JUNIPER NETWORKS | Engineering Simplicity

vSRX Virtual Firewall User Guide for Private and Public Cloud Platforms

Published
2023-10-13

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

vSRX Virtual Firewall User Guide for Private and Public Cloud Platforms

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at https://support.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About This Guide

Use this guide to understand the security features that are supported on vSRX Virtual Firewall instances.

1

CHAPTER

Overview

vSRX Virtual Firewall Overview

SUMMARY

In this topic you learn about vSRX Virtual Firewall architecture and its benefits.

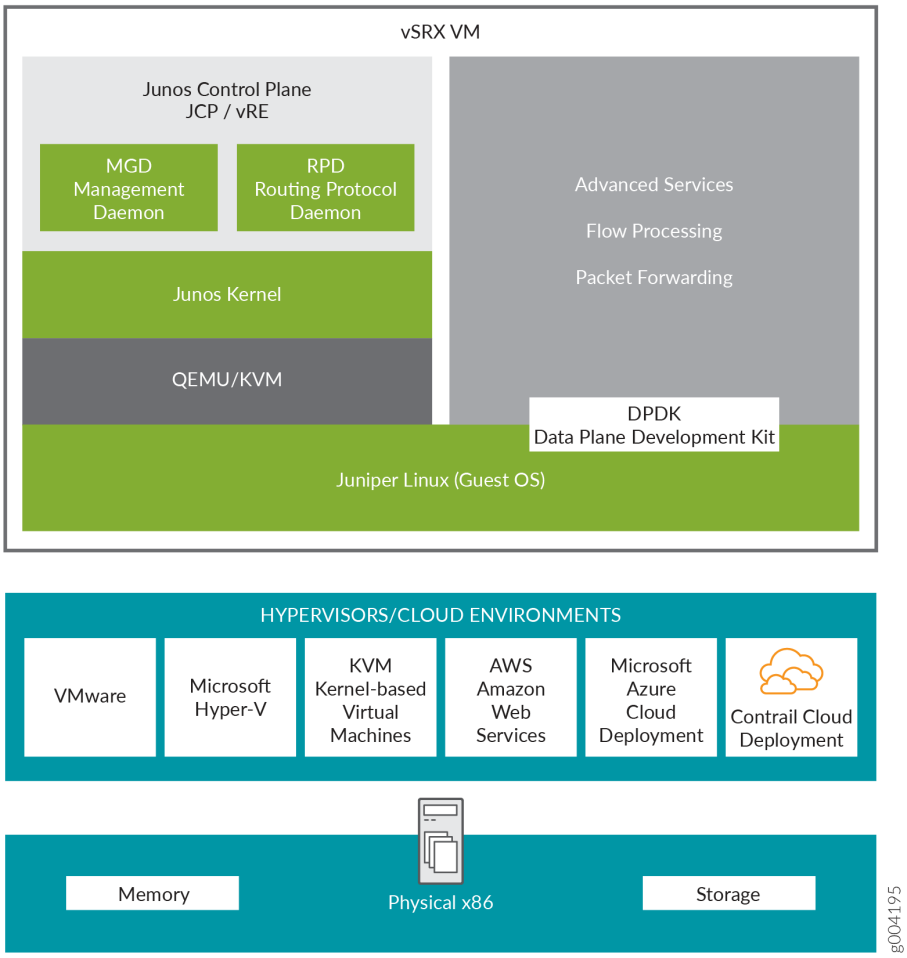
IN THIS SECTION

vSRX Virtual Firewall is a virtual security appliance that provides security and networking services at the perimeter or edge in virtualized private or public *cloud* environments. vSRX Virtual Firewall runs as a virtual machine (*VM*) on a standard x86 server. vSRX Virtual Firewall is built on the Junos operating system (Junos OS) and delivers networking and security features similar to those available on the software releases for the SRX Series Firewalls.

The vSRX Virtual Firewall provides you with a complete Next-Generation Firewall (NGFW) solution, including core firewall, VPN, NAT, advanced Layer 4 through Layer 7 security services such as Application Security, intrusion detection and prevention (IPS), and Content Security features including Enhanced Web Filtering and Anti-Virus. Combined with ATP Cloud, the vSRX Virtual Firewall offers a cloud-based advanced anti-malware service with dynamic analysis to protect against sophisticated malware, and provides built-in machine learning to improve verdict efficacy and decrease time to remediation.

Figure 1 on page 3 shows the high-level architecture.

Figure 1: vSRX Virtual Firewall Architecture



vSRX Virtual Firewall includes the Junos control plane (JCP) and the packet forwarding engine (PFE) components that make up the data plane. vSRX Virtual Firewall uses one virtual CPU (vCPU) for the JCP and at least one vCPU for the PFE. Starting in Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, multi-core vSRX Virtual Firewall supports scaling vCPUs and virtual RAM (vRAM). Additional vCPUs are applied to the data plane to increase performance.

Junos OS runs as a VM on vSRX Virtual Firewall. Junos OS does not have direct access to the NIC and only has a virtual NIC access provided by the hypervisor which might be shared with other VMs running on the same host machine. This virtual access comes with certain restrictions such as a special mode called trust mode, mode access might not be feasible because of possible security issues. To enable RETH model to work in such environments, MAC rewrite behavior is modified Instead of copying the parent virtual MAC address to the children, we keep the children's physical MAC address intact and copy the physical MAC address of the child belonging to the active; node of the cluster to the current MAC of the reth interface. This way, MAC rewrite access is not required when trust mode is disabled.

Setting the Trust mode for VFs (virtual functions), enables the host to change the MAC address of the guest during the run time. This helps vSRX Virtual Firewall interfaces to discover multiple IPv6 neighbours and perform better under scaling conditions. ND learning on vSRX Virtual Firewall interfaces is limited to only 10 IPv6 neighbours. For Linux setting for VF trust mode run the **`ip link set dev enp134s0f1 vf 0 trust on`** command on the host machine.

Verify the configuration:

user@host:~# ip link

```
enp134s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq portid 3cfdfed48ad9 state UP mode DEFAULT group default qlen 1000

link/ether 3c:fd:fe:d4:8a:d9 brd ff:ff:ff:ff:ff:ff

vf 0 MAC 00:00:00:00:00:00, spoof checking on, link-state auto, trust on.
```

Benefits

vSRX Virtual Firewall on standard x86 servers enables you to quickly introduce new services, deliver customized services to customers, and scale security services based on dynamic needs. vSRX Virtual Firewall is ideal for public, private, and hybrid cloud environments.

