Modify thresholds that trigger IPS reactions
- Modify the duration of traffic blocking actions
- Modify the known-good and known-bad lists (of IP addresses or address ranges)
- Configure the known-good and known-bad lists to override signature-based IPS policies

Guidance	The evaluator shall verify that the operational guidance describes the instructions for each function defined in the SFR, describes how to configure the IPS data analysis and reactions, including how to set any configurable defaults and how to configure each of the applicable analysis pattern matching methods and reaction modes.

The provided documentation describes each of the management functions defined in the SFR – examination of this information is performed as part of the other assurance activities in the AGD workbook.

All management functions have been covered in the other requirements, with the exception of performing an IDP engine update. This can be performed via the following commands:

```
set security idp security-package url <URL>
set security idp security-package automatic enable
request security idp security-package download full-update
request security idp security-package install
```

Testing	The evaluator shall use the operational guidance to create a signature and enable it on an interface. The evaluator shall then generate traffic that would be successfully triggered by the signature. The evaluator should observe the TOE applying the corresponding reaction in the signature.

The evaluators composed an IDP attack signature (TCP SYN+ACK flags) and assigned it to a security zone. The evaluators then transmitted traffic through the TOE that matched the attack signature and confirmed that the TOE reacted as configured (by dropping the packet).

Testing	The evaluator shall then disable the signature and attempt to regenerate the same traffic and ensure that the TOE allows the traffic to pass with no reaction.

The evaluators disabled the previously configured IDP signature and confirmed that, when transmitting TCP SYN+ACK traffic, the TOE allowed the traffic to flow to its destination with no further processing.

Testing	The evaluator shall use the operational guidance to import signatures and repeat the test conducted in Test 1.

The evaluators imported a TCP SYN+ACK IDP signature in to the TOE and added it to the configuration. Evaluators confirmed that they were able to repeat Test 1 and the TOE behaviour was as expected.

2.4.8 FMT_SMR.2 Restrictions on Security Roles

TSS	N/A

Accounts assigned to the Security Administrator role are used to manage Junos OS in accordance with [NDcPP].

Guidance	The evaluator shall review the guidance documentation to ensure that it contains instructions for administering the TOE both locally and remotely, including any configuration that needs to be performed on the client for remote administration.

The TOE is administered locally via the console port or remotely via SSH.

The Evaluated Configuration Guide provides:

- Guidance on configuring administrator accounts and passwords (Chapter 2); and
- Guidance on configuring the TOE for SSH (Chapter 4).

No additional configuration is required to enable the console port for use.

Testing	In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of this cPP be tested; for instance, if the TOE can be administered through a local hardware interface; SSH; and TLS/HTTPS; then all three methods of administration must be exercised during the evaluation team’s test activities.

Throughout the course of testing, evaluators utilised both local and remote administration interfaces. Evaluators confirmed that both the local console and SSH interfaces conformed to the requirements of the cPPs.

2.5 Protection of the TSF (FPT)

2.5.1 FPT_SKP_EXT.1 Protection of TSF Data

TSS	The evaluator shall examine the TSS to determine that it details how any preshared keys, symmetric keys, and private keys are stored and that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note. If these values are not stored in plaintext, the TSS shall describe how they are protected/obscured.

Junos OS does not provide a CLI interface to permit the viewing of keys. Cryptographic keys are protected through the enforcement of kernel-level file access rights, limiting access to the contents of cryptographic key containers to processes with cryptographic rights or shell users with root permission.

Guidance	N/A

N/A

Testing	N/A

N/A

2.5.2 FPT_APW_EXT.1 Protection of Administrator Passwords

TSS	The evaluator shall examine the TSS to determine that it details all authentication data that are subject to this requirement, and the method used to obscure the plaintext password data when stored.

Locally stored authentication credentials are protected:

- The passwords are stored in obfuscated form using sha-256.
- Authentication data for public key-based authentication methods are stored in a directory owned by the user (and typically with the same name as the user). This directory contains the files '.ssh/authorized_keys' and '.ssh/authorized_keys2' which are used for SSH public key authentication.

TSS	The TSS shall also detail passwords are stored in such a way that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note.

Locally stored authentication credentials are protected:

- The passwords are stored in obfuscated form using sha-256.

Guidance	N/A

N/A

Testing	N/A

N/A

2.5.3 FPT_TST_EXT.1 TSF testing

TSS	The evaluator shall examine the TSS to ensure that it details the self-tests that are run by the TSF; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.

Junos OS runs the following set of self-tests during power on to check the correct operation of the Junos OS firmware:

- Power on test – determines the boot-device responds, and performs a memory size check to confirm the amount of available memory.
- File integrity test –verifies integrity of all mounted signed packages, to assert that system files have not been tampered with. X.509 certificates are used to verify the integrity of the signed packages. As a connection cannot be established to make a real-time determination of certificate validity during the power-on sequence, Junos OS will use the internal trust store (built from the CRL embedded with the latest firmware update) to determine validity. To further test the integrity of the firmware, the fingerprints of the executables and other immutable files are regenerated and validated against the SHA1 fingerprints contains in the manifest file.
- Crypto integrity test – checks integrity of major CSPs, such as SSH hostkeys and iked credentials, such as CAS, CERTS, and various keys.
- Authentication error – verifies that veriexec is enabled and operates as expected using /opt/sbin/kats/cannot-exec.real.
- Kernel, libmd, OpenSSL, QuickSec, SSH IPsec – verifies correct output from known answer tests for appropriate algorithms

Juniper Networks devices run only binaries supplied by Juniper Networks. Within the package, each Junos OS firmware image includes a digitally signed manifest of executables that are registered with the system only if the signature can be validated. Junos firmware will not execute any binary without a registered fingerprint. This feature protects the system against unauthorized firmware and activity that might compromise the integrity of the device.

These self-tests ensure that only authorized executables are allowed to run thus ensuring the correct operation of the TOE.

Guidance	The evaluator shall also ensure that the guidance documentation describes the possible errors that may result from such tests, and actions the administrator should take in response; these possible errors shall correspond to those described in the TSS.

Errors may occur related to any of the self-tests performed by the TOE (e.g. KAT failure, integrity test failure, etc.). If the TOE encounters an error during self-testing, a kernel panic occurs and the device restarts, causing the self-tests to be run again.

If the errors continue to occur, administrators should contact Juniper support.

Testing	It is expected that at least the following tests are performed: a) Verification of the integrity of the firmware and executable software of the TOE b) Verification of the correct operation of the cryptographic functions necessary to fulfil any of the SFRs. Although formal compliance is not mandated, the self-tests performed should aim for a level of confidence comparable to: a) [FIPS 140-2], chap. 4.9.1, Software/firmware integrity test for the verification of the integrity of the firmware and executable software. Note that the testing is not restricted to the cryptographic functions of the TOE. b) [FIPS 140-2], chap. 4.9.1, Cryptographic algorithm test for the verification of the correct operation of cryptographic functions. Alternatively, national requirements of any CCRA member state for the security evaluation of cryptographic functions should be considered as appropriate. The evaluator shall either verify that the self-tests described above are carried out during initial start-up or that the developer has justified any deviation from this.

Evaluators confirmed, via console output, that the module performs integrity checks, function tests and cryptographic module self-tests compliant with FIPS 140-2 security level 2.

2.5.4 FPT_TST_EXT.3 TSF Testing

TSS	N/A

Juniper Networks devices run only binaries supplied by Juniper Networks. Within the package, each Junos OS firmware image includes a digitally signed manifest of executables that are registered with the system only if the signature can be validated. Junos firmware will not execute any binary without a registered fingerprint. This feature protects the system against unauthorized firmware and activity that might compromise the integrity of the device. These self-tests ensure that only authorized executables are allowed to run thus ensuring the correct operation of the TOE.

Guidance	N/A

N/A

Testing	N/A

N/A

2.5.5 FPT_TUD_EXT.1 Trusted update

TSS	The evaluator shall verify that the TSS describe how to query the currently active version.

Security Administrators are able to query the current version of the TOE firmware using the CLI command “show version”

TSS	The evaluator shall verify that the TSS describes all TSF software update mechanisms for updating the system firmware and software (for simplicity the term 'software' will be used in the following although the requirements apply to firmware and software). The evaluator shall verify that the description includes a digital signature verification of the software before installation and that installation fails if the verification fails.

Updates are downloaded and applied manually (there is no automatic updating of the Junos OS).

The installable firmware package containing the Junos OS has a digital signature that is checked when the Security Administrator attempts to install the package. The firmware is digitally signed, and provides a certificate chain that must terminate at one of the internal CA certificates. The signature of the complete package is verified at the beginning of the installation process before the package is expanded. If signature verification fails, an error message is displayed and the package is not installed.

In the NDcPP deployment, "disable on-download-failure" is set to enforce revocation checks using a CRL in local trust store cache. (An updated CRL is loaded during a firmware update, as it is embedded within the firmware binary.) If the certificate considered for validation is not present in the list of revoked certificates in the local cache, then the validation succeeds. If the CRL is not available in Junos OS cache, the certificate is considered to have failed validation.

TSS	If the ST author indicates that a certificate-based mechanism is used for software update digital signature verification, the evaluator shall verify that the TSS contains a description of how the certificates are contained on the device. The evaluator also ensures that the TSS (or guidance documentation) describes how the certificates are installed/updated/selected, if necessary.

The installable firmware package containing the Junos OS has a digital signature that is checked when the Security Administrator attempts to install the package. The firmware is digitally signed, and provides a certificate chain that must terminate at one of the internal CA certificates. The signature of the complete package is verified at the beginning of the installation process before the package is expanded. If signature verification fails, an error message is displayed and the package is not installed.

In the NDcPP deployment, "disable on-download-failure" is set to enforce revocation checks using a CRL in local trust store cache. (An updated CRL is loaded during a firmware update, as it is embedded within the firmware binary.) If the certificate considered for validation is not present in the list of revoked certificates in the local cache, then the validation succeeds. If the CRL is not available in Junos OS cache, the certificate is considered to have failed validation.

Guidance	The evaluator shall verify that the guidance documentation describes how to query the currently active version. If a trusted update can be installed on the TOE with a delayed activation, the guidance documentation needs to describe how to query the loaded but inactive version.

Per the CLI guide, the currently running version of the TOE can be queried via the `show version` command.

Guidance	The evaluator shall verify that the guidance documentation describes how the verification of the authenticity of the update is performed (digital signature verification or verification of published hash). The description shall include the procedures for successful and unsuccessful verification. The description shall correspond to the description in the TSS.

Per Chapter 1 of the Software Installation and Upgrade Guide:

"*Juniper Networks routing platforms run only binaries supplied by Juniper Networks, and currently do not support third-party binaries. Each Junos OS image includes a digitally signed manifest of executables that are registered with the system only if the signature can be validated. Junos OS will not execute any binary without a registered signature.*"

Guidance	**If this was information was not provided in the TSS**: If the ST author indicates that a certificate-based mechanism is used for software update digital signature verification, the evaluator shall verify that the Guidance Documentation contains a description of how the certificates are contained on the device.

The information pertinent to this requirement is provided in the TSS.

Testing	The evaluator performs the version verification activity to determine the current version of the product. If a trusted update can be installed on the TOE with a delayed activation, the evaluator shall also query the most recently installed version (for this test the TOE shall be in a state where these two versions match). The evaluator obtains a legitimate update using procedures described in the guidance documentation and verifies that it is successfully installed on the TOE. For some TOEs loading the update onto the TOE and activation of the update are separate steps ('activation' could be performed e.g. by a distinct activation step or by rebooting the device). In that case the evaluator verifies after loading the update onto the TOE but before activation of the update that the current version of the product did not change but the most recently installed version has changed to the new product version. After the update, the evaluator performs the version verification activity again to verify the version correctly corresponds to that of the update and that current version of the product and most recently installed version match again.

The evaluators executed the 'show version' command and confirmed that the TOE output the current version of the firmware.

The evaluators loaded a legitimate update file onto the device via USB and, using the commands specified in the Installation and Upgrade Guide, confirmed that the TOE successfully installed the new firmware image.

The TOE does not support delayed activation of updates.

Testing	The evaluator first confirms that no updates are pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test. The evaluator obtains or produces illegitimate updates as defined below, and attempts to install them on the TOE. The evaluator verifies that the TOE rejects all of the illegitimate updates. The evaluator performs this test using all of the following forms of illegitimate updates: 1. A modified version (e.g. using a hex editor) of a legitimately signed update 2. An image that has not been signed 3. An image signed with an invalid signature (e.g. by using a different key as expected for creating the signature or by manual modification of a legitimate signature) 4. If the TOE allows a delayed activation of updates the TOE must be able to display both the currently executing version and most recently installed version. The handling of version information of the most recently installed version might differ between different TOEs depending on the point in time when an attempted update is rejected. The evaluator shall verify that the TOE handles the most recently installed version information for that case as described in the guidance documentation. After the TOE has rejected the update the evaluator shall verify, that both, current version and most recently installed version, reflect the same version information as prior to the update attempt.

The evaluators executed the 'show system version' command via the CLI and confirmed that it indicated a version different to that of the update file to be applied.

The evaluators attempted to apply modified updates (modified via hex editor, unsigned firmware file or signed with an invalid development key) and confirmed that, in each instance, the TOE rejected the update file.

The TOE does not support delayed activation of updates.

2.5.6 FPT_TUD_EXT.2 Trusted Update based on certificates

TSS	The evaluator shall verify that the TSS describes how the TOE reacts if X.509 certificates are used for trusted updates and the administrator attempts to perform the trusted update using an expired certificate.

If an administrator attempts to apply an update that contains an expired certificate, the validation of the certificate will fail and the update will be rejected.

TSS	The TSS shall describe the point at which revocation checking is performed. It is expected that revocation checking is performed when a certificate is used when performing trusted updates. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device.

