

Technology Overview

Best Practices for Managing Policy Memory on High-End SRX Series Devices

Release

11.4



Published: 2011-12-07

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986-1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by the Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, the Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., copyright © 1996, 1997, Maker Communications, Inc.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

Technology Overview Best Practices for Managing Policy Memory on High-End SRX Series Devices

Release 11.4

Copyright © 2011, Juniper Networks, Inc.

All rights reserved.

Revision History

October 2011—R1 Junos OS 11.4

The information in this document is current as of the date listed in the revision history.

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 <http://www.juniper.net/support/eula.html>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

Introduction	1
Global Policy Overview	3
Best Practices for Defining Policies on High-End SRX Series Devices	5
Example: Configuring Global Policy	9
Checking Memory Status	13

Introduction

This document provides an overview of global policies, describes how policy memory is calculated, provides best practice recommendations for managing policy memory, and explains how to monitor policy memory on high-end SRX Series Services Gateways.

Global Policy Overview

In a Junos OS stateful firewall, security policies enforce rules for transit traffic, in terms of what traffic can pass through the firewall, and the actions that need to take place on traffic as it passes through the firewall. Security policies require traffic to enter one security zone and exit another security zone. This combination of a from-zone and to-zone is called a *context*. Each context contains an ordered list of policies. Each policy is processed in the order that it is defined within a context.

You can configure a security policy from the user interface. Security policies control traffic flow from one zone to another zone by defining the kind(s) of traffic permitted from specific IP sources to specific IP destinations at scheduled times. This works well in most cases, but it is not flexible enough. For example, if you want to perform actions on traffic but do not care about the zones (that is, you want to permit all traffic to access a given server in the DMZ), you have to configure policies for each possible context. To avoid creating multiple policies across every possible context, you can create a global policy. Global policies provide you with the flexibility to perform actions on traffic without the restrictions of zone specifications.

Unlike other security policies, global policies do not reference specific source and destination zones (from-zone and to-zone). Global policies allow you to regulate traffic with addresses and applications, regardless of their security zones. Global policies reference user-defined addresses or the predefined address “any.” These addresses can span multiple security zones. For example, if you want to provide access to or from multiple zones, you can create a global policy with the address “any,” which encompasses all addresses in all zones. Selecting the “any” address matches any IP address, and when “any” is used as a source/destination address in any global policy configuration, it matches the source/destination address of any packet.

Traffic is classified by matching the policy’s source address, destination address, and the application that the traffic carries in its protocol header. Each global policy, as with any other security policy, has the following actions: permit, deny, reject, log, count.



NOTE: Global policies do not support VPN tunnels because VPN tunnels require specific zone information (from-zone and to-zone).

Global policies in one logical system are in a separate context than other security policies and have a lower priority than regular security policies in a policy lookup. For example, if a policy lookup is performed, regular security policies have priority over global policies. Therefore, in a policy lookup, regular security policies are searched first and if there is no match, global policy lookup is performed.

Similar to regular policies, global policies in a context are ordered, such that the first matched policy is applied to the traffic.

You can define global policies for each logical system.

**Related
Documentation**

- Security Policies Overview
- Understanding Security Policy Rules
- Understanding Security Policy Elements
- [Example: Configuring Global Policy on page 9](#)

Best Practices for Defining Policies on High-End SRX Series Devices

A secure network is vital to a business. To secure a network, a network administrator must create a security policy that outlines all of the network resources within that business and the required security level for those resources. The security policy applies the security rules to the transit traffic within a context (**from-zone** to **to-zone**) and each policy is uniquely identified by its name. The traffic is classified by matching the source and destination zones, the source and destination addresses, and the application that the traffic carries in its protocol headers with the policy database in the data plane.

SRX1400, SRX3400, SRX3600, SRX5600, and SRX5800 devices support up to 1024 source address objects, 1024 destination address objects, and 128 application objects. These objects can be used in either a single policy rule or multiple policy definitions. SRX1400 devices allow you to define a maximum of 10,000 policy rules, SRX3400 and SRX3600 devices allow 40,000, and SRX5600 and SRX5800 devices allow a maximum of 80,000 policy rules. However, you can have up to 1024 source or destination address objects and 128 application objects, regardless of the number of policies defined.