Some of the key benefits of vSRX Virtual Firewall in a virtualized private or public cloud multitenant environment include:

- *Stateful firewall* protection at the tenant edge
- Faster deployment of virtual firewalls into new sites
- Ability to run on top of various hypervisors and public cloud infrastructures
- Full routing, *VPN*, core security, and networking capabilities
- Application security features (including IPS and App-Secure)
- Content security features (including Anti Virus, Web Filtering, Anti Spam, and Content Filtering)
- Centralized management with Junos Space Security Director and local management with J-Web Interface
- Juniper Networks Juniper Advanced Threat Prevention Cloud (ATP Cloud) integration

Release History Table

Release	Description
15.1X49-D70	Starting in Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, multi-core vSRX Virtual Firewall supports scaling vCPUs and virtual RAM (vRAM). Additional vCPUs are applied to the data plane to increase performance.

2

CHAPTER

Managing vSRX Virtual Firewall

vSRX Virtual Firewall Configuration and Management Tools

SUMMARY

This topic provides an overview of the various tools available to configure and manage a vSRX Virtual Firewall VM once it has been successfully deployed.

IN THIS SECTION

Understanding the Junos OS CLI and Junos Scripts

Junos OS CLI is a Juniper Networks specific command shell that runs on top of a UNIX-based operating system kernel.

Built into Junos OS, Junos script automation is an onboard toolset available on all Junos OS platforms, including routers, switches, and security devices running Junos OS (such as a vSRX Virtual Firewall instance).

You can use the Junos OS CLI and the Junos OS scripts to configure, manage, administer, and troubleshoot vSRX Virtual Firewall.

Understanding the J-Web Interface

The *J-Web* interface allows you to monitor, configure, troubleshoot, and manage vSRX Virtual Firewall instances by means of a Web browser. J-Web provides access to all the configuration statements supported by the vSRX Virtual Firewall instance.

Understanding Junos Space Security Director

As one of the Junos Space Network Management Platform applications, Junos Space Security Director helps organizations improve the reach, ease, and accuracy of security policy administration with a scalable, GUI-based management tool. Security Director automates security provisioning of a vSRX Virtual Firewall instance through one centralized Web-based interface to help administrators manage all phases of the security policy life cycle more quickly and intuitively, from policy creation to remediation.

RELATED DOCUMENTATION

- CLI User Interface Overview
- J-Web Overview
- Security Director
- Mastering Junos Automation Programming
- Spotlight Secure Threat Intelligence

Managing Security Policies for Virtual Machines Using Junos Space Security Director

SUMMARY

This topic provides you an overview of how you can manage security policies for VMs using security director.

Security Director is a Junos Space management application designed to enable quick, consistent, and accurate creation, maintenance, and application of network security policies for your security devices, including vSRX Virtual Firewall instances. With Security Director, you can configure security-related policy management including IPsec VPNs, firewall policies, NAT policies, IPS policies, and Content Security policies. and push the configurations to your security devices. These configurations use objects such as addresses, services, NAT pools, application signatures, policy profiles, VPN profiles, template definitions, and templates. These objects can be shared across multiple security configurations; shared objects can be created and used across many security policies and devices. You can create these objects prior to creating security configurations.

When you finish creating and verifying your security configurations from Security Director, you can publish these configurations and keep them ready to be pushed to all security devices, including vSRX Virtual Firewall instances, from a single interface.

The Configure tab is the workspace where all of the security configuration happens. You can configure firewall, IPS, NAT, and Content Security policies; assign policies to devices; create and apply policy schedules; create and manage VPNs; and create and manage all the shared objects needed for managing your network security.

RELATED DOCUMENTATION

- Security Director

Configure a vSRX Virtual Firewall Chassis Cluster in Junos OS

IN THIS SECTION

Chassis Cluster Overview

Chassis cluster groups a pair of the same kind of vSRX Virtual Firewall instances into a cluster to provide network node redundancy. The vSRX Virtual Firewall instances in a chassis cluster must be running the same Junos OS release, and each instance becomes a node in the chassis cluster. You connect the control virtual interfaces on the respective nodes to form a *control plane* that synchronizes the configuration and Junos OS kernel state on both nodes in the cluster. The control link (a *virtual network* or *vSwitch*) facilitates the redundancy of interfaces and services. Similarly, you connect the *data plane* on the respective nodes over the fabric virtual interfaces to form a unified data plane. The fabric link (a

virtual network or vSwitch) allows for the management of cross-node flow processing and for the management of session redundancy.

The control plane software operates in active/passive mode. When configured as a chassis cluster, one node acts as the primary and the other as the secondary to ensure stateful failover of processes and services in the event of a system or hardware failure on the primary . If the primary fails, the secondary takes over processing of control plane traffic.

NOTE: If you configure a chassis cluster across two hosts, disable igmp-snooping on the bridge that each host physical interface belongs to and that the control virtual NICs (vNICs) use. This ensures that the control link heartbeat is received by both nodes in the chassis cluster.

The chassis cluster data plane operates in active/active mode. In a chassis cluster, the data plane updates session information as traffic traverses either node, and it transmits information between the nodes over the fabric link to guarantee that established sessions are not dropped when a failover occurs. In active/active mode, traffic can enter the cluster on one node and exit from the other node.

Chassis cluster functionality includes:

- Resilient system architecture, with a single active control plane for the entire cluster and multiple *Packet Forwarding Engine*s. This architecture presents a single device view of the cluster.
- Synchronization of configuration and dynamic runtime states between nodes within a cluster.
- Monitoring of physical interfaces, and failover if the failure parameters cross a configured threshold.
- Support for generic routing encapsulation (*GRE*) and IP-over-IP (IP-IP) tunnels used to route encapsulated IPv4 or *IPv6* traffic by means of two internal interfaces, gr-0/0/0 and ip-0/0/0, respectively. Junos OS creates these interfaces at system startup and uses these interfaces only for processing GRE and IP-IP tunnels.

At any given instant, a cluster node can be in one of the following states: hold, primary, secondary-hold, secondary, ineligible, or disabled. Multiple event types, such as interface monitoring, Services Processing Unit (SPU) monitoring, failures, and manual failovers, can trigger a state transition.

Enable Chassis Cluster Formation

You create two vSRX Virtual Firewall instances to form a chassis cluster, and then you set the cluster ID and node ID on each instance to join the cluster. When a vSRX Virtual Firewall instance joins a cluster, it becomes a node of that cluster. With the exception of unique node settings and management IP addresses, nodes in a cluster share the same configuration.

You can deploy up to 255 chassis clusters in a *Layer 2* domain. Clusters and nodes are identified in the following ways:

- The *cluster ID* (a number from 1 to 255) identifies the cluster.
- The *node ID* (a number from 0 to 1) identifies the cluster node.

Generally, on SRX Series Firewalls, the cluster ID and node ID are written into EEPROM. On the vSRX Virtual Firewall instance, vSRX Virtual Firewall stores and reads the IDs from **boot/loader.conf** and uses the IDs to initialize the chassis cluster during startup.

Prerequisites

Ensure that your vSRX Virtual Firewall instances comply with the following prerequisites before you enable chassis clustering:

- You have committed a basic configuration to both vSRX Virtual Firewall instances that form the chassis cluster. See *Configure vSRX Using the CLI*.
- Use `show version` in Junos OS to ensure that both vSRX Virtual Firewall instances have the same software version.
- Use `show system license` in Junos OS to ensure that both vSRX Virtual Firewall instances have the same licenses installed.