In the NDcPP deployment, "disable on-download-failure" is set to enforce revocation checks using a CRL in local trust store cache72. (An updated CRL is loaded during a firmware update, as it is embedded within the firmware binary.) If the certificate considered for validation is not present in the list of revoked certificates in the local cache, then the validation succeeds. If the CRL is not available in Junos OS cache, the certificate is considered to have failed validation.

Guidance	The evaluator shall verify that the guidance documentation describes how the TOE reacts if X.509 certificates are used for trusted updates and the administrator attempts to perform the trusted update using an expired certificate. The description shall correspond to the description in the TSS.

If an administrator attempts to apply an update that contains an expired certificate, the validation of the certificate will fail and the update will be rejected.

Testing	The evaluator shall verify that the update mechanism includes a certificate validation according to FIA_X509_EXT.1 and a check for the Code Signing purpose in the extendedKeyUsage.

The evaluators verified (via source code review and functional exercise) that the update verification mechanism used by the TOE checks for a Code Signing purpose in the certificate provided with the update image.

Testing	The evaluator shall digitally sign the update with an invalid certificate and verify that update installation fails.

The evaluators attempted to apply a firmware image signed with an invalid certificate and confirmed that the TOE rejected the image.

Testing	The evaluator shall digitally sign the application with a certificate that does not have the Code Signing purpose and verify that application installation fails. The evaluator shall repeat the test using a valid certificate and a certificate that contains the Code Signing purpose and verify that the application installation succeeds.

The evaluators attempted to apply a firmware image signed by a certificate that did not have the Code Signing extendedKeyUsage purpose set. Evaluators confirmed that the TOE rejected the firmware image.

Evaluators attempted to apply a firmware image signed by a certificate that did contain the Code Signing purpose set and confirmed that the image was applied successfully.

Testing	The evaluator shall use a previously valid but expired certificate and verifies that the TOE reacts as described in the TSS and the guidance documentation. Testing for this element is performed in conjunction with the assurance activities for FPT_TUD_EXT.1.

Evaluators attempted to apply a firmware update that contained an expired certificate. Evaluators confirmed that the TOE rejected the image due to the expired certificate.

Testing	The evaluator shall demonstrate that checking the validity of a certificate is performed at the time a certificate is used when performing trusted updates. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device

Evaluators attempted to apply updates that contained expired, modified or otherwise invalid certificates. The evaluators confirmed that, in every instance, the TOE identified either that the certificate was invalid or that the certificate was corrupted/modified.

2.5.7 FPT_STM_EXT.1 Reliable Time Stamps

TSS	The evaluator shall examine the TSS to ensure that it lists each security function that makes use of time, and that it provides a description of how the time is maintained and considered reliable in the context of each of the time related functions.

All events recorded by syslog are timestamped. The clock function of Junos OS provides a source of date and time information for the appliance, used in audit timestamps, which is maintained using the hardware Time Stamp Counter as the clock source.

Guidance	The evaluator examines the guidance documentation to ensure it instructs the administrator how to set the time. If the TOE supports the use of an NTP server, the guidance documentation instructs how a communication path is established between the TOE and the NTP server, and any configuration of the NTP client on the TOE to support this communication.

Per the CLI guide, the date/time can be set via the CLI using `set date YYYYMMDDHHMM.ss` command.

The TOE does not support the use of an NTP server.

Testing	If the TOE supports direct setting of the time by the Security Administrator then the evaluator uses the guidance documentation to set the time. The evaluator shall then use an available interface to observe that the time was set correctly.

The evaluators set the system date and time via the `set date YYYYMMDDHHMM.ss` command and confirmed (via the `show system uptime` command) that the TOE time had been set correctly.

2.5.8 FPT_FLS.1/SelfTest Fail Secure

TSS	The evaluator shall ensure the TSS describes how the TOE ensures a shutdown upon a self-test failure, a failed integrity check of the TSF executable image, or a failed health test of the noise source. If there are instances when a shut-down does not occur, e.g., a failure is deemed non-security relevant, those cases are identified and a rationale supporting the classification and justification why the TOE's ability to enforce its security policies is not affected.

When any self-test fails, the device halts in an error state. No command line input or traffic to any interface is processed. The device must be power cycled to attempt to return to operation.

Guidance	N/A

N/A

Testing	N/A

N/A

2.6 TOE Access (FTA)

2.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

TSS	N/A

The Security Administrator can set the TOE so that a user session is terminated after a period of inactivity.

Guidance	The evaluator shall confirm that the guidance documentation states whether local administrative session locking or termination is supported and instructions for configuring the inactivity time period.

Per the CLI guide, the timeout period for local (serial) connections can be set via the `set cli idle-timeout <minutes>` command.

Testing	The evaluator follows the guidance documentation to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a local interactive session with the TOE. The evaluator then observes that the session is either locked or terminated after the configured time period. If locking was selected from the component, the evaluator then ensures that re-authentication is needed when trying to unlock the session.

The evaluators configured a number of idle timeout periods for the local console connection. The evaluators confirmed that, for each time period defined, the TOE terminated the session after the period of inactivity had expired.

The evaluators confirmed that, once a session had been terminated, re-authentication was required before access to the TOE was restored.

2.6.2 FTA_SSL.3 TSF-initiated Termination

TSS	N/A

The Security Administrator can set the TOE so that a user session is terminated after a period of inactivity.

Guidance	The evaluator shall confirm that the guidance documentation states whether local administrative session locking or termination is supported and instructions for configuring the inactivity time period.

Per the CLI guide, the timeout period for local (serial) connections can be set via the `set cli idle-timeout <minutes>` command.

Testing	The evaluator follows the guidance documentation to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a remote interactive session with the TOE. The evaluator then observes that the session is terminated after the configured time period.

The evaluators configured a number of idle timeout periods for the remote SSH connection. The evaluators confirmed that, for each time period defined, the TOE terminated the session after the period of inactivity had expired.

The evaluators confirmed that, once a session had been terminated, re-authentication was required before access to the TOE was restored.

2.6.3 FTA_SSL.4 User-initiated Termination

TSS	N/A

User sessions (local and remote) can be terminated by users.

Guidance	The evaluator shall confirm that the guidance documentation states how to terminate a local or remote interactive session.

Per the CLI guide, any active CLI session can be closed from the CLI via the `exit` command.

Testing	The evaluator initiates an interactive local session with the TOE. The evaluator then follows the guidance documentation to exit or log off the session and observes that the session has been terminated.

The evaluators established an administrative session via the local console. Once the session had been established, evaluators executed the 'exit' command and confirmed that the session was terminated.

Testing	The evaluator initiates an interactive remote session with the TOE. The evaluator then follows the guidance documentation to exit or log off the session and observes that the session has been terminated.

The evaluators established an administrative session via the remote SSH. Once the session had been established, evaluators executed the 'exit' command and confirmed that the session was terminated.

2.6.4 FTA_TAB.1 Default TOE Access Banners

TSS	The evaluator shall check the TSS to ensure that it details each method of access (local and remote) available to the administrator (e.g., serial port, SSH, HTTPS).

Junos enables Security Administrators to configure an access banner provided with the authentication prompt. The banner can provide warnings against unauthorized access to the secure switch as well as any other information that the Security Administrator wishes to communicate.

The TOE provides user access either through the system console or remotely over the Trusted Path using the SSHv2 protocol.

Guidance	The evaluator shall check the guidance documentation to ensure that it describes how to configure the banner message.

Per the Evaluated Configuration Guide, the login banner can be set via the `set system login message login-message-banner-text` command.

Testing	The evaluator follows the guidance documentation to configure a notice and consent warning message. The evaluator shall then, for each method of access specified in the TSS, establish a session with the TOE. The evaluator shall verify that the notice and consent warning message is displayed in each instance.

The evaluators configured a warning and consent message using the command specified in the guidance documentation.

Evaluators confirmed that the configured message was displayed when connecting to the TOE via both local and remote administrative channels.

2.7 Trusted Path/Channels (FPT)

2.7.1 FTP_ITC.1 Inter-TSF trusted channel

TSS	The evaluator shall examine the TSS to determine that, for all communications with authorized IT entities identified in the requirement, each communications mechanism is identified in terms of the allowed protocols for that IT entity, and the method of assured identification of the non-TSF endpoint.

Junos OS provides an SSH server to support Trusted Channels using SSHv2 protocol which ensures the confidentiality and integrity of communication with the remote audit server. Assured identification of Junos OS is guaranteed by using public key based authentication for SSH.

The TOE implements and supports IPsec.

TSS	The evaluator shall also confirm that all protocols listed in the TSS are specified and included in the requirements in the ST.

Confirmed as part of other TSS work units.

Guidance	The evaluator shall confirm that the guidance documentation contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken.

The TOE utilises IPsec and SSH between itself and remote identities, for both general IPsec traffic and syslog transfer.

The Evaluated Configuration Guide provides instructions for configuring both IPsec VPN connections to remote hosts and the syslog stream via NETCONF to a syslog server. In the event that the connections are unintentionally broken, the TOE shall attempt to reconnect to the remote device.

Testing	The evaluators shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful.

Testing of both SSH and IPsec is performed as part of other evaluation activities.

Testing	For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the guidance documentation to ensure that in fact the communication channel can be initiated from the TOE.

Evaluators performed Wireshark monitoring for both SSH and IPsec and confirmed that both trusted channels could be initiated from the TOE.

Testing	The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.

The evaluators performed Wireshark monitoring for both SSH and IPsec and confirmed that data sent via these channels was not transmitted in plaintext.

Testing	The evaluators shall, for each protocol associated with each authorized IT entity tested during test 1, the connection is physically interrupted. The evaluator shall ensure that when physical connectivity is restored, communications are appropriately protected.

The evaluators established both SSH and IPsec connections between the TOE and a peer before physically interrupting communications.

For SSH, the connection was terminated immediately after physical interruption and required direct intervention before the connection was re-established.

For IPsec, the connection was interrupted immediately after physical interruption. Once the physical connection was restored, the TOE and peer underwent protocol negotiation and a secure IPsec tunnel was restored.

2.7.2 FTP_TRP.1/Admin Trusted Path

TSS	The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected.

Junos OS provides an SSH Server to support Trusted Paths using SSHv2 protocol which ensures the confidentiality and integrity of user sessions. The encrypted communication path between Junos OS SSH Server and a remote administrator is provided by the use of an SSH session.

TSS	The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.

Confirmed as part of other TSS work units.

Guidance	The evaluator shall confirm that the guidance documentation contains instructions for establishing the remote administrative sessions for each supported method.

For SSH connections, a valid username/password combination or SSH key must be provided and authentication must complete successfully before access to TSF functions is granted. The SSH client used must support those ciphers/key exchange methods used by the TOE in its evaluated configuration.

Testing	The evaluators shall ensure that communications using each specified (in the guidance documentation) remote administration method is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful.

Testing of remote administration via SSH within IPsec is performed as part of other evaluation activities.

Testing	The evaluator shall ensure, for each communication channel, the channel data is not sent in plaintext.

The evaluators performed Wireshark monitoring for SSH within IPsec and confirmed that data sent via this channel was not transmitted in plaintext.

Testing	The evaluators shall ensure that, for each protocol tested during test 1, the connection is physically interrupted. The evaluator shall ensure that when physical connectivity is restored, communications are appropriately protected.

The evaluators established SSH and IPsec connections between a remote administrator and the TOE before physically interrupting communications.

For SSH, the connection was terminated immediately after physical interruption and required direct intervention before the connection was re-established.

For IPsec, the connection was interrupted immediately after physical interruption. Once the physical connection was restored, the TOE and peer underwent protocol negotiation and a secure IPsec tunnel was restored.

2.8 User Data Protection (FDP)

2.8.1 FDP_RIP.2 Full Residual Information Protection

TSS	The evaluator shall check to ensure that the TSS describes packet processing to the extent that they can determine that no data will be reused when processing network packets. The evaluator shall ensure that this description at a minimum describes how the previous data are zeroized/overwritten, and at what point in the buffer processing this occurs.

The only resource made available to information flowing through a TOE is the temporary storage of packet information when access is requested and when information is being routed. User data is not persistent when resources are released by one user/process and allocated to another user/process.

Temporary storage (memory) used to build network packets is erased when the resource is called into use by the next user/process. Junos knows, and keeps track of, the length of the packet. This means that when memory allocated from a previous user/process arrives to build the next network packet, Junos is aware of when the end of the packet is reached and pads a short packet with zeros accordingly. Therefore, no residual information from packets in a previous information stream can traverse through the TOE.

Guidance	N/A

N/A

Testing	N/A

N/A

2.9 Packet Filtering (FPF)

2.9.1 FPF_RUL_EXT.1 Rules for Packet Filtering

2.9.1.1 FPF_RUL_EXT.1.1

TSS	The evaluator shall verify that the TSS provide a description of the TOE's initialization/startup process, which clearly indicates where processing of network packets begins to take place, and provides a discussion that supports the assertion that packets cannot flow during this process.

The boot sequence of the TOE appliances also aids in establishing the securing domain and preventing tamping or bypass of security functionality. The following steps list the boot sequence for the TOE:

- BIOS hardware and memory checks
- Loading and initialization of the FreeBSD Kernel OS
- FIPS self-tests and firmware integrity tests are executed
- The init utility is started (mounts file systems, sets up network cards to communicate on the network, and generally starts all the processes that usually are run on a FreeBSD system at startup)
- Daemon programs such as Internet Service Daemon (INETD), Routing Protocol Daemon (RPD), Syslogd are started; Routing and forwarding tables are initialized
- Management Daemon (or MGD) is loaded, allowing access to management interface
- Physical interfaces are active

Once the interfaces are brought up, they will start to receive and send packets based on the current configuration (or not receive or send any packets if they have not been previously configured). Interfaces are brought up only after successful loading of kernel and Information Flow subsystems, and these interfaces cannot send or receive packets unless previously configured by an Administrator.