Therefore, as you increase the number of addresses and applications in each rule, the amount of memory that is used by the policy definition increases, and sometimes the system runs out of memory with fewer than 80,000 policies.



NOTE: We recommend using global policies whenever possible. Global policies provide you with the flexibility to perform actions on traffic without the restrictions of zone specifications.

To get the actual memory utilization of a policy on the Packet Forwarding Engine (PFE) and the Routing Engine (RE), you need to take various components of the memory tree into consideration. The memory tree includes the following two components:

- Policy context—Used to organize all policies in this context. Policy context includes variables like source and destination zones.
- Policy entity—Used to hold the policy data. Policy entity calculates memory using parameters such as policy name, IP addresses, address count, applications, firewall authentication, WebAuth, IPsec, count, application services, and Junos Services Framework (JSF).

Additionally, the data structures used to store policies, rule sets, and other components use different memory on the Packet Forwarding Engine and on the Routing Engine. For example, address names for each address in the policy are stored on the Routing Engine, but no memory is allocated at the Packet Forwarding Engine level. Similarly, port ranges are expanded to prefix and mask pairs and are stored on the Packet Forwarding Engine, but no such memory is allocated on the Routing Engine.

Accordingly, depending on the policy configuration, the policy contributors to the Routing Engine are different from those to the Packet Forwarding Engine, and memory is allocated dynamically.

Memory is also consumed by the “deferred delete” state. In the deferred delete state, when an SRX Series device applies a policy change, there is transitory peak usage whereby both the old and new policies are present. So for a brief period, both old and new policies exist on the Packet Forwarding Engine, taking up twice the memory requirements.

Therefore, there is no definitive way to infer clearly how much memory is used by either component (Packet Forwarding Engine or Routing Engine) at any given point in time, because memory requirements are dependent on specific configurations of policies, and memory is allocated dynamically.

The following best practices for policy implementation allow you to better use system memory and to optimize policy configuration:

- Use single prefixes for source and destination addresses. For example, instead of using /32 addresses and adding each address separately, use a large subnet that covers most of the IP address you require.
- Use application “any” whenever possible. Each time you define an individual application in the policy, you can use an additional 52 bytes.
- Use fewer IPv6 addresses since IPv6 addresses consume more memory.
- Use fewer zone pairs in policy configurations. Each source or destination zone uses about 16,048 bytes of memory.
- The following parameters can change how memory is consumed by the bytes as specified:
 - Firewall authentication—About 16 bytes or more (unfixed)
 - Web authentication—About 16 bytes or more (unfixed)
 - IPsec—12 bytes
 - Application services—28 bytes
 - Count—64 bytes
- Use global policies because they decrease overall memory usage in terms of source and destination prefixes and applications used.
- Check memory utilization before and after compiling policies.



NOTE: The memory requirement for each device is different. Some devices support 512,000 sessions by default and the bootup memory is usually at 72 to 73 percent. Other devices can have up to 1M sessions and the bootup memory can be up to 83 to 84 percent. In the worst-case scenario, to support about 80,000 policies in the SPU, the SPU should boot up with a flowd kernel memory consumption of up to 82 percent, and with at least 170M bytes of memory available.

**Related
Documentation**

- Security Policies Overview

- Understanding Security Policy Rules
- Understanding Global Address Books
- [Global Policy Overview on page 3](#)
- [Example: Configuring Global Policy on page 9](#)
- [Checking Memory Status on page 13](#)

Example: Configuring Global Policy

Unlike other security policies in Junos OS, global policies do not reference specific source and destination zones. Global policies reference the predefined address “any” or user-defined addresses that can span multiple security zones. Global policies give you the flexibility of performing actions on traffic without any zone restrictions. For example, you can create a global policy so that every host in every zone can access the company website, www.juniper.net. Using a global policy is a convenient shortcut when there are many security zones. Traffic is classified by matching its source address, destination address, and the application that the traffic carries in its protocol header.