You must set the same chassis cluster ID on each vSRX Virtual Firewall node and reboot the vSRX Virtual Firewall VM to enable chassis cluster formation.

1. In operational command mode, set the chassis cluster ID and node number on vSRX Virtual Firewall node 0.

```
user@vsrx0>set chassis cluster cluster-id number node 0 reboot
```

2. In operational command mode, set the chassis cluster ID and node number on vSRX Virtual Firewall node 1.

```
user@vsrx1>set chassis cluster cluster-id number node 1 reboot
```

NOTE: The vSRX Virtual Firewall interface naming and mapping to vNICs changes when you enable chassis clustering. See *Requirements for vSRX on KVM* for a summary of interface names and mappings for a pair of vSRX Virtual Firewall VMs in a cluster (node 0 and node 1).

Chassis Cluster Quick Setup with J-Web

To configure chassis cluster from *J-Web*:

1. Enter the vSRX Virtual Firewall node 0 interface IP address in a Web browser.
2. Enter the vSRX Virtual Firewall username and password, and click **Log In**. The J-Web dashboard appears.
3. Click **Configuration Wizards> Cluster (HA) Setup** from the left panel. The Chassis Cluster Setup Wizard appears. Follow the steps in the setup wizard to configure the cluster ID and the two nodes in the cluster, and to verify connectivity.

   NOTE: Use the built-in Help icon in J-Web for further details on the Chassis Cluster Setup wizard.

   NOTE: Navigate to **Configure>Device Settings>Cluster (HA) Setup** from Junos OS release 18.1 and later to configure the chassis cluster setup.
4. Configure the secondary node Node1 by selecting **Yes, this is the secondary unit to be setup (Node 1)** using radio button.
5. Click **Next**.
6. Specify the settings such as **Enter password**, **Re-enter password**, **Node 0 FXP0 IP**, and **Node 1 FXP0 IP** for secondary node access.
7. Click **Next**.
8. Select the secondary unit's Control Port and Fabric Port.
9. Click **Next**.
10. (Optional) Select **Save a backup file before proceeding with shutdown** using check box to re-configure it for chassis cluster.
11. Click **Next**.
12. Click **Shutdown and continue** to connect to other unit.
13. Click **Refresh Browser**.
14. Configure the primary node Node0 by selecting **No, this is the primary unit to be setup (Node 0)** to configure primary unit and establish a chassis cluster configuration.
15. Click **Next**.
16. Specify the settings such as **Enter password**, **Re-enter password**, **Node 0 FXP0 IP**, and **Node 1 FXP0 IP** for primary node access.
17. Click **Next** to restart the primary unit.

18. (Optional) Select **Save a backup file before proceeding with shutdown** to save a backup file of current settings before proceeding.
19. Click **Reboot and continue**. After completing the reboot, power on the secondary unit to establish the chassis cluster connection.
20. Login to the device console and add static route to get the J-Web access.
21. Login to the J-Web and click **Configuration Wizards> Cluster (HA) Setup** from the left panel. The Chassis Cluster Setup Wizard appears.
22. Click **Next** to get the primary unit connected.
23. Configure the basic settings **DHCP Client**, **IP address**, **Default gateway**, **Member interface Node 0**, **Member interface Node 1**.
24. Click **Next** to complete the chassis cluster configuration.
25. Click **Finish** to exit the wizard. You can access the primary node using J-Web.

Manually Configure a Chassis Cluster with J-Web

You can use the *J-Web* interface to configure the primary node 0 vSRX Virtual Firewall instance in the cluster. Once you have set the cluster and node IDs and rebooted each vSRX Virtual Firewall, the following configuration will automatically be synced to the secondary node 1 vSRX Virtual Firewall instance.

Select **Configure>Chassis Cluster>Cluster Configuration**. The Chassis Cluster configuration page appears.

> **NOTE**: Navigate to **Configure>Device Settings>Cluster (HA) Setup** from Junos OS release 18.1 and later to configure the HA cluster setup.

Table 1 on page 14 explains the contents of the HA Cluster Settings tab.

Table 2 on page 15 explains how to edit the Node Settings tab.

Table 3 on page 16 explains how to add or edit the HA Cluster Interfaces table.

Table 4 on page 17 explains how to add or edit the HA Cluster Redundancy Groups table.

Table 1: Chassis Cluster Configuration Page

Field	Function
Node Settings	
Node ID	Displays the node ID.
Cluster ID	Displays the cluster ID configured for the node.
Host Name	Displays the name of the node.
Backup Router	Displays the router used as a gateway while the Routing Engine is in secondary state for redundancy-group 0 in a chassis cluster.
Management Interface	Displays the management interface of the node.
IP Address	Displays the management IP address of the node.
Status	Displays the state of the redundancy group. • **Primary**–Redundancy group is active. • **Secondary**–Redundancy group is passive.
Chassis Cluster>HA Cluster Settings>Interfaces	
Name	Displays the physical interface name.
Member Interfaces/IP Address	Displays the member interface name or IP address configured for an interface.
Redundancy Group	Displays the redundancy group.
Chassis Cluster>HA Cluster Settings>Redundancy Group	

Table 1: Chassis Cluster Configuration Page *(Continued)*

Field	Function
Group	Displays the redundancy group identification number.
Preempt	Displays the selected preempt option. • **True**–Primary Role can be preempted based on priority. • **False**–Primary Role cannot be preempted based on priority.
Gratuitous ARP Count	Displays the number of gratuitous Address Resolution Protocol (*ARP*) requests that a newly elected primary device in a chassis cluster sends out to announce its presence to the other network devices.
Node Priority	Displays the assigned priority for the redundancy group on that node. The eligible node with the highest priority is elected as primary for the redundant group.

Table 2: Edit Node Setting Configuration Details

Field	Function	Action
Node Settings		
Host Name	Specifies the name of the host.	Enter the name of the host.
Backup Router	Displays the device used as a gateway while the Routing Engine is in the secondary state for redundancy-group 0 in a chassis cluster.	Enter the IP address of the backup router.
Destination		
IP	Adds the destination address.	Click **Add**.
Delete	Deletes the destination address.	Click **Delete**.

Table 2: Edit Node Setting Configuration Details *(Continued)*

Field	Function	Action
Interface		
Interface	Specifies the interfaces available for the router. **NOTE**: Allows you to add and edit two interfaces for each fabric link.	Select an option.
IP	Specifies the interface IP address.	Enter the interface IP address.
Add	Adds the interface.	Click **Add**.
Delete	Deletes the interface.	Click **Delete**.