TSS	The evaluator shall verify that the TSS also includes a narrative that identifies the components (e.g., active entity such as a process or task) involved in processing the network packets and describes the safeguards that would prevent packets flowing through the TOE without applying the ruleset in the event of a component failure. This could include the failure of a component, such as a process being terminated, or a failure within a component, such as memory buffers full and cannot process packets.

Junos is composed of a number of separate executables, or daemons. If a failure occurs in the "flow" daemon (flowd) causing it to halt, no packet processing will occur and no packets will be forwarded. A failure in another daemon will not prevent the flow daemon from enforcing the policy rule set.

The Information Flow subsystem is responsible for processing the arriving packets from the network to the TOE's network interface. Based on Administrator-configured policy, interface and zone information, the packet flows through the various modules of the Information Flow subsystem. Rules within policies are processed in an Administrator-defined order when network traffic flows through the TOE network interfaces.

The Information Flow subsystem consists of the following modules:

- IP Classification Module
- Attack Detection Module
- Session Lookup Module
- Security Policy Module
- Session Setup Module

- Inetd Module
- Rdp Module

Guidance	The operational guidance associated with this requirement is assessed in the subsequent test assurance activities.

N/A

Testing	The evaluator shall attempt to get network traffic to flow through the TOE while the TOE is being initialized. A steady flow of network packets that would otherwise be denied by the ruleset should be directed at the TOE's interfaces, with packet sniffers listening to see if any network traffic is allowed through.

The evaluators restarted the TOE and attempted to ping from one subnet to another while the TOE was initialising. The evaluators confirmed that no traffic was allowed to flow through the TOE.

2.9.1.2 FPF_RUL_EXT.1.2

TSS	The evaluator shall verify that the TSS indicates that the following protocols are supported: • RFC 791 (IPv4) • RFC 2460 (IPv6) • RFC 793 (TCP) • RFC 768 (UDP)

The TOE performs stateful network traffic filtering on network packets using the following network traffic protocols and network fields conforming to the described RFCs:

- Internet Control Message Protocol version 4 (ICMPv4)
  - RFC 792 (ICMPv4)
- Internet Control Message Protocol version 6 (ICMPv6)
  - RFC 4443 (ICMPv6)
- Internet Protocol (IPv4)
  - RFC 791 (IPv4)
- Internet Protocol version 6 (IPv6)
  - RFC 2460 (IPv6)
- Transmission Control Protocol (TCP)
  - RFC 793 (TCP)
- User Datagram Protocol (UDP)
  - RFC 768 (UDP)

TSS	The evaluator shall verify that the TSS describes how conformance with the identified RFCs has been determined by the TOE developer (e.g., third party interoperability testing, protocol compliance testing).

Conformance to these RFCs is demonstrated by protocol compliance testing by the product QA team.

Guidance	The evaluator shall verify that the operational guidance indicates that the following protocols are supported: • RFC 791 (IPv4) • RFC 2460 (IPv6) • RFC 793 (TCP) • RFC 768 (UDP) The guidance will describe the other protocols contained within the ST (e.g., IPsec, IKE, potentially HTTPS, SSH, and TLS) that are processed by the TOE. The evaluator ensures it is made clear what protocols were not considered as part of the TOE evaluation.

Per Chapter 5, Table 6 of the Evaluated Configuration Guide, the TOE supports IPv4/6, TCP and UDP. The TOE also processes IPsec, IKE, SSH, OSPF, BGP and RIP (the latter three are not included in the scope of evaluation).

Testing	N/A

N/A

2.9.1.3 FPF_RUL_EXT.1.3

TSS	N/A

The TOE performs stateful network traffic filtering on network packets using the following network traffic protocols and network fields conforming to the described RFCs:

- Internet Control Message Protocol version 4 (ICMPv4)
  - Type, Code
- Internet Control Message Protocol version 6 (ICMPv6)
  - Type, Code
- Internet Protocol (IPv4)
  - Source address, Destination Address, Transport Layer, Protocol
- Internet Protocol version 6 (IPv6)
  - Source address, Destination Address, Transport Layer, Protocol
- Transmission Control Protocol (TCP)
  - Source port, Destination port
- User Datagram Protocol (UDP)
  - Source port, Destination port

Guidance	N/A

N/A

Testing	N/A

N/A

2.9.1.4 FPF_RUL_EXT.1.4

TSS	N/A

The TOE shall allow permit, deny, and log operations to be associated with rules and these rules can be assigned to distinct network interfaces.

Guidance	N/A

N/A

Testing	N/A

N/A

2.9.1.5 FPF_RUL_EXT.1.5

TSS	The evaluator shall verify that the TSS describes a Packet Filtering policy and the following attributes are supported: • IPv4 ○ Source address ○ Destination Address ○ Protocol • IPv6 ○ Source address ○ Destination Address ○ Next Header (Protocol) • TCP ○ Source Port ○ Destination Port • UDP ○ Source Port ○ Destination Port

Firewall Policies match Source Zone, Destination Zone, Source IP, Destination IP, Source Port, Destination Port, and Protocol. Interface and VLAN matching can be achieved through the use of zones. Rules are organized into a firewall policy rulebase.

The TOE performs stateful network traffic filtering on network packets using the following network traffic protocols and network fields conforming to the described RFCs:

- Internet Control Message Protocol version 4 (ICMPv4)
  - Type, Code
- Internet Control Message Protocol version 6 (ICMPv6)
  - Type, Code
- Internet Protocol (IPv4)
  - Source address, Destination Address, Transport Layer Protocol
- Internet Protocol version 6 (IPv6)
  - Source address, Destination Address, Transport Layer Protocol
- Transmission Control Protocol (TCP)
  - Source port, Destination port
- User Datagram Protocol (UDP)
  - Source port, Destination port

TSS	The evaluator shall verify that each rule can identify the following actions: permit, deny, and log.

The TOE shall allow permit, deny, and log operations to be associated with rules and these rules can be assigned to distinct network interfaces.

TSS	The evaluator shall verify that the TSS identifies all interface types subject to the Packet Filtering policy and explains how rules are associated with distinct network interfaces. Where interfaces can be grouped into a common interface type (e.g., where the same internal logical path is used, perhaps where a common device driver is used) they can be treated collectively as a distinct network interface.

The TOE is capable of inspecting all traffic passing through the TOE’s Ethernet interfaces (inline mode). Ethernet interfaces can be assigned to Zones on which firewall and IDP policies are predicated.

Guidance	The evaluators shall verify that the operational guidance identifies the following attributes as being configurable within Packet filtering rules for the associated protocols: • IPv4 ○ Source address ○ Destination Address ○ Protocol • IPv6 ○ Source address ○ Destination Address ○ Next Header (Protocol) • TCP ○ Source Port ○ Destination Port • UDP ○ Source Port ○ Destination Port

Per Chapter 11 of the Evaluated Configuration Guide, the TOE supports the following protocols and attributes:

- IPv4 - RFC 791, Internet Protocol
    - Source address
    - Destination address
    - Transport Layer Protocol
- IPv6 - RFC 2460, Internet Protocol
    - Source address
    - Destination address
    - Transport Layer Protocol
- TCP - RFC 793, Transmission Control Protocol
    - Source port
    - Destination port
- UDP - RFC 768, User Datagram Protocol
    - Source port
    - Destination port

Guidance	The evaluator shall verify that the operational guidance indicates that each rule can identify the following actions: permit, deny, and log.

Per the Evaluated Configuration Guide, the following actions can be associated with each security flow policy:

- Bypass— The Permit option directs the traffic traversing the device through the stateful firewall inspection, but not through the IPsec VPN tunnel.
- Discard— The Deny option inspects and drops all packets that do not match any Permit policies.
- Protect— The traffic is routed through an IPsec tunnel based on the combination of route lookup and Permit policy inspection.
- Log— This option logs traffic and session information for all the modes mentioned above.

Guidance	The evaluator shall verify that the operational guidance explains how rules are associated with distinct network interfaces.

Per the Evaluated Configuration Guide:

- Interfaces are assigned to one or more security zones.
- Security rules/policies are assigned to security screens. Each screen may be assigned to one or more security zones.
- A chain is created from Rule -> Screen -> Zone -> Interface.

Testing	The evaluator shall use the instructions in the operational guidance to test that packet filter rules can be created that permit, deny, and log packets for each of the following attributes: • IPv4 o Source address o Destination Address o Protocol • IPv6 o Source address o Destination Address o Next Header (Protocol) • TCP o Source Port o Destination Port • UDP o Source Port o Destination Port

The evaluators constructed and tested numerous packet filtering rules that exercised each of the protocols, attributes and reactions listed in this requirement. The evaluators confirmed that, for each configured rule, the TOE behaved as expected and audit logs were generated appropriately.

Testing	Repeat the test assurance activity above to ensure that Packet filtering rules can be defined for each distinct network interface type supported by the TOE.

The evaluators confirmed that packet filtering rules could be created and assigned to the interface types supported by the TOE.

2.9.1.6 FPF_RUL_EXT.1.6

TSS	The evaluator shall verify that the TSS describes the algorithm applied to incoming packets, including the processing of default rules, determination of whether a packet is part of an established session, and application of administrator defined and ordered ruleset.

The Information Flow subsystem is responsible for processing the arriving packets from the network to the TOE's network interface. Based on Administrator-configured policy, interface and zone information, the packet flows through the various modules of the Information Flow subsystem. Rules within policies are processed in an Administrator-defined order when network traffic flows through the TOE network interfaces.

By default, the TOE behavior is to deny packets when there is no rule match unless another required condition allows the network traffic if a security risk is found in the packet. E.g., denial-of-service attacks, the packet is dropped and an event is logged. The packet does not continue to the next module for processing. If the packet is not dropped by a given module, the interrupt handling routine calls the function for the next relevant module.

The IP Classification module retrieves information from packets received on the network interface device, classifies packets into several categories, saves classification information in packet processing context, and provides other modules with that information for assisting further processing.

The Attack Detection module provides inline attack detection such as IP Spoofing for the security appliance. This module monitors arriving traffic, performs predefined attack detection services (prevents attacks), and issues actions when an attack is found.

The Session Lookup module performs lookups in the session table that is used for all interfaces based on the information in incoming packets. Specifically, the lookup is based on the exact match of source IP address and port, destination IP address and port, protocol attributes (e.g., SYN, ACK, RST, and FIN), and egress/ingress zone. The input is passed to the module as a set of parameters from the Attack Detection module via a function call. Sessions are removed when terminated.

The Session Setup module is only available for packets that do not match current established sessions. It is activated after the Session Lookup module. If packet has a matched session, it will skip the session setup module and proceed to the Security Policy module, and other modules. Eventually if the packet is not destined for the TOE, the Network interface will pass the traffic out of the appliance.

The Security Policy module examines traffic passing through the TOE (via Session Setup module) and determines if the traffic can pass based on administrator-configured access policies. The Security Policy module is the core of the firewall and IPS functionalities in the TOE: The policy enforcement engine fulfils the security requirements for the user. The Security Policy module will deny packets when there is no policy match unless another policy allows the traffic.

The Session Setup module performs the auditing of denied packets. If there is a policy to specifically deny traffic, traffic matching this deny policy is dropped and logged in traffic log. If there is no policy to deny traffic, traffic that does not match any policy is dropped and not logged. In either case, Session Setup module does not create any sessions for denied traffic. Sessions are created for allowed traffic.

The INETD module provides internet services for the TOE. The module listens on designated ports used by internet services such as FTP. When a TCP or UDP packet arrives with a particular destination port number, INETD launches the appropriate server program (e.g., SSHD) to handle the connection.

The RPD (Routing Protocol Daemon) module provides the implementations and algorithms for the routing protocols and route calculations. The primary goal of the RPD is to create and maintain the Routing Information Base (RIB), which is a database of routing entries. Each routing entry consists of a destination address and some form of next hop information. RPD module maintains the internal routing table and properly distributes routes from the routing table to Kernel subsystem used for traffic forwarding at the Network interface.

Guidance	The evaluator shall verify that the operational guidance describes how the order of Packet filtering rules is determined and provides the necessary instructions so that an administrator can configure the order of rule processing.

By default, traffic filtering rules are enforced in a top-to-bottom order of precedence (i.e. terms are examined sequentially). However, if the administrator wishes to change the order of the rules in place, the following command (from the CLI User Guide) allows them to do so:

```
insert <statement-path> identifier1 (before | after) identifier2
```

Testing	The evaluator shall devise two equal Packet filtering rules with alternate operations – permit and deny. The rules should then be deployed in two distinct orders and in each case the evaluator shall ensure that the first rule is enforced in both cases by generating applicable packets and using packet capture and logs for confirmation.

The evaluators constructed two packet filtering rules, the only difference between the two being the action to be taken (permit/deny).

The evaluators confirmed that rules are processed in the order in which they are in defined. Wireshark monitoring and audit log examination corroborated this finding.

Testing	The evaluator shall repeat the procedure above, except that the two rules should be devised where one is a subset of the other (e.g., a specific address vs. a network segment). Again, the evaluator should test both orders to ensure that the first is enforced regardless of the specificity of the rule.

The evaluators constructed two packet filtering rules, one being a subset of the other (10.0.2.0/24 and 10.0.2.2).

The evaluators confirmed that rules are processed in the order in which they are in defined. Wireshark monitoring and audit log examination corroborated this finding.

2.9.1.7 FPF_RUL_EXT.1.7

TSS	The evaluator shall verify that the TSS describes the process for applying Packet filtering rules and also that the behavior (either by default, or as configured by the administrator) is to deny packets when there is no rule match unless another required conditions allows the network traffic (i.e., FPF_RUL_EXT.1.6 or FPF_RUL_EXT.1.7).