This example shows how to configure a global policy to deny or permit traffic.

- [Requirements on page 9](#)
- [Overview on page 9](#)
- [Configuration on page 9](#)
- [Verification on page 11](#)

Requirements

Before you begin:

- Review the firewall security policies.
See Security Policies Overview, Understanding Security Policy Rules, and Understanding Security Policy Elements.
- Configure an address book and create addresses for use in the policy.
See Example: Configuring Address Books and Address Sets.
- Create an application (or application set) that indicates that the policy applies to traffic of that type.
See Example: Configuring Applications and Application Sets.

Overview

This configuration example shows how to configure a global policy that accomplishes what multiple security policies (using zones) would have accomplished. Global policy gp1 permits all traffic while policy gp2 denies all traffic.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set security address-book global address server1 www.juniper.net
set security address-book global address server2 www.mail.com
set security policies global policy gp1 match source-address server1
```

```
set security policies global policy gp1 match destination-address server2
set security policies global policy gp1 match application any
set security policies global policy gp1 then permit
set security policies global policy gp2 match source-address server2
set security policies global policy gp2 match destination-address server1
set security policies global policy gp2 match application junos-ftp
set security policies global policy gp2 then deny
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the [Junos OS CLI User Guide](#).

To configure a global policy to permit or deny all traffic:

1. Create addresses.

```
[edit security]
user@host# set security address-book global address server1 www.juniper.net
user@host# set security address-book global address server2 www.mail.com
```

2. Create the global policy to permit all traffic.

```
[edit security]
user@host#set policy global policy gp1 match source-address server1
user@host#set policy global policy gp1 match destination-address server2
user@host#set policy global policy gp1 match application any
user@host#set policy global policy gp1 then permit
```

3. Create the global policy to deny all traffic.

```
[edit security]
user@host#set policy global policy gp2 match source-address server2
user@host#set policy global policy gp2 match destination-address server1
user@host#set policy global policy gp2 match application junos-ftp
user@host#set policy global policy gp2 then deny
```

Results From configuration mode, confirm your configuration by entering the **show security policies** and **show security policies <global>** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show security policies
Default policy: permit-all
Global policies:
  Policy: gp1, State: enabled, Index: 4, Scope Policy: 0, Sequence number: 1
    Source addresses: server1
    Destination addresses: server2
    Applications: any
    Action: permit
  Policy: gp2, State: enabled, Index: 5, Scope Policy: 0, Sequence number: 2
    Source addresses: server2
    Destination addresses: server1
    Applications: junos-ftp
    Action: deny
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Verifying Global Policy Configuration

Purpose Verify that global policies p1 and p2 are configured as required.

Action From operational mode, enter the **show security policy <global>** command.

Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Junos OS CLI Reference](#)
- Security Policies Overview

Checking Memory Status

Memory for flow entities (for example, policies, zones, addresses) on SRX1400, SRX3400, SRX3600, SRX5600, and SRX5800 devices is dynamically allocated. However, certain practices can help monitor the current memory usage on the device and optimize parameters to better size system configuration, especially during policy implementation.

You can isolate memory issues by comparing memory values before and after policy configurations.

To check memory usage:

- Use the **show chassis routing-engine** command to check overall Routing Engine (RE) memory usage. The following output from this command shows memory utilization at 39 percent:

```
user@host# show chassis routing-engine
Routing Engine status:
Slot 0:
  Current state           Master
  Election priority       Master (default)
  DRAM                    1024 MB
  Memory utilization       39 percent
  CPU utilization:
    User                  0 percent
    Background            0 percent
    Kernel                 2 percent
    Interrupt             0 percent
    Idle                  97 percent
  Model                   RE-PPC-1200-A
  Start time              2011-07-09 19:19:49 PDT
  Uptime                  37 days, 15 hours, 44 minutes, 13
seconds
  Last reboot reason      0x3:power cycle/failure watchdog
  Load averages:         1 minute  5 minute 15 minute
                        0.22      0.16   0.07
```