Table 3: Add HA Cluster Interface Configuration Details

Field	Function	Action
Fabric Link > Fabric Link 0 (fab0)		
Interface	Specifies fabric link 0.	Enter the interface IP fabric link 0.
Add	Adds fabric interface 0.	Click **Add**.
Delete	Deletes fabric interface 0.	Click **Delete**.
Fabric Link > Fabric Link 1 (fab1)		
Interface	Specifies fabric link 1.	Enter the interface IP for fabric link 1.
Add	Adds fabric interface 1.	Click **Add**.
Delete	Deletes fabric interface 1.	Click **Delete**.

Table 3: Add HA Cluster Interface Configuration Details *(Continued)*

Field	Function	Action
Redundant Ethernet		
Interface	Specifies a logical interface consisting of two physical Ethernet interfaces, one on each chassis.	Enter the logical interface.
IP	Specifies a redundant Ethernet IP address.	Enter a redundant Ethernet IP address.
Redundancy Group	Specifies the redundancy group ID number in the chassis cluster.	Select a redundancy group from the list.
Add	Adds a redundant Ethernet IP address.	Click **Add**.
Delete	Deletes a redundant Ethernet IP address.	Click **Delete**.

Table 4: Add Redundancy Groups Configuration Details

Field	Function	Action
Redundancy Group	Specifies the redundancy group name.	Enter the redundancy group name.
Allow preemption of primaryship	Allows a node with a better priority to initiate a failover for a redundancy group. **NOTE**: By default, this feature is disabled. When disabled, a node with a better priority does not initiate a redundancy group failover (unless some other factor, such as faulty network connectivity identified for monitored interfaces, causes a failover).	–

Table 4: Add Redundancy Groups Configuration Details *(Continued)*

Field	Function	Action
Gratuitous ARP Count	Specifies the number of gratuitous Address Resolution Protocol requests that a newly elected primary sends out on the active redundant Ethernet interface child links to notify network devices of a change in primary role on the redundant Ethernet interface links.	Enter a value from 1 to 16. The default is 4.
node0 priority	Specifies the priority value of node0 for a redundancy group.	Enter the node priority number as 0.
node1 priority	Specifies the priority value of node1 for a redundancy group.	Select the node priority number as 1.
Interface Monitor		
Interface	Specifies the number of redundant Ethernet interfaces to be created for the cluster.	Select an interface from the list.
Weight	Specifies the weight for the interface to be monitored.	Enter a value from 1 to 125.
Add	Adds interfaces to be monitored by the redundancy group along with their respective weights.	Click **Add**.
Delete	Deletes interfaces to be monitored by the redundancy group along with their respective weights.	Select the interface from the configured list and click **Delete**.
IP Monitoring		
Weight	Specifies the global weight for IP monitoring.	Enter a value from 0 to 255.
Threshold	Specifies the global threshold for IP monitoring.	Enter a value from 0 to 255.

Table 4: Add Redundancy Groups Configuration Details *(Continued)*

Field	Function	Action
Retry Count	Specifies the number of retries needed to declare reachability failure.	Enter a value from 5 to 15.
Retry Interval	Specifies the time interval in seconds between retries.	Enter a value from 1 to 30.
IPV4 Addresses to Be Monitored		
IP	Specifies the IPv4 addresses to be monitored for reachability.	Enter the IPv4 addresses.
Weight	Specifies the weight for the redundancy group interface to be monitored.	Enter the weight.
Interface	Specifies the logical interface through which to monitor this IP address.	Enter the logical interface address.
Secondary IP address	Specifies the source address for monitoring packets on a secondary link.	Enter the secondary IP address.
Add	Adds the IPv4 address to be monitored.	Click **Add**.
Delete	Deletes the IPv4 address to be monitored.	Select the IPv4 address from the list and click **Delete**.

SEE ALSO

- Chassis Cluster Feature Guide for Security Devices

CHAPTER 3

Supported vSRX Virtual Firewall Features

Junos OS Features Supported on vSRX Virtual Firewall

SUMMARY

This topic provides details of the Junos OS features supported and not supported on vSRX Virtual Firewall.

IN THIS SECTION

SRX Series Features Supported on vSRX Virtual Firewall

vSRX Virtual Firewall inherits most of the branch SRX Series features with the following considerations shown in Table 5 on page 21.

To determine the Junos OS features supported on vSRX Virtual Firewall, use the Juniper Networks Feature Explorer, a Web-based application that helps you to explore and compare Junos OS feature information to find the right software release and hardware platform for your network. Find Feature Explorer at: Feature Explorer: vSRX .

Table 5: vSRX Virtual Firewall Feature Considerations

Feature	Description
IDP	The IDP feature is subscription based and must be purchased. After purchase, you can activate the IDP feature with the license key. For SRX Series IDP configuration details, see: Understanding Intrusion Detection and Prevention for SRX Series

Table 5: vSRX Virtual Firewall Feature Considerations *(Continued)*

Feature	Description	
IPSec VPNs	Starting in Junos OS Release 19.3R1, vSRX Virtual Firewall supports the following authentication algorithms and encryption algorithms: • Authentication algorithm: hmac-sha1-96 and HMAC-SHA-256-128 authentication • Encryption algorithm: aes-128-cbc Starting in Junos OS Release 20.3R1, vSRX Virtual Firewall supports 10,000 IPsec VPN tunnels. To support the increased number of IPsec VPN tunnels, a minimum of 19 vCPUs are required. Out of the 19 vCPUs, 3 vCPUs must be dedicated to RE. You must run the `request system software add optional://junos-ike.tgz` command the first time you wish to enable increased IPsec tunnel capacity. For subsequent software upgrades of the instance, the junos-ike package is upgraded automatically from the new Junos OS releases installed in the instance. DH group15, group16, group21 is also added when we install junos-ike package. If chassis cluster is enabled then run this command on both the nodes. You can configure the number of vCPUs allocated to Junos Routing Engine using the `set security forwarding-options resource-manager cpu re <value>`. **NOTE**: 64 G memory is required to support 10000 tunnels in PMI mode. [See show security ipsec security-associations, show security ike tunnel-map, and show security ipsec tunnel-distribution.]	
IPsec VPN - Tunnel Scaling on vSRX Virtual Firewall	Types of Tunnels	Number of tunnels supported
	Site-Site VPN tunnels	2000
	AutoVPN tunnels	10,000
	IKE SA (Site-to-site)	2000
	IKE SA (AutoVPN)	10,000
	IKE SA (Site-to-site + AutoVPN)	10,000

Table 5: vSRX Virtual Firewall Feature Considerations *(Continued)*

Feature	Description	
	IPSec SA pairs (Site-to-site)	10,000 With 2000 IKE SAs, we can have 10,000 IPSec SA.
	IPSec SA pairs (AutoVPN)	10,000
	Site-to-site + AutoVPN IPSec SA pairs	2000 Site-to-site 8000 AutoVPN
	Site-to-site + AutoVPN tunnels	2000 Site-to-site 8000 AutoVPN
ISSU	ISSU is not supported.	
Logical Systems	Starting in Junos OS Release 20.1R1, you can configure logical systems and tenant systems on vSRX Virtual Firewall and vSRX Virtual Firewall 3.0 instances. With Junos OS, you can partition a single security device into multiple logical devices that can perform independent tasks. Each logical system has its own discrete administrative domain, logical interfaces, routing instances, security firewall and other security features. See Logical Systems Overview.	