The Information Flow subsystem is responsible for processing the arriving packets from the network to the TOE's network interface. Based on Administrator-configured policy, interface and zone information, the packet flows through the various modules of the Information Flow subsystem. Rules within policies are processed in an Administrator-defined order when network traffic flows through the TOE network interfaces.

By default, the TOE behavior is to deny packets when there is no rule match unless another required condition allows the network traffic. If a security risk is found in the packet (e.g., denial-of-service attacks), the packet is dropped and an event is logged. The packet does not continue to the next module for processing. If the packet is not dropped by a given module, the interrupt handling routine calls the function for the next relevant module.

Guidance	The evaluator shall verify that the operational guidance describes the behavior if no rules or special conditions apply to the network traffic. If the behavior is configurable, the evaluator shall verify that the operational guidance provides the appropriate instructions to configure the behavior to deny packets with no matching rules.

Per the Evaluated Configuration Guide, the default reject-all rule can be implemented via the following command:

```
set security policies default-policy deny-all
```

This rule will be applied to all traffic received that does not meet any other configured rule.

Testing	The evaluator shall configure the TOE to permit and log each defined IPv4 Transport Layer Protocol (see table 5-2) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv4 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that they are permitted (i.e., by capturing the packets after passing through the TOE) and logged.

The evaluators configured the TOE to permit and log each of the transport layer protocols identified in the VPNEP with a combination of addresses. Evaluators confirmed, via audit log examination and Wireshark analysis, that the TOE identified the protocol in use and processed the packet appropriately.

Testing	The evaluator shall configure the TOE to permit all traffic except to deny and log each defined IPv4 Transport Layer Protocol (see table 5-2) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv4 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged.

The evaluators configured the TOE to permit and log each of the transport layer protocols identified in the VPNEP with a combination of addresses. Evaluators confirmed, via audit log examination and Wireshark analysis, that the TOE identified the protocol in use and processed the packet appropriately.

Testing	The evaluator shall configure the TOE to permit and log each defined IPv4 Transport Layer Protocol (see table 5-2) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. Additionally, the evaluator shall configure the TOE to deny and log each defined IPv4 Transport Layer Protocol (See table 5-2) in conjunction with different (than those permitted above) combinations of a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv4 Transport Layer Protocol and outside the scope of all source and destination addresses configured above in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE).

The evaluators configured the TOE to permit and log each of the transport layer protocols identified in the VPNEP with a combination of addresses. Evaluators confirmed, via audit log examination and Wireshark analysis, that the TOE identified the protocol in use and processed the packet appropriately.

Testing	The evaluator shall configure the TOE to permit and log each defined IPv6 Transport Layer Protocol (see table 5-2) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv6 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that they are permitted (i.e., by capturing the packets after passing through the TOE) and logged.

The evaluators configured the TOE to permit and log each of the IPv6 transport layer protocols identified in the VPNEP with a combination of addresses. Evaluators confirmed, via audit log examination and Wireshark analysis, that the TOE identified the protocol in use and processed the packet appropriately.

Testing	The evaluator shall configure the TOE to permit all traffic except to deny and log each defined IPv6 Transport Layer Protocol (see table 5-2) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv6 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged.

The evaluators configured the TOE to permit and log each of the IPv6 transport layer protocols identified in the VPNEP with a combination of addresses. Evaluators confirmed, via audit log examination and Wireshark analysis, that the TOE identified the protocol in use and processed the packet appropriately.

Testing	The evaluator shall configure the TOE to permit and log each defined IPv6 Transport Layer Protocol (see table 5-2) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. Additionally, the evaluator shall configure the TOE to deny and log each defined IPv6 Transport Layer Protocol (see table 5-2) in conjunction with different (than those permitted above) combinations of a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv6 Transport Layer Protocol and outside the scope of all source and destination addresses configured above in order to ensure that they are dropped (i.e., by capturing no applicable packets passing through the TOE) and logged.

The evaluators configured the TOE to permit and log each of the IPv6 transport layer protocols identified in the VPNEP with a combination of addresses. Evaluators confirmed, via audit log examination and Wireshark analysis, that the TOE identified the protocol in use and processed the packet appropriately.

Testing	The evaluator shall configure the TOE to permit and log protocol 6 (TCP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination TCP ports in order to ensure that they are permitted (i.e., by capturing the packets after passing through the TOE) and logged.

The evaluators configured the TOE to permit and log TCP with a selected source and destination address/port. Evaluators confirmed, via audit log examination and Wireshark analysis, that the TOE identified the ports in use and permitted the traffic.

Testing	The evaluator shall configure the TOE to deny and log protocol 6 (TCP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination TCP ports in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged.

The evaluators configured the TOE to permit and log TCP with a selected source and destination address/port. Evaluators confirmed, via audit log examination and Wireshark analysis, that the TOE identified the ports in use and denied the traffic.

Testing	The evaluator shall configure the TOE to permit and log protocol 17 (UDP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination UDP ports in order to ensure that they are permitted (i.e., by capturing the packets after passing through the TOE) and logged. Here the evaluator ensures that the UDP port 500 (IKE) is included in the set of tests.

The evaluators configured the TOE to permit and log UDP with a selected source (25) and destination address/port (500). Evaluators confirmed, via audit log examination and Wireshark analysis, that the TOE identified the protocol in use and permitted the traffic.

Testing	The evaluator shall configure the TOE to deny and log protocol 17 (UDP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination UDP ports in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged. Again, the evaluator ensures that UDP port 500 is included in the set of tests.

The evaluators configured the TOE to permit and log UDP with a selected source (25) and destination address/port (500). Evaluators confirmed, via audit log examination and Wireshark analysis, that the TOE identified the protocol in use and denied the traffic.

2.10 Firewall (FFW)

2.10.1 FFW_RUL_EXT.1 Stateful Traffic Filtering

2.10.1.1 FFW_RUL_EXT.1.1

TSS	The evaluator shall verify that the TSS provides a description of the TOE's initialization/startup process, which clearly indicates where processing of network packets begins to take place, and provides a discussion that supports the assertion that packets cannot flow during this process.

The boot sequence of the TOE appliances also aids in establishing the securing domain and preventing tamping or bypass of security functionality. The following steps list the boot sequence for the TOE:

- BIOS hardware and memory checks
- Loading and initialization of the FreeBSD Kernel OS
- FIPS self-tests and firmware integrity tests are executed
- The init utility is started (mounts file systems, sets up network cards to communicate on the network, and generally starts all the processes that usually are run on a FreeBSD system at startup)
- Daemon programs such as Internet Service Daemon (INETD), Routing Protocol Daemon (RPD), Syslogd are started; Routing and forwarding tables are initialized
- Management Daemon (or MGD) is loaded, allowing access to management interface
- Physical interfaces are active

Once the interfaces are brought up, they will start to receive and send packets based on the current configuration (or not receive or send any packets if they have not been previously configured). Interfaces are brought up only after successful loading of kernel and Information Flow subsystems, and these interfaces cannot send or receive packets unless previously configured by an Administrator.

TSS	The evaluator shall verify that the TSS also include a narrative that identifies the components (e.g., active entity such as a process or task) involved in processing the network packets and describe the safeguards that would prevent packets flowing through the TOE without applying the ruleset in the event of a component failure. This could include the failure of a component, such as a process being terminated, or a failure within a component, such as memory buffers full and cannot process packets.

Junos is composed of a number of separate executables, or daemons. If a failure occurs in the "flow" daemon (flowd) causing it to halt, no packet processing will occur and no packets will be forwarded. A failure in another daemon will not prevent the flow daemon from enforcing the policy rule set.

The Information Flow subsystem is responsible for processing the arriving packets from the network to the TOE's network interface. Based on Administrator-configured policy, interface and zone information, the packet flows through the various modules of the Information Flow subsystem. Rules within policies are processed in an Administrator-defined order when network traffic flows through the TOE network interfaces.

The Information Flow subsystem consists of the following modules:

- IP Classification Module
- Attack Detection Module
- Session Lookup Module
- Security Policy Module
- Session Setup Module

- Inetd Module
- Rdp Module

Guidance	The guidance documentation associated with this requirement is assessed in the subsequent test assurance activities.

N/A

Testing	The evaluator shall attempt to get network traffic to flow through the TOE while the TOE is being initialized. A steady flow of network packets that would otherwise be denied by the ruleset should be sourced and be directed at a host. The evaluator shall verify using a packet sniffer that none of the generated network traffic is permitted through the firewall during initialization.

The evaluators restarted the TOE and attempted to ping from one subnet to another while the TOE was initialising. The evaluators confirmed that no traffic was allowed to flow through the TOE.

Testing	The evaluator shall attempt to get network traffic to flow through the TOE while the TOE is being initialized. A steady flow of network packets that would be permitted by the ruleset should be sourced and be directed at a host. The evaluator shall verify using a packet sniffer that none of the generated network traffic is permitted through the firewall during initialization and is only permitted once initialization is complete.

The evaluators restarted the TOE and attempted to ping from one subnet to another while the TOE was initialising. The evaluators confirmed that no traffic was allowed to flow through the TOE until the initialisation had completed and the network interfaces brought online.

2.10.1.2 FFW_RUL_EXT.1.2 / FFW_RUL_EXT.1.3 / FFW_RUL_EXT.1.4

TSS	The evaluator shall verify that the TSS describes a stateful packet filtering policy and the following attributes are identified as being configurable within stateful traffic filtering rules for the associated protocols: • ICMPv4 ○ Type ○ Code • ICMPv6 ○ Type ○ Code • IPv4 ○ Source address ○ Destination Address ○ Transport Layer Protocol • IPv6 ○ Source address ○ Destination Address ○ Transport Layer Protocol and where defined by the ST author, Extension Header Type, Extension Header Fields • TCP ○ Source Port ○ Destination Port • UDP ○ Source Port ○ Destination Port

Firewall Policies match Source Zone, Destination Zone, Source IP, Destination IP, Source Port, Destination Port, and Protocol. Interface and VLAN matching can be achieved through the use of zones. Rules are organized into a firewall policy rulebase.

The TOE performs stateful network traffic filtering on network packets using the following network traffic protocols and network fields conforming to the described RFCs:

- Internet Control Message Protocol version 4 (ICMPv4)
  - Type, Code
- Internet Control Message Protocol version 6 (ICMPv6)
  - Type, Code
- Internet Protocol (IPv4)
  - Source address, Destination Address, Transport Layer Protocol
- Internet Protocol version 6 (IPv6)
  - Source address, Destination Address, Transport Layer Protocol
- Transmission Control Protocol (TCP)
  - Source port, Destination port
- User Datagram Protocol (UDP)
  - Source port, Destination port

TSS	The evaluator shall verify that each rule can identify the following actions: permit or drop with the option to log the operation.

The TOE shall allow permit, deny, and log operations to be associated with rules and these rules can be assigned to distinct network interfaces.

TSS	The evaluator shall verify that the TSS identifies all interface types subject to the stateful packet filtering policy and explains how rules are associated with distinct network interfaces.

The TOE is capable of inspecting all traffic passing through the TOE's Ethernet interfaces (inline mode). Ethernet interfaces can be assigned to Zones on which firewall and IDP policies are predicated.

Guidance	The evaluators shall verify that the guidance documentation identifies the following attributes as being configurable within stateful traffic filtering rules for the associated protocols: • ICMPv4 ○ Type, Code • ICMPv6 ○ Type, Code • IPv4 ○ Source address, Destination Address, Transport Layer Protocol • IPv6 ○ Source address, Destination Address, Transport Layer Protocol and where defined by the ST author, Extension Header Type, Extension Header Fields • TCP ○ Source Port, Destination Port • UDP ○ Source Port, Destination Port

Table 14 within the Evaluated Configuration Guide lists all of the protocols and applicable attributes listed above.

Guidance	The evaluator shall verify that the guidance documentation indicates that each rule can identify the following actions: permit, drop, and log.

Per the Evaluated Configuration Guide, the following actions can be associated with each security flow policy:

- Bypass— The Permit option directs the traffic traversing the device through the stateful firewall inspection, but not through the IPsec VPN tunnel.
- Discard— The Deny option inspects and drops all packets that do not match any Permit policies.
- Protect— The traffic is routed through an IPsec tunnel based on the combination of route lookup and Permit policy inspection.
- Log— This option logs traffic and session information for all the modes mentioned above.

Guidance	The evaluator shall verify that the guidance documentation explains how rules are associated with distinct network interfaces.

Per the Evaluated Configuration Guide:

- Interfaces are assigned to one or more security zones.
- Security rules/policies are assigned to security screens. Each screen may be assigned to one or more security zones.
- A chain is created from Rule -> Screen -> Zone -> Interface.

Testing	The evaluator shall use the instructions in the operational guidance to test that packet filter rules can be created that permit, deny, and log packets for each of the following attributes: • IPv4 ○ Source address ○ Destination Address ○ Protocol • IPv6 ○ Source address ○ Destination Address ○ Transport Layer Protocol and where defined by the ST author, xtension Header Type, Extension Header Fields • TCP ○ Source Port ○ Destination Port • UDP ○ Source Port ○ Destination Port

The evaluators constructed and tested numerous packet filtering rules that exercised each of the protocols, attributes and reactions listed in this requirement. The evaluators confirmed that, for each configured rule, the TOE behaved as expected and audit logs were generated appropriately.

Testing	Repeat the test assurance activity above to ensure that Packet filtering rules can be defined for each distinct network interface type supported by the TOE.

The evaluators confirmed that packet filtering rules could be created and assigned to the interface types supported by the TOE.

2.10.1.3 FFW_RUL_EXT.1.5

TSS	The evaluator shall verify that the TSS identifies the protocols that support stateful session handling. The TSS shall identify TCP, UDP, and ICMP if selected by the ST author.