- Use the **show system processes extensive** command to acquire information on the processes running on the Routing Engine.

```
user@host# show system processes extensive
last pid: 13469; load averages: 0.10, 0.12, 0.07 up 37+15:46:43
11:06:02
127 processes: 3 running, 97 sleeping, 27 waiting

Mem: 203M Active, 47M Inact, 107M Wired, 343M Cache, 110M Buf, 282M Free
Swap: 2032M Total, 2032M Free
```

PID	USERNAME	THR	PRI	NICE	SIZE	RES	STATE	TIME	WCPU	COMMAND
11	root	1	171	52	OK	16K	RUN	863.7H	96.14%	idle
1128	root	1	97	0	32768K	13852K	select	19.1H	1.12%	chassisd
12	root	1	-20	-139	OK	16K	RUN	473:40	0.00%	swi7:clock
1038	root	1	76	0	OK	16K		369:31	0.00%	bcmLINK.0
1158	root	1	96	0	7288K	3144K	select	62:31	0.00%	

```

license-check
1162 root      1  96    0 13684K  6444K select  48:11  0.00% 12ald
1188 root      1  96    0 19248K  8068K select  47:54  0.00% mib2d
1147 root      1  96    0 12572K  2592K select  31:20  0.00% shm-rtssdbd

1187 root      2  96    0 28948K  5944K select  22:15  0.00% pfed
1152 root      1  96    0 47476K  8104K select  18:28  0.00% idpd
1189 root      1  96    0 18744K 13684K select  15:53  0.00% snmpd
1129 root      1  96    0  7684K  3384K select  14:57  0.00% alarmd
  14 root      1 -40 -159    0K    16K WAIT   14:29  0.00% swi2:

net
1125 root      1  96    0  2120K   932K select  13:14  0.00% bslockd
  25 root      1 -68 -187    0K    16K WAIT   12:40  0.00% irq36:

tsec1
1161 root      1   4    0 43320K 10948K kqread   8:41  0.00% rpd
  57 root      1 -16    0    0K    16K -        6:40  0.00% schedcpu

```

Use the **find nsd** option in the **show system processes extensive** command to see direct usage on the Network Security Daemon (NSD) with its total memory in use as 10 megabytes and CPU utilization of 0 percent.

```

user@host# show system processes extensive | find nsd
1182 root      1  96    0 10976K  5676K select   2:08  0.00% nsd
1191 root      4   4    0  8724K  3764K select   1:57  0.00% slbd
1169 root      1  96    0  8096K  3520K select   1:51  0.00% jsrpd
1200 root      1   4    0    0K    16K peer_s   1:10  0.00% peer

proxy
1144 root      1  96    0  9616K  3528K select   1:08  0.00% lacpd
1138 root      1  96    0  6488K  2932K select   1:02  0.00% ppmd
1130 root      1  96    0  7204K  2208K select   1:02  0.00% craftd
1163 root      1  96    0 16928K  5188K select   0:58  0.00% cosd
1196 root      1   4    0    0K    16K peer_s   0:54  0.00% peer

proxy
  47 root      1 -16    0    0K    16K sdfus    0:54  0.00%

softdepflush
1151 root      1  96    0 15516K  9580K select   0:53  0.00% appidd
  900 root      1  96    0  5984K  2876K select   0:41  0.00% eventd

```

- Check the configuration file size. Save your configuration file with a unique name before exiting the CLI. Then, enter the **ls -l filename** command from the shell prompt in the UNIX-level shell to check the file size as shown in the following sample output:

```

user@host> start shell
% ls -l config
-rw-r--r--  1 remote  staff 12681 Feb 15 00:43 config

```

Related Documentation

- [Best Practices for Defining Policies on High-End SRX Series Devices on page 5](#)
- Security Policies Overview
- Understanding Security Policy Elements
- [Global Policy Overview on page 3](#)
- [Example: Configuring Global Policy on page 9](#)