Table 5: vSRX Virtual Firewall Feature Considerations *(Continued)*

Feature	Description
PowerMode IPsec	Starting in Junos OS Release 20.1R1, vSRX Virtual Firewall 3.0 instances support PowerMode IPsec that provides IPsec performance improvements using Vector Packet Processing (VPP) and Intel AES-NI instructions. PowerMode IPsec is a small software block inside the SRX PFE (SRX Packet Forwarding Engine) that is activated when PowerMode is enabled. Supported Features in PowerMode IPsec • IPsec functionality • Traffic selectors • Secure tunnel interface (st0) • All control plane IKE functionality • Auto VPN with traffic selector • Auto VPN with routing protocol • IPv6 • Stateful Layer 4 firewall • High-Availability • NAT-T Non-Supported Features in PowerMode IPsec • NAT • IPsec in IPsec • GTP/SCTP firewall • Application firewall/AppSecure • QoS • Nested tunnel • Screen

Table 5: vSRX Virtual Firewall Feature Considerations *(Continued)*

Feature	Description
	• Multicast • Host traffic
Ethernet Switching and Bridging	Starting in Junos OS Release 22.1R1, vSRX Virtual Firewall and vSRX Virtual Firewall 3.0 instances deployed on KVM and VMware platforms support flexible VLAN tagging on revenue and reth interfaces. Flexible VLAN tagging supports transmission of 802.1Q VLAN single-tag frames on logical interfaces on the Ethernet port. Also, avoids multiple virtual functions on the network interface card (NIC) and reduces the need of additional interfaces. [See Configuring VLAN Tagging and flexible-vlan-tagging (Interfaces).]
Tenant Systems	Starting in Junos OS Release 20.1R1, you can configure tenant systems on vSRX Virtual Firewall and vSRX Virtual Firewall 3.0 instances. A tenant system provides logical partitioning of the SRX Series Firewall into multiple domains similar to logical systems and provides high scalability. See Tenant Systems Overview.
Transparent mode	The known behaviors for transparent mode support on vSRX Virtual Firewall are: • The default MAC learning table size is restricted to 16,383 entries. For information about configuring transparent mode for vSRX Virtual Firewall, see Layer 2 Bridging and Transparent Mode Overview.

Table 5: vSRX Virtual Firewall Feature Considerations *(Continued)*

Feature	Description
Content Security	• The Content Security feature is subscription based and must be purchased. After purchase, you can activate the Content Security feature with the license key. • Starting in Junos OS Release 19.4R1, vSRX Virtual Firewall 3.0 instances support the Avira scan engine, which is an on-device antivirus scanning engine. See On-Device Antivirus Scan Engine. • For SRX Series Content Security configuration details, see Unified Threat Management Overview. • For SRX Series Content Security antispam configuration details, see Antispam Filtering Overview. • **Advanced resource management (vSRX 3.0)**—Starting in Junos OS Release 19.4R1, vSRX Virtual Firewall 3.0 manages the additional system resource requirements for Content Security-and IDP-specific services by reallocating CPU cores and extra memory. These values for memory and CPU cores are not user configured. Previously, system resources such as memory and CPU cores were fixed. You can view the allocated CPU and memory for advance security services on vSRX Virtual Firewall 3.0 instance by using the `show security forward-options resource-manager settings` command. To view the flow session scaling, use the `show security monitoring` command. [See show security monitoring and show security forward-options resource-manager settings.]
Tunnels	Only GRE and IP-IP

Some Junos OS software features require a license to activate the feature. To understand more about vSRX Virtual Firewall Licenses, see, Licenses for vSRX. Please refer to the Licensing Guide for general information about License Management. Please refer to the product Data Sheets for further details, or contact your Juniper Account Team or Juniper Partner.

SRX Series Features Not Supported on vSRX Virtual Firewall

vSRX Virtual Firewall inherits many features from the SRX Series Firewall product line. Table 6 on page 27 lists SRX Series features that are not applicable in a virtualized environment, that are not currently supported, or that have qualified support on vSRX Virtual Firewall.

Table 6: SRX Series Features Not Supported on vSRX Virtual Firewall

SRX Series Feature	vSRX Virtual Firewall Notes
Application Layer Gateways	
Avaya H.323	Not supported
Authentication with IC Series devices	
Layer 2 enforcement in UAC deployments	Not supported **NOTE**: UAC-IDP and UAC-Content Security also are not supported.
Chassis cluster support **NOTE**: Support for chassis clustering to provide network node redundancy is only available on a vSRX Virtual Firewall deployment in Contrail, VMware, KVM, and Windows Hyper-V Server 2016.	
Chassis cluster for VirtIO driver	Only supported with KVM **NOTE**: The link status of VirtIO interfaces is always reported as UP, so a vSRX Virtual Firewall chassis cluster cannot receive link up and link down messages from VirtIO interfaces.
Dual control links	Not supported
In-band and low-impact cluster upgrades	Not supported
LAG and LACP (Layer 2 and Layer 3)	Not supported
Layer 2 Ethernet switching	Not supported

Table 6: SRX Series Features Not Supported on vSRX Virtual Firewall *(Continued)*

SRX Series Feature	vSRX Virtual Firewall Notes
Low-latency firewall	Not supported
Class of service	
High-priority queue on SPC	Not supported
Tunnels	A vSRX Virtual Firewall VM deployed on Microsoft Azure Cloud does not support GRE, IP-IP and multicast.
Data plane security log messages (stream mode)	
TLS protocol	Not supported
Diagnostic tools	
Flow monitoring cflowd version 9	Not supported
Ping Ethernet (CFM)	Not supported
Traceroute Ethernet (CFM)	Not supported
DNS proxy	
Dynamic DNS	Not supported
Ethernet link aggregation	
LACP in standalone or chassis cluster mode	Not supported
Layer 3 LAG on routed ports	Not supported
Static LAG in standalone or chassis cluster mode	Not supported

Table 6: SRX Series Features Not Supported on vSRX Virtual Firewall *(Continued)*

SRX Series Feature	vSRX Virtual Firewall Notes
Ethernet link fault management	
Physical interface (encapsulations) • `ethernet-ccc` • `ethernet-tcc` • `extended-vlan-ccc` • `extended-vlan-tcc`	Not supported
Interface family • `ccc, tcc` • `ethernet-switching`	Not supported
Flow-based and packet-based processing	
End-to-end packet debugging	Not supported
Network processor bundling	
Services offloading	
Interfaces	
Aggregated Ethernet interface	Not supported
IEEE 802.1X dynamic VLAN assignment	Not supported
IEEE 802.1X MAC bypass	Not supported

Table 6: SRX Series Features Not Supported on vSRX Virtual Firewall *(Continued)*