The TOE accepts network packets if it matches an established TCP, UDP or ICMP session using:

- TCP: source and destination addresses, source and destination ports, sequence number, flags
- UDP: source and destination addresses, source and destination ports
- ICMP: source and destination addresses, type, code

TSS	The evaluator shall verify that the TSS describes how stateful sessions are established (including handshake processing) and maintained.

The TOE supports FTP (RFC 959) to dynamically establish sessions allowing network traffic according to Administrator rules. Session events will be logged in accordance with ‘log’ operations defined in the rules. Source and destination addresses, source and destination ports, transport layer protocol, and TOE Interface are recorded in each log record.

TSS	The evaluator shall verify that for TCP, the TSS identifies and describes the use of the following attributes in session determination: source and destination addresses, source and destination ports, sequence number, and individual flags.

The TOE accepts network packets if it matches an established TCP, UDP or ICMP session using:

- TCP: source and destination addresses, source and destination ports, sequence number, flags
- UDP: source and destination addresses, source and destination ports
- ICMP: source and destination addresses, type, code

TSS	The evaluator shall verify that for UDP, the TSS identifies and describes the following attributes in session determination: source and destination addresses, source and destination ports.

The TOE accepts network packets if it matches an established TCP, UDP or ICMP session using:

- TCP: source and destination addresses, source and destination ports, sequence number, flags
- UDP: source and destination addresses, source and destination ports
- ICMP: source and destination addresses, type, code

TSS	The evaluator shall verify that for ICMP (if selected), the TSS identifies and describes the following attributes in session determination: source and destination addresses, other attributes chosen in FFW_RUL_EXT.1.5.

The TOE accepts network packets if it matches an established TCP, UDP or ICMP session using:

- TCP: source and destination addresses, source and destination ports, sequence number, flags
- UDP: source and destination addresses, source and destination ports
- ICMP: source and destination addresses, type, code

TSS	The evaluator shall verify that the TSS describes how established stateful sessions are removed. The TSS shall describe how connections are removed for each protocol based on normal completion and/or timeout conditions. The TSS shall also indicate when session removal becomes effective (e.g., before the next packet that might match the session is processed).

The TOE will remove existing traffic flows due to session inactivity timeout, or completion of the session.

Junos implements what is referred to as an Application Layer gateway (ALG) that inspects FTP traffic to determine the port number used for data sessions. The ALG permits data traffic for the duration of the session, closing the port when the session ends. In this context, "session" refers to the TCP data transfer connection, not the duration of the FTP control session. Junos implements ALGs for a number of protocols.

Guidance	The evaluator shall verify that the guidance documentation describes stateful session behaviours. For example, a TOE might not log packets that are permitted as part of an existing session.

Administrators may assign the "session-init" and "session-close" log operations to a security flow policy. When these clauses are in place, the TOE will log all session establishment and closedown actions associated with dynamic sessions. Source and destination addresses, source and destination ports, transport layer protocol, and TOE Interface are recorded in each log record.

Traffic received as part of an existing session will be captured by the Firewall log, but not included in Syslog.

Testing	The evaluator shall configure the TOE to permit and log TCP traffic. The evaluator shall initiate a TCP session. While the TCP session is being established, the evaluator shall introduce session establishment packets with incorrect flags to determine that the altered traffic is not accepted as part of the session (i.e., a log event is generated to show the ruleset was applied). After a TCP session is successfully established, the evaluator shall alter each of the session determining attributes (source and destination addresses, source and destination ports, sequence number, flags) one at a time in order to verify that the altered packets are not accepted as part of the established session.

The evaluators configured the TOE to permit and log TCP packets. The evaluators then commenced TCP session establishment and, while establishment was underway, introduced additional TCP packets that did not contain the expected flags. The evaluator confirmed that, for each out-of-sequence/malformed packet, the TOE did not accept the packets as part of the initially established session and an audit log was generate appropriately.

Evaluators, once the TCP session establishment process had been completed, sent altered packets through the TOE that did not match the established session attributes. The evaluators confirmed that, in each case, a new TCP session was established by the TOE and an audit log was generated appropriately for each event.

Testing	The evaluator shall terminate the TCP session established per Test 1 as described in the TSS. The evaluator shall then immediately send a packet matching the former session definition in order to ensure it is not forwarded through the TOE without being subject to the ruleset.

The evaluators terminated the TCP session created in the previous test. The evaluators then attempted to forward a packet through the TOE that utilised the session identifiers for the previous session. Evaluators confirmed that the TOE did not accept this packet.

Testing	The evaluator shall expire (i.e., reach timeout) the TCP session established per Test 1 as described in the TSS. The evaluator shall then send a packet matching the former session in order to ensure it is not forwarded through the TOE without being subject to the ruleset.

The evaluators allowed the TCP session created in test 1 to time out. The evaluators then attempted to forward a packet through the TOE that utilised the session identifiers for the previous session. Evaluators confirmed that the TOE did not accept this packet.

Testing	The evaluator shall configure the TOE to permit and log UDP traffic. The evaluator shall establish a UDP session. Once a UDP session is established, the evaluator shall alter each of the session determining attributes (source and destination addresses, source and destination ports) one at a time in order to verify that the altered packets are not accepted as part of the established session.

Evaluators configured the TOE to permit all UDP traffic from one subnet to another. The evaluators then established a UDP session between two peers and attempted to transmit traffic as part of the same session but with altered attributes. The evaluators confirmed that the TOE did not permit this traffic as part of the established session and instead created new session entries for each altered packet.

Testing	The evaluator shall expire (i.e., reach timeout) the UDP session established per Test 4 as described in the TSS. The evaluator shall then send a packet matching the former session in order to ensure it is not forwarded through the TOE without being subject to the ruleset.

Evaluators configured the TOE to permit all UDP traffic from one subnet to another. The evaluators then established a UDP session between two peers and allowed the session to time out. Evaluators then transmitted a packet using the expired session and confirmed that the TOE created a new session with new attributes.

Testing	If ICMP is selected, the evaluator shall configure the TOE to permit and log ICMP traffic. The evaluator shall establish a session for ICMP as defined in the TSS. Once an ICMP session is established, the evaluator shall alter each of the session determining attributes (source and destination addresses, other attributes chosen in FFW_RUL_EXT.1.5) one at a time in order to verify that the altered packets are not accepted as part of the established session.

Evaluators configured the TOE to permit all ICMP traffic from one subnet to another. The evaluators then established an ICMP session between two peers and attempted to transmit traffic as part of the same session but with altered attributes. The evaluators confirmed that the TOE did not permit this traffic as part of the established session and instead created new session entries for each altered packet.

Testing	If applicable, the evaluator shall terminate the ICMP session established per Test 6 as described in the TSS. The evaluator shall then immediately send a packet matching the former session definition in order to ensure it is not forwarded through the TOE without being subject to the ruleset.

The evaluators terminated the ICMP session created in the previous test. The evaluators then attempted to forward a packet through the TOE that utilised the session identifiers for the previous session. Evaluators confirmed that the TOE did not accept this packet as part of the previous session.

Testing	The evaluator shall expire (i.e., reach timeout) the ICMP session established per Test 6 as described in the TSS. The evaluator shall then send a packet matching the former session in order to ensure it is not forwarded through the TOE without being subject to the ruleset.

Evaluators configured the TOE to permit all ICMP traffic from one subnet to another. The evaluators then established an ICMP session between two peers and allowed the session to time out. Evaluators then transmitted a packet using the expired session and confirmed that the TOE created a new session with new attributes.

2.10.1.4 FFW_RUL_EXT.1.6

TSS	The evaluator shall verify that the TSS identifies the following as packets that will be automatically dropped and are counted or logged: a) Packets which are invalid fragments, including a description of what constitutes an invalid fragment b) Fragments that cannot be completely re-assembled c) Packets where the source address is defined as being on a broadcast network d) Packets where the source address is defined as being on a multicast network e) Packets where the source address is defined as being a loopback address f) The TSF shall reject 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 reject 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) Packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified i) Other packets defined in FFW_RUL_EXT.1.6

The TSF shall enforce the following default reject rules with logging on all network traffic:

- invalid fragments;
- fragmented IP packets which cannot be re-assembled completely;
- where the source address is defined as being on a broadcast network;
- where the source address is defined as being on a multicast network;
- where the source address is defined as being a loopback address;
- where the source address is a multicast;
- where the source or destination address is defined as being an address "reserved for future use" as specified in RFC 5735 for IPv4;
- where the source or destination address is defined as an "unspecified address" or an address "reserved for future definition and use" as specified in RFC 3513 for IPv6;
- with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified;
- packets are checked for validity. "Invalid fragments" are those that violate these rules:
  - No overlap
  - The total fragments in one packet should not be more than 62 pieces
  - The total length of merged fragments should not larger than 64k
  - All fragments in one packet should arrive in 2 seconds
  - The total queued fragments has limitation, depending on the platform
  - The total number of concurrent fragment processing for different packet has limitations depending on platform

Guidance	The evaluator shall verify that the guidance documentation describes packets that are discarded and potentially logged by default. If applicable protocols are identified, their descriptions need to be consistent with the TSS. If logging is configurable, the evaluator shall verify that applicable instructions are provided to configure auditing of automatically rejected packets.

The Evaluated Configuration Guide provides TOE administrators with the configuration steps necessary to configure the following default rules:

- Default 'deny-all' rule (drop all traffic that doesn't match any other rules);
- Drop invalid fragments and fragmented IP packets;
- Drop packets with spoofed source address;
- Drop packets where the source address is defined on a multicast network, a loopback address, or a multicast address.
- Drop packets where the source or destination address of a packet is a link-local address, an address "reserved for future use" as specified in RFC 5735 for IPv4, an "unspecified address" or an address "reserved for future definition and use" as specified in RFC 3513 for IPv6;
- Drop an illegal or out-of-sequence TCP packet; and
- Drop unassigned IPv6 packets.

All traffic matching the rules specified above is logged by default.

Testing	The evaluator shall test each of the conditions for automatic packet rejection in turn. In each case, the TOE should be configured to allow all network traffic and the evaluator shall generate a packet or packet fragment that is to be rejected. The evaluator shall use packet captures to ensure that the unallowable packet or packet fragment is not passed through the TOE.

The evaluators utilised scapy to generate packets and test each of the automatic packet rejection rules specified in the requirement. The evaluators confirmed that, for each packet/fragment transmitted, the TOE dropped the packet/fragment and generated a corresponding audit log.

Testing	For each of the cases above, the evaluator shall use any applicable guidance to enable dropped packet logging or counting. In each case above, the evaluator shall ensure that the rejected packet or packet fragment was recorded (either logged or an appropriate counter incremented).

The evaluators utilised the 'show security screen statistics zone <zone name>' command to view the packet statistics for each security zone. Evaluators confirmed that, each time a packet was dropped, the applicable counter in the zone statistics was incremented. Evaluators also confirmed that an audit log entry is generated for each dropped packet.

2.10.1.5 FFW_RUL_EXT.1.7

TSS	The evaluator shall verify that the TSS explains how the following traffic can be dropped and counted or logged: a) Packets where the source address is equal to the address of the network interface where the network packet was received b) Packets where the source or destination address of the network packet is a linklocal address c) Packets where the source address does not belong to the networks associated with the network interface where the network packet was received, including a description of how the TOE determines whether a source address belongs to a network associated with a given network interface

The TSF shall enforce the following default reject rules with logging on all network traffic:

- where the source address is equal to the address of the network interface where the network packet was received;
- where the source address does not belong to the networks associated with the network interface where the network packet was received;
- packets where the source or destination address is a link-local address;

The TOE is configured to associate network interfaces to IP subnets. Source IP addresses are then associated with the network interface.

Guidance	The evaluator shall verify that the guidance documentation describes how the TOE can be configured to implement the required rules. If logging is configurable, the evaluator shall verify that applicable instructions are provided to configure auditing of automatically rejected packets.

The Evaluated Configuration Guide provides TOE administrators with the configuration steps necessary to configure the following default rules:

- Default 'deny-all' rule (drop all traffic that doesn't match any other rules);
- Drop invalid fragments and fragmented IP packets;
- Drop packets with spoofed source address;
- Drop packets where the source address is defined on a multicast network, a loopback address, or a multicast address.
- Drop packets where the source or destination address of a packet is a link-local address, an address "reserved for future use" as specified in RFC 5735 for IPv4, an "unspecified address" or an address "reserved for future definition and use" as specified in RFC 3513 for IPv6;
- Drop an illegal or out-of-sequence TCP packet; and
- Drop unassigned IPv6 packets.

All traffic matching the rules specified above is logged by default.

Testing	The evaluator shall configure the TOE to drop and log network traffic where the source address of the packet matches that of the TOE network interface upon which the traffic was received. The evaluator shall generate suitable network traffic to match the configured rule and verify that the traffic is dropped and a log message generated.

The TOE automatically drops and logs network traffic where the source address of the packet matches that of the TOE network interface upon which the traffic was received. The evaluators utilised scapy to craft packets that met this description and attempted to send them through the TOE.

Evaluators confirmed that the TOE did not permit these packets to flow through the TOE and logged these events appropriately.

Testing	The evaluator shall configure the TOE to drop and log network traffic where the source IP address of the packet fails to match the network reachability information of the interface to which it is targeted, e.g. if the TOE believes that network 192.168.1.0/24 is reachable through interface 2, network traffic with a source address from the 192.168.1.0/24 network should be generated and sent to an interface other than interface 2. The evaluator shall verify that the network traffic is dropped and a log message generated.

The TOE automatically drops and logs network traffic where the source IP address of the packet fails to match the network reachability information of the interface to which it is targeted. The evaluators utilised scapy to craft packets that met this description and attempted to send them through the TOE.

Evaluators confirmed that the TOE did not permit these packets to flow through the TOE and logged these events appropriately.

2.10.1.6 FFW_RUL_EXT.1.8

TSS	The evaluator shall verify that the TSS describes the algorithm applied to incoming packets, including the processing of default rules, determination of whether a packet is part of an established session, and application of administrator defined and ordered ruleset.

The Information Flow subsystem is responsible for processing the arriving packets from the network to the TOE's network interface. Based on Administrator-configured policy, interface and zone information, the packet flows through the various modules of the Information Flow subsystem. Rules within policies are processed in an Administrator-defined order when network traffic flows through the TOE network interfaces.

By default, the TOE behavior is to deny packets when there is no rule match unless another required condition allows the network traffic. If a security risk is found in the packet. (e.g. denial-of-service attacks), the packet is dropped and an event is logged. The packet does not continue to the next module for processing. If the packet is not dropped by a given module, the interrupt handling routine calls the function for the next relevant module.

The IP Classification module retrieves information from packets received on the network interface device, classifies packets into several categories, saves classification information in packet processing context, and provides other modules with that information for assisting further processing.

The Attack Detection module provides inline attack detection such as IP Spoofing for the security appliance. This module monitors arriving traffic, performs predefined attack detection services (prevents attacks), and issues actions when an attack is found.

The Session Lookup module performs lookups in the session table that is used for all interfaces based on the information in incoming packets. Specifically, the lookup is based on the exact match of source IP address and port, destination IP address and port, protocol attributes (e.g., SYN, ACK, RST, and FIN), and egress/ingress zone. The input is passed to the module as a set of parameters from the Attack Detection module via a function call. The module returns matching wing if a match is found and 0 otherwise. Sessions are removed when terminated.

The Session Setup module is only available for packets that do not match current established sessions. It is activated after the Session Lookup module. If packet has a matched session, it will skip the session setup module and proceed to the Security Policy module, and other modules. Eventually if the packet is not destined for the TOE, the Network interface will pass the traffic out of the appliance.

The Security Policy module examines traffic passing through the TOE (via Session Setup module) and determines if the traffic can pass based on administrator-configured access policies. The Security Policy module is the core of the firewall and IPS functionalities in the TOE: It is the policy enforcement engine that fulfils the security requirements for the user. The Security Policy module will deny packets when there is no policy match unless another policy allows the traffic.

The Session Setup module performs the auditing of denied packets. If there is a policy to specifically deny traffic, traffic matching this deny policy is dropped and logged in traffic log. If there is no policy to deny traffic, traffic that does not match any policy is dropped and not logged. In either case, Session Setup module does not create any sessions for denied traffic. Sessions are created for allowed traffic.

The INETD module provides internet services for the TOE. The module listens on designated ports used by internet services such as FTP. When a TCP or UDP packet arrives with a particular destination port number, INETD launches the appropriate server program (e.g., SSHD) to handle the connection.

The RPD (Routing Protocol Daemon) module provides the implementations and algorithms for the routing protocols and route calculations. The primary goal of the RPD is to create and maintain the Routing Information Base (RIB), which is a database of routing entries. Each routing entry consists of a destination address and some form of next hop information. RPD module maintains the internal routing table and properly distributes routes from the routing table to Kernel subsystem used for traffic forwarding at the Network interface

Guidance	The evaluator shall verify that the guidance documentation describes how the order of stateful traffic filtering rules is determined and provides the necessary instructions so that an administrator can configure the order of rule processing.

By default, traffic filtering rules are enforced in a top-to-bottom order of precedence (i.e. terms are examined sequentially). However, if the administrator wishes to change the order of the rules in place, the following command (from the CLI User Guide) allows them to do so:

```
insert <statement-path> identifier1 (before | after) identifier2
```

Testing	The evaluator shall devise two equal stateful traffic filtering rules with alternate operations – permit and drop. The rules should then be deployed in two distinct orders and in each case the evaluator shall ensure that the first rule is enforced in both cases by generating applicable packets and using packet capture and logs for confirmation.

The evaluators constructed two packet filtering rules, the only difference between the two being the action to be taken (permit/deny).

The evaluators confirmed that rules are processed in the order in which they are in defined. Wireshark monitoring and audit log examination corroborated this finding.

Testing	The evaluator shall repeat the procedure above, except that the two rules should be devised where one is a subset of the other (e.g., a specific address vs. a network segment). Again, the evaluator should test both orders to ensure that the first is enforced regardless of the specificity of the rule.

The evaluators constructed two packet filtering rules, one being a subset of the other (10.0.2.0/24 and 10.0.2.2).

The evaluators confirmed that rules are processed in the order in which they are in defined. Wireshark monitoring and audit log examination corroborated this finding.

2.10.1.7 FFW_RUL_EXT.1.9

TSS	The evaluator shall verify that the TSS describes the process for applying stateful traffic filtering rules and also that the behavior (either by default, or as configured by the administrator) is to deny packets when there is no rule match unless another required conditions allows the network traffic (i.e., FFW_RUL_EXT.1.5 or FFW_RUL_EXT.2.1).

The Information Flow subsystem is responsible for processing the arriving packets from the network to the TOE's network interface. Based on Administrator-configured policy, interface and zone information, the packet flows through the various modules of the Information Flow subsystem. Rules within policies are processed in an Administrator-defined order when network traffic flows through the TOE network interfaces.

By default, the TOE behavior is to deny packets when there is no rule match unless another required condition allows the network traffic. If a security risk is found in the packet (e.g. denial-of-service attacks), the packet is dropped and an event is logged. The packet does not continue to the next module for processing. If the packet is not dropped by a given module, the interrupt handling routine calls the function for the next relevant module.

Guidance	The evaluator shall verify that the guidance documentation describes the behavior if no rules or special conditions apply to the network traffic. If the behavior is configurable, the evaluator shall verify that the guidance documentation provides the appropriate instructions to configure the behavior to deny packets with no matching rules.

Per the Evaluated Configuration Guide, the default reject-all rule can be implemented via the following command:

```
set security policies default-policy deny-all
```

Testing	For each attribute in FFW_RUL_EXT.1.2, the evaluator shall construct a test to demonstrate that the TOE can correctly compare the attribute from the packet header to the ruleset, and shall demonstrate both the permit and deny for each case. The evaluator shall check the log in each case to confirm that the relevant rule was applied. The evaluator shall record a packet capture for each test to demonstrate the correct TOE behaviour.

The evaluators completed this test objective as part of FFW_RUL_EXT.1 and FPF_RUL_EXT.1 test activities. The evaluators confirmed that the TOE is able to permit/drop and log packets appropriately for all applicable protocols and attributes.

2.10.1.8 FFW_RUL_EXT.1.10

TSS	The evaluator shall verify that the TSS describes how the TOE tracks and maintains information relating to the number of half-open TCP connections. The TSS should identify how the TOE behaves when the administratively defined limit is reached and should describe under what circumstances stale half-open connections are removed (e.g. after a timer expires).

The TOE can be configured to drop connection attempts after a defined number of half-open TCP connections using the Junos screen 'tcp syn-flood', which provides both source and destination thresholds on the number of uncompleted TCP connections, as well as a timeout period.

The source threshold option allows administrators to specify the number of SYN segments received per second from a single source IP address—regardless of the destination IP address—before Junos OS begins dropping connection requests from that source.

Similarly, the destination threshold option allows administrators to specify the number of SYN segments received per second for a single destination IP address before Junos OS begins dropping connection requests to that destination.

The timeout option allows administrators to set the maximum length of time before an uncompleted connection is dropped from the queue.

Guidance	The evaluator shall verify that the guidance documentation describes the behaviour of imposing TCP half-open connection limits and its default state if unconfigured. The evaluator shall verify that the guidance clearly indicates the conditions under which new connections will be dropped e.g. per-destination or per-client.

Per the ECG and the IPS Feature Guide, the SYN Flood attack screen can be used to set a limit on half-open connection states (this is not configured by default).

The administrator may set a limit (e.g. 1000 half-open connections) using the following command:

```
set security screen ids-option zone-syn-flood tcp syn-flood source-threshold 1000
```

This limit can also be set on a destination-based metric.

The TOE will automatically drop all SYN packets received above this threshold unless configured otherwise.

Testing	The evaluator shall define a TCP half-open connection limit on the TOE. The evaluator shall generate TCP SYN requests to pass through the TOE to the target system using a randomised source IP address and common destination IP address. The number of SYN requests should exceed the TCP half-open threshold defined on the TOE. TCP SYN-ACK messages should not be acknowledged. The evaluator shall verify through packet capture that once the defined TCP half-open threshold has been reached, subsequent TCP SYN packets are not transmitted to the target system. The evaluator shall verify that when the configured threshold is reached that, depending upon the selection, either a log entry is generated or a counter is incremented.

The evaluators used the ids-option configuration to define a syn-flood (half-open TCP connection) limit of 1000 packets. Evaluators then began flooding the destination subnet with TCP SYN packets.

Evaluators confirmed via zone counter statistics and the audit log that, once 1000 packets had been received, the TOE dropped all subsequent packets.

2.10.2 FFW_RUL_EXT.2 Stateful Filtering of Dynamic Protocols

TSS	The evaluator shall verify that the TSS identifies the protocols that can cause the automatic creation of dynamic packet filtering rules. In some cases rather than creating dynamic rules, the TOE might establish stateful sessions to support some identified protocol behaviors.

The TOE supports FTP (RFC 959) to dynamically establish sessions allowing network traffic according to Administrator rules. Session events will be logged in accordance with 'log' operations defined in the rules. Source and destination addresses, source and destination ports, transport layer protocol, and TOE Interface are recorded in each log record.

TSS	The evaluator shall verify that the TSS explains the dynamic nature of session establishment and removal. The TSS also shall explain any logging ramifications.

Junos implements what is referred to as an Application Layer gateway (ALG) that inspects FTP traffic to determine the port number used for data sessions. The ALG permits data traffic for the duration of the session, closing the port when the session ends. In this context, "session" refers to the TCP data transfer connection, not the duration of the FTP control session. Junos implements ALGs for a number of protocols.

TSS	The evaluator shall verify that for each of the protocols selected, the TSS explains the dynamic nature of session establishment and removal specific to the protocol.

The TOE will remove existing traffic flows due to session inactivity timeout, or completion of the session.

Guidance	The evaluator shall verify that the guidance documentation describes dynamic session establishment capabilities.

Per Chapter 11 (Configuring Traffic Filtering Rules) of the ECG, the TOE can be configured to permit or deny dynamic FTP sessions. These sessions are handled in an identical manner to all other protocols supported by the TOE that are relevant to the evaluation.

Sessions can be permitted/denied, both session initialisation and close can be logged and limits can be put in place to prevent dynamic sessions above a defined threshold.

Guidance	The evaluator shall verify that the guidance documentation describes the logging of dynamic sessions consistent with the TSS.

Per the various traffic policy configuration examples provided in the ECG, Session initialisation and closure are logged per the applicable traffic policy that the dynamic session is associated with.

Testing	The evaluator shall define stateful traffic filtering rules to permit and log traffic for each of the supported protocols and drop and log TCP and UDP ports above 1024. Subsequently, the evaluator shall establish a connection for each of the selected protocols in order to ensure that it succeeds. The evaluator shall examine the generated logs to verify they are consistent with the guidance documentation.

The evaluators configured the TOE per the requirement (permit FTP, drop TCP/UDP > 1024). The evaluators confirmed that an FTP session with a peer was established and logged.

Testing	Continuing from Test 1, the evaluator shall determine (e.g., using a packet sniffer) which port above 1024 opened by the control protocol, terminate the connection session, and then verify that TCP or UDP (depending on the protocol selection) packets cannot be sent through the TOE using the same source and destination addresses and ports.

The evaluators utilised Wireshark to determine the port number above 1024 being used by the FTP session established in the previous test. Evaluators terminated the session and attempted to transmit additional packets using the same high-number port. Evaluators confirmed that the TOE did not permit the packet to reach its destination.

Testing	For each additionally supported protocol, the evaluator shall repeat the procedure above for the protocol. In each case the evaluator must use the applicable RFC or standard in order to determine what range of ports to block in order to ensure the dynamic rules are created and effective.

The TOE supports dynamic session establishment for FTP only.

2.11 Intrusion Prevention (IPS)

2.11.1 IPS_NTA_EXT.1 Network Traffic Analysis

2.11.1.1 IPS_NTA_EXT.1.1

TSS	The evaluator shall verify that the TSS explains the TOE's capability of analyzing IP traffic in terms of the TOE's policy hierarchy (precedence).

An IDP policy is made up of rule bases, and each rule base contains a set of rules that specify rule parameters, such as traffic match conditions, action, and logging requirements. IDP policies can then be associated to firewall policies. IDP can be invoked on a firewall rule-by-rule basis for maximum granularity. Only firewall policies marked for IDP will be processed by IDP engine, all other rules will only be processed by the firewall.

Rules within policies are processed in an Administrator-defined order when network traffic flows through the TOE network interfaces.

TSS	The TSS should identify if the TOE's policy hierarchy order is configurable by the administrator for IPS policy elements (known-good lists, known-bad lists, signature-based rules, and anomaly-based rules).

Rules within policies are processed in an Administrator-defined order when network traffic flows through the TOE network interfaces.

TSS	Regardless of whether the precedence is configurable, the evaluator shall verify that the TSS describes the default precedence as well as the IP analyzing functions supported by the TOE.

Firewall Policies match Source Zone, Destination Zone, Source IP, Destination IP, Source Port, Destination Port, and Protocol. Interface and VLAN matching can be achieved through the use of zones. Rules are organized into a firewall policy rulebase. Within IPS Policies, further matching for specific attacks is done on Source Zone, Destination Zone, Source IP, Destination IP, Source Port, Destination Port, and Protocol. Interface matching can be achieved through the use of zones. Attack Actions are configurable on a rule by rule basis. Rules within policies are processed in an Administrator-defined order when network traffic flows through the TOE network interfaces.