SRX Series Feature	vSRX Virtual Firewall Notes
IEEE 802.1X port-based authentication control with multisupplicant support	Not supported
Interleaving using MLFR	Not supported
PoE	Not supported
PPP interface	Not supported
PPPoE-based radio-to-router protocol	Not supported
PPPoE interface **NOTE**: Starting in Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1, the vSRX Virtual Firewall supports Point-to-Point Protocol over Ethernet (PPPoE) interface.	Not supported
Promiscuous mode on interfaces	Only supported if enabled on the hypervisor
IPSec and VPNs	
Acadia - Clientless VPN	Not supported
DVPN	Not supported
Hardware IPsec (bulk crypto) Cavium/RMI	Not supported
IPsec tunnel termination in routing instances	Supported on virtual router only
Multicast for AutoVPN	Not supported
IPv6 support	

Table 6: SRX Series Features Not Supported on vSRX Virtual Firewall *(Continued)*

SRX Series Feature	vSRX Virtual Firewall Notes
DS-Lite concentrator (also called Address Family Transition Router [AFTR])	Not supported
DS-Lite initiator (aka B4)	Not supported
J-Web	
Enhanced routing configuration	Not supported
New Setup wizard (for new configurations)	Not supported
PPPoE wizard	Not supported
Remote VPN wizard	Not supported
Rescue link on dashboard	Not supported
Content Security configuration for Kaspersky antivirus and the default Web filtering profile	Not supported
Log file formats for system (control plane) logs	
Binary format (binary)	Not supported
WELF	Not supported
Miscellaneous	
GPRS **NOTE**: Starting in Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, vSRX Virtual Firewall supports GPRS.	Not supported

Table 6: SRX Series Features Not Supported on vSRX Virtual Firewall *(Continued)*

SRX Series Feature	vSRX Virtual Firewall Notes
Hardware acceleration	Not supported
Outbound SSH	Not supported
Remote instance access	Not supported
USB modem	Not supported
Wireless LAN	Not supported
MPLS	
Crcuit cross-connect (CCC) and translational cross-connect (TCC)	Not supported
Layer 2 VPNs for Ethernet connections	Only if promiscuous mode is enabled on the hypervisor
Network Address Translation	
Maximize persistent NAT bindings	Not supported
Packet capture	
Packet capture	Only supported on physical interfaces and tunnel interfaces, such as *gr*, *ip*, and *st0*. Packet capture is not supported on redundant Ethernet interfaces (*reth*).
Routing	
BGP extensions for IPv6	Not supported
BGP Flowspec	Not supported

Table 6: SRX Series Features Not Supported on vSRX Virtual Firewall *(Continued)*

SRX Series Feature	vSRX Virtual Firewall Notes
BGP route reflector	Not supported
CRTP	Not supported
Switching	
Layer 3 Q-in-Q VLAN tagging	Not supported
Transparent mode	
Content Security	Not supported
Content Security	
Express AV	Not supported
Kaspersky AV	Not supported
Upgrading and rebooting	
Autorecovery	Not supported
Boot instance configuration	Not supported
Boot instance recovery	Not supported
Dual-root partitioning	Not supported
OS rollback	Not supported
User interfaces	

Table 6: SRX Series Features Not Supported on vSRX Virtual Firewall *(Continued)*

SRX Series Feature	vSRX Virtual Firewall Notes
NSM	Not supported
SRC application	Not supported
Junos Space Virtual Director	Only supported with VMware

Software Receive Side Scaling

IN THIS SECTION

Overview

Contemporary NICs support multiple receive and transmit descriptor queues (multi-queue). On reception, a NIC can send different packets to different queues to distribute processing among CPUs. The NIC distributes packets by applying a filter to each packet that assigns it to one of a small number of logical flows. Packets for each flow are steered to a separate receive queue, which in turn can be processed by separate CPUs. This mechanism is generally known as Receive-side Scaling (RSS). The goal of RSS technique is to increase performance uniformly. RSS is enabled when latency is a concern or whenever receive interrupt processing forms a bottleneck. Spreading load between CPUs decreases queue length. For low latency networking, the optimal setting is to allocate as many queues as there are CPUs in the system (or the NIC maximum, if lower). The most efficient high-rate configuration is likely the one with the smallest number of receive queues where no receive queue overflows due to a saturated CPU. You can improve bridging throughput with Receive Side Scaling.

As per flow thread affinity architecture each flow thread (FLT) polls for packet from dedicated receiving queue of NIC and process the packets until run to completion. Therefore, flow threads are bound to NIC

receiving (RX) and transmitting (TX) queues for packet processing to avoid any disagreement. Hence, NIC must have same number of RX and TX queues as number of vSRX Virtual Firewall data plane CPU to support multi core vSRX Virtual Firewall flavors. Software RSS (SWRSS) removes this limitation of NIC HW queues to run vSRX Virtual Firewall multi-core flavors by implementing software-based packet spraying across various FLT thread.

Software RSS offloads the handling of individual flows to one of the multiple kernel, so the flow thread that takes the packets from the NIC can process more packets. Similar to RSS, network throughput improvement when using SWRSS has a linear correlation with CPU utilization.

In SWRSS, each NIC port is initialized with equal or lesser number of hardware RX/TX queues as that of I/O threads. I/O threads are determined based on total data-path CPU and minimum of NIC queues among all the NIC interface in vSRX Virtual Firewall. For example, if I/O thread is computed as 4, then number of HW queue per NIC port can be less or equal to 4 queues.

If NICs do not have sufficient number of queues as FLT threads in vSRX Virtual Firewall instances supported, then Software RSS (SWRSS) is enabled by flowd data-path. SWRSS implements software model of packet distribution across FLTs after obtaining the packets from NIC receiving queues. By removing NIC HW queue limitation, SWRSS helps to scale vCPUs by supporting various vSRX Virtual Firewall instance types.

During the I/O operation the packets are fetched from receiving queues of NIC ports and packet classification is performed. Followed by distribution of packets to FLT threads virtual queues. These virtual queues are implemented over DPDK ring queue. In the transmission path, SWRSS fetches the packets from virtual transmitting queues of FLT threads and pushes these packets to NIC transmitting queues for transmit.

Number of SWRSS I/O threads are selected based on total CPU and number of NIC queues found in vSRX Virtual Firewall instances. Mix mode of operation with HWRSS and and SWRSS is not supported.

Understanding Software Receive Side Scaling Configuration

This topic provide you details on types of Software Receive Side Scaling (SWRSS) and its configuration.

SWRSS supports two modes of operation and it gets enabled based on number of data-path CPU needed. These modes are Shared IO mode and dedicated IO mode. These modes are enabled based on number of data-path CPUs needed. vSRX Virtual Firewall and vSRX3.0 supports dedicated I/O mode only.