Once stateful firewall processing of packets has been performed by the Information Flow subsystem, if a firewall policy that has been marked for IDP processing is triggered, the packets are processed by the IPS subsystem as follows:

- Fragmentation Processing – IP Fragments are reordered and reassembled. Duplicate, over/undersized, overlapping, incomplete and other invalid fragments are discarded.
- Flow Module SSL Decryption – sessions are checked for existing IP Actions, if none exists, new sessions are created. If a destination is marked for SSL decryption, a copy of the SSL traffic will be sent to the decryption engine. The original packet will be queue until inspection is complete.
- Packet Serialization and TCP Reassembly – packets are ordered and all TCP packets are reassembled into complete application messages.
- Application ID – pattern matching is performed on the traffic to determine what application the traffic is. The traffic is still inspected for Attacks, even if application cannot be determined.
- Protocol Decoding – protocol parsing and decoding is performed. Messages are deconstructed into application "contexts" which identify components of messages. Protocol Anomaly Detection is performed, along with AppDoS (if configured) by thresholds of these contexts.
- Attack Signature Matching – signatures are detected via deterministic finite automaton (DFA) pattern matching.

- IDP Attack Actions – when an attack is detected the corresponding policy configured action is executed.

Guidance	The evaluator shall verify that the guidance describes the default precedence. If the precedence is configurable. The evaluator shall verify that the guidance explains how to configure the precedence.

By default, rules are enforced in a top-to-bottom order of precedence (i.e. terms are examined sequentially). However, if the administrator wishes to change the order of the rules in place, the following command (from the CLI User Guide) allows them to do so:

```
insert <statement-path> identifier1 (before | after) identifier2
```

Testing	N/A

N/A

2.11.1.2 IPS_NTA_EXT.1.2

TSS	The evaluator shall verify that the TSS indicates that the following protocols are supported: • IPv4 • IPv6 • ICMPv4 • ICMPv6 • TCP • UDP

The TOE performs stateful network traffic filtering on network packets using the following network traffic protocols and network fields conforming to the described RFCs:

- Internet Control Message Protocol version 4 (ICMPv4)
  - RFC 792 (ICMPv4)
- Internet Control Message Protocol version 6 (ICMPv6)
  - RFC 4443 (ICMPv6)
- Internet Protocol (IPv4)
  - RFC 791 (IPv4)
- Internet Protocol version 6 (IPv6)
  - RFC 2460 (IPv6)
- Transmission Control Protocol (TCP)
  - RFC 793 (TCP)
- User Datagram Protocol (UDP)
  - RFC 768 (UDP)

TSS	The evaluator shall verify that the TSS describes how conformance with the identified protocols has been determined by the TOE developer. (e.g., third party interoperability testing, protocol compliance testing)

Conformance to these RFCs is demonstrated by protocol compliance testing by the product QA team.

Guidance	The Guidance associated with this requirement is assessed in the subsequent assurance activities.

N/A

Testing	N/A

N/A

2.11.1.3 IPS_NTA_EXT.1.3

TSS	The evaluator shall verify that the TSS identifies all interface types capable of being deployed in the modes of promiscuous, and or inline mode as well as the interfaces necessary to facilitate each deployment mode (at a minimum, the interfaces need to support inline mode).

The TOE is capable of inspecting all traffic passing through the TOE’s Ethernet interfaces (inline mode). Ethernet interfaces can be assigned to Zones on which firewall and IDP policies are predicated.

TSS	The TSS should also provide descriptions how the management interface is distinct from sensor interfaces.

The TOE supports management through the console port, as well as through a dedicated Ethernet management port whose traffic is never processed for routing. Remote management of the TOE can also be performed via SSH as described in Section 7.1.3.

Guidance	The evaluator shall verify that the operational guidance provides instructions on how to deploy each of the deployment methods outlined in the TSS. The evaluator shall also verify that the operational guidance provides instructions of applying IPS policies to interfaces for each deployment mode. If the management interface is configurable the evaluator shall verify operational guidance explains how to configure the interface into a management interface.

The TOE’s primary interfaces are deployed in Inline mode by default and, thus, do not require any additional configuration. No interfaces can operate in Promiscuous mode. Ethernet interfaces can be configured into management interfaces by permitting SSH on that interface.

IDP rules are not explicitly tied to a single interface – the engine listens on all interfaces. Rather, rules are assigned a “to zone” and “from zone” – the engine monitors all inbound traffic and, when a match is found, the rules are enforced as configured.

Guidance	The evaluator shall verify that the operational guidance explains how the TOE sends commands to remote traffic filtering devices.

The TOE does not utilise remote filtering devices.

Testing	N/A

N/A

2.11.2 IPS_IPB_EXT.1 IP Blocking

TSS	The evaluator shall verify how good/bad lists affect the way in which traffic is analyzed with respect to processing packets. The TSS should also provide detail with the attributes that create a known good list, a known bad list, their associated rules, including how to define the source or destination IP address (e.g. a single IP address or a range of IP addresses).

The TOE supports the definition of known-good and known-bad lists of source and/or destination addresses at the firewall rule level as described in Section 7.8. Address ranges can be defined by creating address book entries and attaching them to firewall policies.

TSS	The evaluator shall also verify that the TSS identifies all the roles and level of access for each of those roles that have been specified in the requirement.

The Security Administrator has the capability to:

- Perform management functions
    - Enable, disable signatures applied to sensor interfaces, and determine the behaviour of IPS functionality

- Modify these parameters that define the network traffic to be collected and analysed:
  - Source IP addresses (host address and network address);
  - Destination IP addresses (host address and network address);
  - Source port (TCP and UDP);
  - Destination port (TCP and UDP);
  - Protocol (IPv4 and IPv6)
  - ICMP type and code

Guidance	The evaluator shall verify that the administrative guidance provides instructions with how each role specified in the requirement can create, modify and delete the attributes of a known good and known bad lists.

The administrator may assign individual IP addresses, ranges or entire subnets to the Address Book – these address book entries may then be assigned to security flow policies, which may be configured to permit or deny traffic to/from the relevant IP addresses.

Testing	The evaluator shall use the instructions in the operational guidance to create a known-bad address list. Using a single IP address, a list of addresses or a range of addresses from that list, the evaluator shall attempt to send traffic through the TOE that would otherwise be allowed by the TOE and observe the TOE automatically drops that traffic.

The evaluators used the TOE address book and security policy functions to create a bad-address list. The evaluators confirmed that the TOE did not permit traffic to flow to address on the 'bad' list.

Testing	The evaluator shall use the instructions in the operational guidance to create a known-good address list. Using a single IP address, a list of addresses or a range of addresses from that list, the evaluator shall attempt to send traffic that would otherwise be denied by the TOE and observe the TOE automatically allowing traffic.

The evaluators used the TOE address book and security policy functions to create a good-address list. The evaluators confirmed that the TOE did not permit traffic to flow to address on the 'good' list.

Testing	The evaluator shall add conflicting IP addresses to each list and ensure that the TOE handles conflicting traffic in a manner consistent with the precedence in IPS_NTA_EXT.1.1.

The evaluators created security policies with conflicting good-bad lists. The evaluators confirmed that the TOE enforced security policies in the administrator-defined order.

2.11.3 IPS_SBD_EXT.1 Signature-Based IPS Functionality

2.11.3.1 IPS_SBD_EXT.1.1

TSS	The evaluator shall verify that the TSS describes what is comprised within a signature rule.

The TOE supports stateful signature based attack detection defined as Attack Objects. Attack Objects use context based matching to match regular expressions in specific locations where they occur. Attack Objects can be composed of multiple signatures and protocol anomalies, including logical expressions between signatures for compound matching.

TSS	The evaluator shall verify that each signature can be associated with a reaction specified in IPS_SBD_EXT.1.5.

Signatures can be defined to match the any of above header-field values, using the command "set security idp custom-attack", along with the actions (allow/block), using the command "set security idp idp-policy", that the TOE will perform when a match is found in the processed packets.

TSS	The evaluator shall verify that the TSS identifies all interface types capable of applying signatures and explains how rules are associated with distinct network interfaces. Where interfaces can be grouped into a common interface type (e.g., where the same internal logical path is used, perhaps where a common device driver is used) they can be treated collectively as a distinct network interface.

The TOE is capable of inspecting all traffic passing through the TOE's Ethernet interfaces (inline mode). Ethernet interfaces can be assigned to Zones on which firewall and IDP policies are predicated.

Guidance	The evaluator shall verify that the operational guidance provides instructions with how to create and/or configure rules using the following protocols and header inspection fields: • IPv4: Version; Header Length; Packet Length; ID; IP Flags; Fragment Offset; Time to Live (TTL); Protocol; Header Checksum; Source Address; Destination Address; and IP Options. • IPv6: Version; traffic class; flow label; payload length; next header; hop limit; source address; destination address; routing header; home address options. • ICMP: Type; Code; Header Checksum; and Rest of Header (varies based on the ICMP type and code). • ICMPv6: Type; Code; and Header Checksum. • TCP: Source port; destination port; sequence number; acknowledgement number; offset; reserved; TCP flags; window; checksum; urgent pointer; and TCP options. • UDP: Source port; destination port; length; and UDP checksum.

The IDP Feature Guide provides administrators with configuration examples and guidance on how to develop and implement custom IDP rules using each of the protocols and fields specified above.

Guidance	The evaluator shall verify that the operational guidance provides instructions with how to select and/or configure reactions specified in IPS_SBD_EXT.1.5 in the signature rules.

The IDP Feature Guide provides administrators with command examples and configuration data on how to configure an IDP rule to perform allow or drop operations.

Testing	The evaluator shall use the instructions in the operational guidance to test that packet header signatures can be created and/or configured with the selected and/or configured reactions specified in IPS_SBD_EXT.1.5 for each of the attributes listed below. Each attribute shall be individually assigned to its own unique signature: • IPv4: Version; Header Length; Packet Length; ID; IP Flags; Fragment Offset; Time to Live (TTL); Protocol; Header Checksum; Source Address; Destination Address; and IP Options. • IPv6: Version; traffic class; flow label; payload length; next header; hop limit; source address; destination address; routing header; home address options. • ICMP: Type; Code; Header Checksum; and Rest of Header (varies based on the ICMP type and code). • ICMPv6: Type; Code; and Header Checksum;. • TCP: Source port; destination port; sequence number; acknowledgement number; offset; reserved; TCP flags; window; checksum; urgent pointer; and TCP options. • UDP: Source port; destination port; length; and UDP checksum. Using packet sniffers, the evaluator will generate traffic to trigger a signature and using packet captures will ensure that the reactions of each rule are performed as expected.

Evaluators developed an IDP signature for each protocol and attribute specified above. The evaluators utilised scapy to generate packets for each relevant signature and utilised Wireshark and audit log analysis to determine whether the TOE reacted as configured. The evaluators confirmed that the TOE allowed each applicable reaction to be assigned to each signature.

Testing	Repeat the test assurance activity above to ensure that signature-based IPS policies can be defined for each distinct network interface type capable of applying signatures as supported by the TOE.

Evaluators confirmed during testing that IPS signatures could be applied to security zones that can then be associated with any applicable interface type.

2.11.3.2 IPS_SBD_EXT.1.2

TSS	The evaluator shall verify that the TSS describes what is comprised within a string-based detection signature.

The TOE also supports string-based pattern-matching inspection of packet payload data for the above listed protocols. For TCP payload inspection, Junos OS provides pre-defined attack signatures to detect FTP commands, HTTP commands and content, and STMP states. Alternative, administrators can define custom-attack signatures for these application layer protocols using the command "set security idp custom-attack".

TSS	The evaluator shall verify that each packet payload string-based detection signature can be associated with a reaction specified in IPS_SBD_EXT.1.5.

Signatures can be defined to match the any of above header-field values, using the command "set security idp custom-attack", along with the actions (allow/block), using the command "set security idp idp-policy", that the TOE will perform when a match is found in the processed packets.

Guidance	The evaluator shall verify that the operational guidance provides instructions with how to configure rules using the packet payload string-based detection fields defined in IPS_SBD_EXT.1.2. The operational guidance shall provide configuration instructions, if needed, to detect payload across multiple packets.

The IDP Feature Guide provides administrators with configuration examples and guidance on how to develop and implement custom IDP rules using each of the payload string-based detection fields (e.g. FTP, SMTP) defined in IPS_SBD_EXT.1.2.

Testing	The evaluator shall use the instructions in the operational guidance to test that packet payload string-based detection rules can be assigned to the reactions specified in IPS_SBD_EXT.1.5 using the attributes specified in IPS_SBD_EXT.1.2. However it is not required (nor is it feasible) to test all possible strings of protocol data, the evaluator shall ensure that a selection of strings in the requirement is selected to be tested. At a minimum at least one string using each of the following attributes from IPS_SBD_EXT.1.2 should be tested for each protocol. The evaluator shall generate packets that match the string in the rule and observe the corresponding reaction is as configured. • Test at least one string of characters for ICMPv4 data: beyond the first 4 bytes of the ICMP header. • Test at least one string of characters for ICMPv6 data: beyond the first 4 bytes of the ICMP header. • TCP data (characters beyond the 20 byte TCP header): o Test at least one FTP (file transfer) command: help, noop, stat, syst, user, abort, acct, allo, appe, cdup, cwd, dele, list, mkd, mode, nlst, pass, pasv, port, pass, quit, rein, rest, retr, rmd, rnfr, rnto, site, smnt, stor, stou, stru, and type. o HTTP (web) commands and content: ▪ Test both GET and POST commands ▪ Test at least one administrator-defined strings to match URLs/URIs, and web page content. o Test at least one SMTP (email) state: start state, SMTP commands state, mail header state, mail body state, abort state. o Test at least one string in any additional attribute type defined within [selection: [assignment: other types of TCP payload inspection]; • Test at least one string of UDP data: characters beyond the first 8 bytes of the UDP header; • Test at least one string for each additional attribute type defined in [assignment: other types of packet payload inspection]]

The evaluators developed IDP signatures for ICMPv4/v6 and UDP header data, the FTP 'nlst' command, HTTP GET/POST commands, URL and body content and the SMTP 'body' state.