In dedicated I/O mode flowd process creates dedicated I/O threads for I/O operation. Based on number of required I/O threads for vSRX Virtual Firewall, I/O thread is associated to a dedicated NIC port. NIC ports receiving and transmitting queue is then bonded to each I/O thread in round robin method for uniform distribution and to avoid I/O thread locks. Each dedicated I/O thread pulls the packets in burst

mode from NIC receiving queue and distributes to FLT threads and vice versa for TX path for packet transmit.

SWRSS is enabled based on the number of vCPUs. If NIC does not have sufficient number of queues as flow thread (FLT) in vSRX Virtual Firewall with different vCPUs, then Software RSS (SWRSS) is enabled by flowd process.

SWRSS is not enabled in the following scenarios:

- When the NIC has sufficient number of hardware RX or TX queues for required PFE data-path CPU.
- When the vSRX Virtual Firewall (based on number of vCPUs) and NIC result the smaller number of FLT CPUs as that obtained in nearest hardware RSS (HWRSS) mode. In such scenario, vSRX Virtual Firewall will be enabled with HWRSS mode which results more FLT CPU than SWRSS mode, providing better packet processing throughput.
- SWRSS is not recommended for vSRX Virtual Firewall with certain type of NIC that supports lesser number of NIC queues than needed to run dedicated IO thread. In such cases, SWRSS is enabled but extra CPUs are attached to FLT CPU, until I/O CPUs are completely utilized.

If SWRSS is not enabled use the `set security forwarding-options receive-side-scaling software-rss mode enable` command to enable SWRSS. When you run this command SWRSS will be enabled by force regardless of the NIC RSS or the number of vCPUs. If you do not enable SWRSS using the CLI then enabling of SWRSS automatically is decided based on the default ratio of FLT: IO (4:1).

To configure the number of required IO threads, use the `set security forwarding-options receive-side-scaling software-rss io-thread-number <1-8>` command. To view the actual number of vCPUs assigned to IO flow threads use the `show security forwarding-options resource-manager` command.

You can decide enabling of SWRSS automatically or by force based on the architecture and conception of IO thread and worker thread. Enabling SWRSS impacts the performance, so we recommend that the number of IO thread should be changed only if required and until the performance impact bottleneck point is reached.

GTP Traffic with TEID Distribution and SWRSS

IN THIS SECTION

Overview GTP Traffic Distribution with TEID Distribution and SWRSS

IN THIS SECTION

The topic provides an overview of asymmetric fat tunnel solution for GTP traffic with TEID distribution and SWRSS.

With TEID-based hash distributions feature, the GTP packets would be distributed to the flow thread according to the hash value calculated by TEID. The algorithm of hash calculation is same as GTP distribution in flow module, which ensures the GTP packets would not be reinjected again in the flow process.

There is a 4-byte field inside GTP payload called tunnel endpoint identifier (TEID), which is used to identify different connections in the same GTP tunnel.

A fat GTP tunnel carries data from different users. IPsec tunnels on the security gateway could be a fat tunnel due to the fat GTP tunnel. vSRX Virtual Firewall can create one GTP session with a high-bandwidth of GTP traffic. However, the throughput is limited to one core processor's performance.

If you use TEID-based hash distribution for creating GTP-U sessions, then you can:

- Enable vSRX Virtual Firewall and vSRX Virtual Firewall 3.0 instances to process asymmetric fat tunnels for parallel encryption on multiple cores for one tunnel.
- You can split a fat GTP session to multiple sessions and distribute them to different cores. This helps to increase the bandwidth for fat GTP tunnel.

The TEID based hash distribution creates GTP-U sessions to multiple cores. The clear text traffic acts as a fat GTP tunnel. This helps a fat GTP session to split into multiple slim GTP sessions and handle them on multiple cores simultaneously.

GTP Traffic Performance with TEID Distribution and SWRSS

vSRX Virtual Firewall instances support Software Receive Side Scaling (SWRSS) feature. SWRSS is a technique in the networking stack to increase parallelism and improve performance for multi-processor systems. If NICs do not have sufficient number of queues as flow thread (FLT), based on vSRX Virtual Firewall type, then Software RSS (SWRSS) is enabled by flowd process.

With Software Receive Side Scaling (SWRSS) support on vSRX Virtual Firewall and vSRX Virtual Firewall 3.0, you can assign more vCPUs to the vSRX Virtual Firewall regardless of the limitation of RSS queue of underlying interfaces.

Based on SWRSS you can improve the GTP traffic performance using Tunnel endpoint identifier (TEID) distribution and asymmetric fat tunnel solution by:

- Assigning specific number of vCPUs for input output flow usage—With SWRSS enabled, you can assign more vCPUs for input/output (IO) threads when the IO threads are less. Or you can assign less vCPUs for IO threads if the flow process is consuming more vCPU. Use the `set security forwarding-options receive-side-scaling software-rss io-thread-number <io-thread-number>`.
- Distributing the packets to flow threads according to the TEID inside the packet, which would avoid reinjecting the packets in flow process—This feature is enabled when both SWRSS is enabled and when you configure the `set security forwarding-process application-services enable-gtpu-distribution` command.

  With this feature, the GTP packets would be distributed to the flow thread according to the hash value calculated by TEID. The algorithm of hash calculation is same as GTP distribution in flow module, which ensures the GTP packets would not be reinjected again in flow process.
- Utilizing fragment matching and forwarding mechanism in input/output thread when GTPU distribution is enabled—This mechanism ensures that all the fragments of the same packet would be distributed to one flow thread according to the TEID.

  SWRSS uses IP pair hash to distribute packets to flow threads. For GTP traffic with GTPU distribution enabled, TEID distribution is used to distribute packets to the flow threads. For fragmented packets, TEID cannot be retrieved from non-first fragments. This will require fragment matching and forwarding logic to ensure all fragments are forwarded to the flow thread based on TEID.

Enabling GTP-U TEID Distribution with SWRSS for Asymmetric Fat Tunnels

The following configuration helps you enable PMI and GTP-U traffic distribution with SWRSS enabled.

Before you begin, understand:

- SWRSS concepts and configurations.
- How to establish PMI and GTP-U

With Software Recieve Side Scaling (SWRSS) enabled, you can assign more vCPUs for input/output (IO) threads when the IO threads are less. Or you can assign less vCPUs for IO threads if the flow process is

consuming more vCPU. You can configure the number of IO threads required. With SWRSS is enabled and IO threads configured, reboot the vSRX Virtual Firewall for configuration to take effect. After IO threads are configured, distribute the GTP traffic to the configured IO threads according to TEID-based hash distribution for splitting a fat GTP session to multiple slim GTP sessions and process them on multiple cores in parallel.

NOTE: When PMI mode is enabled with TEID distribution and SWRSS support, performance of PMI is improved. If you want to enable PMI mode then run the `set securtiy flow power-mode-ipsec` command.

The following steps provide you details on how to enable SWRSS, configure IO threads, enable PMI mode for GTP sessions with TEID distribution for obtaining asymmetric fat tunnels:

1. SWRSS is enabled by default when NICs do not have sufficient number of queues as flow thread (FLT) based on vSRX Virtual Firewall type, then Software RSS (SWRSS) is enabled by flowd process. But, when SWRSS is not enabled use the following CLIs to enable. When you run this command SWRSS will be be enabled by force regardless of the NIC RSS or number of vCPUs.

   Enable SWRSS.

   ```
   [edit]
   user@host# set security forwarding-options receive-side-scaling software-rss mode enable
   ```

2. Configure the number of IO threads required. In this configuration we are configuring eight IO threads. The assigned number of vCPUs would be assigned for IO threads, and the rest vCPUs would be assigned for flow thread.

   ```
   [edit]
   user@host# set security forwarding-options receive-side-scaling software-rss io-thread-number
   8
   ```

3.

   ```
   [edit security]
   user@host# set flow power-mode-ipsec
   ```

4. Configure GTP-U session distribution.

   ```
   [edit security]
   user@host# set forwarding-process application-services enable-gtpu-distribution
   ```

5. From the configuration mode, confirm your configuration by entering the `show` command.

```
[edit security]
user@host# show
forwarding-options {
    receive-side-scaling {
        software-rss {
             mode enable;
        io-thread-number 8;
flow {
    power-mode-ipsec;
}
forwarding-process {
    application-services {
        enable-gtpu-distribution;
    }
}
```

From the operational mode run the following command to view the actual number of vCPUs assigned to IO/flow threads.

```
show security forward-options resource-manager settings
```

```
------------------------------------------------------------------------
Owner         Type                    Current settings      Next settings
SWRSS-IO      CPU core number         2                     2
SWRSS         SWRSS mode              Enable                Enable
```

6. Commit the configuration.

```
[edit security]
user@host# commit
```

7. Reboot the vSRX Virtual Firewall for the configuration to take effect. After rebooting the whole device, PFE would check the IO-thread value according to the NIC RSS queue and its memory.

CHAPTER 4

Monitoring and Troubleshooting

Monitoring

IN THIS SECTION

This topic provides details on how you can monitor your vSRX Virtual Firewall instances using SNMP and AWS monitoring features.

Monitoring helps in maintaining the reliability, availability, and performance of your vSRX Virtual Firewall instances and your AWS solutions. You should collect monitoring data from all your AWS solutions so that you can easily debug any multi-point failure.

Monitoring vSRX Virtual Firewall Instances Using SNMP

You can monitor your vSRX Virtual Firewall instance details such as health and storage at instance level, using SNMP monitoring.

For details on SNMP monitoring, refer the SNMP MIB information in the MIB Explorer at: https://apps.juniper.net/mib-explorer/.

You can also find all the applicable SNMP OIDs from the Juniper MIB from the vSRX Virtual Firewall CLI, using the `show snmp mib walk 1.3.6.1.4.1.2636` command.

Some examples of useful OID's for monitoring system health are:

```
jnxOperatingCPU.1.1.0.0
jnxOperating5MinAvgCPU.1.1.0.0
jnxFwddMicroKernelCPUUsage.0
jnxFwddRtThreadsCPUUsage.0
jnxHrStoragePercentUsed.1
jnxJsNodeCurrentTotalSession.0
jnxJsNodeMaxTotalSession.0
jnxJsNodeSessionCreationPerSecond.0
```

NOTE: For monitoring storage capacity on the vSRX Virtual Firewall instance you can use SNMP monitoring. Using SNMP monitoring, you can be notified for any vSRX Virtual Firewall instance storage that is impacted. The storage related OID indicates the storage percentage, which is used to detect the storage capacity.

For best practices for enabling SNMP monitoring in Junos, see https://www.juniper.net/documentation/en_US/junos/topics/ task/configuration/snmp-best-practices-basic-config.html.

Monitoring vSRX Virtual Firewall Instances Using AWS Features

AWS provides various tools that you can use to monitor Amazon EC2. You can configure some of the tools to do the monitoring for you, while some of the tools require manual intervention. For more information, see https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring_automated_manual.html.

Monitoring Your Instances Using CloudWatch—You can monitor your instances using Amazon CloudWatch, which collects and processes raw data from Amazon EC2 into readable, near real-time metrics. These statistics are recorded for a period of 15 months, so that you can access historical information and gain a better perspective on how your web application or service is performing. For more information see:

- **Monitoring Amazon EC2**—https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring_ec2.html.
- **Monitoring Your Instances Using CloudWatch**—https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html and https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-cloudwatch.html.

Backup and Recovery

This topic provides details on how you can backup and recover your configuration files in case of instance or service failure, both externally within AWS and locally on your vSRX Virtual Firewall instance console

To save the vSRX Virtual Firewall configuration file locally, perform the following steps:

1. Log into the vSRX Virtual Firewall instance and go to the configuration mode.

2. Execute the command `save /var/tmp/<file-name>`

   The current vSRX Virtual Firewall configurations are saved in the above mentioned path.

3. Using your Secure Copy Protocol (SCP) client, download the saved configuration files to your local system.

4. Using the instructions at https://aws.amazon.com/getting-started/tutorials/backup-files-to-amazon-s3/?trk=gs_card, create a S3 bucket on AWS and upload the saved configuration file. You can retrieve the saved configuration file as well.

For backup and recovery of configuration files within AWS:

NOTE: You must have an FTP server that is accessible from the vSRX Virtual Firewall instance.

1. Run the below configuration.

```
External example  system {
      archival {
        configuration {
            transfer-on-commit;
            archive-sites {
                "ftp://username:password@192.168.1.10";
                }
            }
      }
}
```

2. You can then run and commit the following configuration command on the vSRX Virtual Firewall instance.

```
set system archival configuration transfer-on-commit archive-sites ftp://
username:password@<FTP_Server_IP_Address>.
```

Finding the Software Serial Number for vSRX Virtual Firewall

You need the software serial number to open a support case or to renew a vSRX Virtual Firewall license.

The serial number is a unique 14-digit number that Juniper Networks uses to identify your particular software installation. You can find the software serial number in the Software Serial Number Certificate attached to the e-mail that was sent when you ordered your Juniper Networks software or license. You can also use the show system license command to find the software serial number.

Use the `show system license` command to find the vSRX Virtual Firewall software serial number.

```
vsrx> show system license
```

```
License usage:
                                Licenses     Licenses    Licenses    Expiry
  Feature name                       used    installed      needed
  Virtual Appliance                     1            1           0    58 days

Licenses installed:
  License identifier: E420588955
  License version: 4
  Software Serial Number: 20150625
  Customer ID: vSRX-JuniperEval
  Features:
    Virtual Appliance - Virtual Appliance
      count-down, Original validity: 60 days

  License identifier: JUNOS657051
  License version: 4
  Software Serial Number: 9XXXXAXXXXXXX9
  Customer ID: MyCompany
  Features:
    Virtual Appliance - Virtual Appliance
      permanent
```

For more information, see Licenses for vSRX