Evaluators configured each detection rule, in turn, with each of the available reactions and confirmed that a) the TOE is able to detect each of the data types specified in the requirement; and b) react to each traffic type as configured.

Testing	The evaluator shall repeat one of the tests in Test 1 but generate multiple non-fragmented packets that contain the string in the rule defined.

The evaluators repeated the UDP header data test by transmitting 10+ non-fragmented packets that contained the string configured in the detection rule. Evaluators confirmed that the TOE identified the string across all of the non-fragmented packets.

Testing	Repeat the test assurance activity above to ensure that signature-based IPS policies can be defined for each distinct network interface type capable of applying signatures as supported by the TOE.

Evaluators confirmed during testing that IPS signatures could be applied to security zones that can then be associated with any applicable interface type.

2.11.3.3 IPS_SBD_EXT.1.3

TSS	The evaluator shall verify that the TSS describes how the attacks defined in IPS_SBD_EXT.1.3 are processed by the TOE and what reaction is triggered when these attacks are identified.

The TOE is capable of detecting the following signatures using Junos predefined screen options:

- IP Fragments Overlap (Teardrop attack, Bonk attack, or Boink attack)
- IP source address equal to the IP destination (Land attack)
- Fragmented ICMP Traffic (e.g. Nuke attack)
- Large ICMP Traffic (Ping of Death attack)
- TCP NULL flags
- TCP SYN+FIN flags
- TCP FIN only flags
- UDP Bomb Attack
- ICMP flooding (Smurf attack, and ping flood)
- TCP flooding (e.g. SYN flood)
- IP protocol scanning
- TCP port scanning
- UDP port scanning
- ICMP scanning

The default action for the above screens is to drop the packets. To allow the packets through, the "alarm-without-drop" action can be defined using the command "set security screen idsoption".

The TOE is also capable of detecting the following signatures:

- TCP SYN+RST flags, by defining an custom attack to match "protocol tcp tcp-flags rst" and "protocol tcp tcp-flags syn"[77];
- UDP Chargen DoS attack , by configuring a firewall policy to match the predefined "junos-chargen" with the desired allow/block reaction;
- Flooding of a network (DoS attack), by the configuration of policers that allow establishing prioritization and bandwidth limits for different type of network traffic.

Guidance	The evaluator shall verify that the operational guidance provides instructions with configuring rules to identify the attacks defined in IPS_SBD_EXT.1.3 as well as the reactions to these attacks as specified in IPS_SBD_EXT.1.5.

The Evaluated Configuration Guide provides instructions on how to configure security screens for each of the attacks specified in the requirement. It also provides instructions on how to configure the TOE to block and alert if a match is found, or to ignore the event and allow the traffic to flow through the TOE.

Testing	The evaluator shall create and/or configure rules for each attack signature in IPS_SBD_EXT.1.3. For each attack, the TOE should apply its corresponding signature and enable it to each distinct network interface type capable of applying the signatures. The evaluator shall use packet captures to ensure that the attack traffic is detected by the TOE and a reaction specified in IPS_SBD_EXT.1.5 is triggered and stops the attack. Each attack should be performed one after another so as to ensure that its corresponding signature successfully identified and appropriately reacted to a particular attack.

The evaluators created new attack signatures and/or configured the IDP Attack Screen to detect each of the applicable traffic types defined in this requirement.

Evaluators transmitted traffic through the TOE matching each of the signature types in this requirement and confirmed that, in each case, the TOE identified the malicious traffic and reacted as configured.

2.11.3.4 IPS_SBD_EXT.1.4

TSS	The evaluator shall verify that the TSS describes how the attacks defined in IPS_SBD_EXT.1.4 are processed by the TOE and what reaction is triggered when these attacks are identified.

The TOE is capable of detecting the following signatures using Junos predefined screen options:

- IP Fragments Overlap (Teardrop attack, Bonk attack, or Boink attack)
- IP source address equal to the IP destination (Land attack)
- Fragmented ICMP Traffic (e.g. Nuke attack)
- Large ICMP Traffic (Ping of Death attack)
- TCP NULL flags
- TCP SYN+FIN flags
- TCP FIN only flags
- UDP Bomb Attack
- ICMP flooding (Smurf attack, and ping flood)
- TCP flooding (e.g. SYN flood)
- IP protocol scanning
- TCP port scanning
- UDP port scanning
- ICMP scanning

The default action for the above screens is to drop the packets. To allow the packets through, the "alarm-without-drop" action can be defined using the command "set security screen idsoption".

The TOE is also capable of detecting the following signatures:

- TCP SYN+RST flags, by defining an custom attack to match "protocol tcp tcp-flags rst" and "protocol tcp tcp-flags syn"77;
- UDP Chargen DoS attack , by configuring a firewall policy to match the predefined "junos-chargen" with the desired allow/block reaction;
- Flooding of a network (DoS attack), by the configuration of policers that allow establishing prioritization and bandwidth limits for different type of network traffic.

Guidance	The evaluator shall verify that the operational guidance provides instructions with configuring rules to identify the attacks defined in IPS_SBD_EXT.1.4 as well as the reactions to these attacks as specified in IPS_SBD_EXT.1.5.

The Evaluated Configuration Guide provides instructions on how to configure security screens for each of the attacks specified in the requirement. It also provides instructions on how to configure the TOE to block and alert if a match is found, or to ignore the event and allow the traffic to flow through the TOE.

Testing	The evaluator shall configure individual signatures for each attack in IPS_SBD_EXT.1.4. For each attack, the TOE should apply its corresponding signature and enable it to each distinct network interface type capable of applying signatures. The evaluator shall use packet captures to ensure that the attack traffic is detected by the TOE and a reaction specified in IPS_SBD_EXT.1.5 is triggered and stops the attack. Each attack should be performed one after another so as to ensure that its corresponding signature successfully identified and appropriately reacted to a particular attack.

The evaluators created new attack signatures and/or configured the IDP Attack Screen to detect each of the applicable traffic types defined in this requirement.

Evaluators transmitted traffic through the TOE matching each of the signature types in this requirement and confirmed that, in each case, the TOE identified the malicious traffic and reacted as configured.

2.11.3.5 IPS_SBD_EXT.1.5

TSS	N/A

IDP Attack Actions – when an attack is detected the corresponding policy configured action is executed. Possible actions include:

- No Action
- Drop packet
- Drop connection
- Close client (send an RST packet to the client)
- Close server (sends an RST packet to the server)
- Close client and server (sends an RST packet to both client and server)

Guidance	N/A

N/A

Testing	N/A

N/A

2.11.4 IPS_ABD_EXT.1 Anomaly-Based IPS

2.11.4.1 IPS_ABD_EXT.1.1

TSS	The evaluator shall verify that the TSS describes the composition, construction, and application of baselines or anomaly-based attributes specified in IPS_ABD_EXT.1.1.

The TOE allows administrators to define signatures for anomalous traffic in terms of throughput (bits per second), time of the day for defined source/destination address and source/destination port, frequency of traffic patterns and thresholds of traffic patterns.

Anomaly signatures based on time of day characteristics are implemented by configuring schedulers using the Junos command 'set schedulers' and attaching them to firewall policies, which in turn specify the target traffic in terms of IP addresses and port numbers as well as the action to be perform on signature triggering (allow or block/drop traffic).

Anomaly signatures based on throughput characteristics are implemented by configuring policers with a bandwidth limit and the desired signature action (discard or forward), using the Junos command 'set firewall policer', and attaching it to any interface with the Junos command 'set interfaces'. Traffic exceeding the specified throughput limit is dropped when the policer is configured to discard traffic.

A policer can be applied to specific inbound or outbound IP packets in a Layer 3 traffic flow at a logical interface by using a stateless firewall filter. If an input firewall filter is configured on the same logical interface as a policer, the policer is executed first. If an output firewall filter is configured on the same logical interface as a policer, the firewall filter is executed first.

Guidance	N/A

N/A

Testing	N/A

N/A

2.11.4.2 IPS_ABD_EXT.1.2

TSS	The evaluator shall verify that the TSS provides a description of how baselines are defined and implemented by the TOE, or a description of how anomaly-based rules are defined and configured by the administrator.

The TOE allows administrators to define signatures for anomalous traffic in terms of throughput (bits per second), time of the day for defined source/destination address and source/destination port, frequency of traffic patterns and thresholds of traffic patterns.

Anomaly signatures based on time of day characteristics are implemented by configuring schedulers using the Junos command 'set schedulers' and attaching them to firewall policies, which in turn specify the target traffic in terms of IP addresses and port numbers as well as the action to be perform on signature triggering (allow or block/drop traffic).

Anomaly signatures based on throughput characteristics are implemented by configuring policers with a bandwidth limit and the desired signature action (discard or forward), using the Junos command 'set firewall policer', and attaching it to any interface with the Junos command 'set interfaces'. Traffic exceeding the specified throughput limit is dropped when the policer is configured to discard traffic.

A policer can be applied to specific inbound or outbound IP packets in a Layer 3 traffic flow at a logical interface by using a stateless firewall filter. If an input firewall filter is configured on the same logical interface as a policer, the policer is executed first. If an output firewall filter is configured on the same logical interface as a policer, the firewall filter is executed first.

Guidance	N/A

N/A

Testing	N/A

N/A

2.11.4.3 IPS_ABD_EXT.1.3

TSS	The evaluator shall verify that each baseline or anomaly-based rule can be associated with a reaction specified in IPS_ABD_EXT.1.3.

The TOE allows administrators to define signatures for anomalous traffic in terms of throughput (bits per second), time of the day for defined source/destination address and source/destination port, frequency of traffic patterns and thresholds of traffic patterns.

Anomaly signatures based on time of day characteristics are implemented by configuring schedulers using the Junos command 'set schedulers' and attaching them to firewall policies, which in turn specify the target traffic in terms of IP addresses and port numbers as well as the action to be perform on signature triggering (allow or block/drop traffic).

Anomaly signatures based on throughput characteristics are implemented by configuring policers with a bandwidth limit and the desired signature action (discard or forward), using the Junos command 'set firewall policer', and attaching it to any interface with the Junos command 'set interfaces'. Traffic exceeding the specified throughput limit is dropped when the policer is configured to discard traffic.

A policer can be applied to specific inbound or outbound IP packets in a Layer 3 traffic flow at a logical interface by using a stateless firewall filter. If an input firewall filter is configured on the same logical interface as a policer, the policer is executed first. If an output firewall filter is configured on the same logical interface as a policer, the firewall filter is executed first.

TSS	The evaluator shall verify that the TSS identifies all interface types capable of applying baseline or anomaly-based rules and explains how they are associated with distinct network interfaces. Where interfaces can be grouped into a common interface type (e.g., where the same internal logical path is used, perhaps where a common device driver is used) they can be treated collectively as a distinct network interface.

The TOE is capable of inspecting all traffic passing through the TOE's Ethernet interfaces (inline mode). Ethernet interfaces can be assigned to Zones on which firewall and IDP policies are predicated.

A policer can be applied to specific inbound or outbound IP packets in a Layer 3 traffic flow at a logical interface by using a stateless firewall filter. If an input firewall filter is configured on the same logical interface as a policer, the policer is executed first. If an output firewall filter is configured on the same logical interface as a policer, the firewall filter is executed first.

Guidance	The evaluator shall verify that the operational guidance provides instructions to manually create baselines or anomaly-based rules according to the selections made in IPS_ABD_EXT.1.1. Note that dynamic "profiling" of a network to establish a baseline is outside the scope of this PP.

The provided guidance encompasses the entirety of the functionality provided by the IDP engine. Baselines/anomaly-based sensors can be created for throughput, time of day, frequency or thresholds. Depending on the anomaly type, this is done via the IDP engine (via custom attacks) or via the firewall engine (firewall policers/bandwidth monitors).

Guidance	The evaluator shall verify that the operational guidance provides instructions to associate reactions specified in IPS_ABD_EXT.1.3 with baselines or anomaly-based rules. The evaluator shall verify that the operational guidance provides instructions to associate the different policies with distinct network interfaces.

Per the IDP Feature Guide, rules are configured with reactions based on the following configuration settings:

```
then {
  action {
      < drop-connection | drop-packet | no-action>
  }
}
```

This allows the TOE administrator to configure IDP reactions in line with the reactions specified in the Security Target.

For bandwidth policers, the default reaction is to drop traffic (this does not require additional configuration).

Testing	The evaluator shall use the instructions in the operational guidance to configure baselines or anomaly-based rules for each attributes specified in IPS_ABD_EXT.1.1. The evaluator shall send traffic that does not match the baseline or matches the anomaly-based rule and verify the TOE applies the configured reaction. This shall be performed for each attribute in IPS_ABD_EXT.1.1.

The evaluators configured a number of policies that correspond to the selections made in this requirement. For each policy, evaluators generated traffic that did not match the expected baseline. In each case, the evaluators confirmed that the TOE a) detected the anomalous traffic; and b) reacted as expected.

Testing	Repeat the test assurance activity above to ensure that baselines or anomaly-based rules can be defined for each distinct network interface type supported by the TOE.

Evaluators confirmed during testing that IPS signatures could be applied to security zones that can then be associated with any applicable interface type.

---- END OF DOCUMENT ----