

# **|JUNIPER SWITCHING| |LAB EXERCISE 4|**

Routing Policy and Firewall Filters



# Juniper Networks

## Enterprise Switching Summer School

---

### 9.b

#### Lab 4: Routing Policy and Firewall Filters

Juniper Networks, the Juniper Networks logo, JUNOS, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. JUNOSe is a trademark of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

*OJXE Summer School Detailed Lab Guide*

Copyright © 2009 Juniper Networks, Inc.

All rights reserved. Printed in USA.

Revision History:

June 2009

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

The information in this document has been carefully verified and is believed to be accurate for software Release 9.4R2.9. Juniper Networks assumes no responsibilities for any inaccuracies that may appear in this document. In no event will Juniper Networks be liable for direct, indirect, special, exemplary, incidental or consequential damages resulting from any defect or omission in this document, even if advised of the possibility of such damages.

Juniper Networks reserves the right to change, modify, transfer or otherwise revise this publication without notice.

YEAR 2000 NOTICE

Juniper Networks hardware and software products do not suffer from Year 2000 problems and hence are Year 2000 compliant. The JUNOS Software has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

SOFTWARE LICENSE

The terms and conditions for using Juniper Networks software are described in the software license provided with the software, or to the extent applicable, in an agreement executed between you and Juniper Networks, or Juniper Networks agent. By using Juniper Networks software, you indicate that you understand and agree to be bound by its license terms and conditions. Generally speaking, the software license restricts the manner in which you are permitted to use the Juniper Networks software, may contain prohibitions against certain uses, and may state conditions under which the license is automatically terminated. You should consult the software license for further details.

# Contents

General Lab Instructions .....	4
Lab 4 .....	5
Routing Policy and Firewall Filters (Detailed).....	5
Overview .....	5
Part 1: Enabling OSPF.....	5
Part 2: Creating Static Routes to Redistribute into OSPF .....	8
Part 3: Configuring a Firewall Filter.....	11

## General Lab Instructions

---

Your lab pod consists of two Juniper Networks EX-4200 switches. Switches are connected together over 3 GE interfaces: ge-0/0/10, Ge-0/0/11 and ge-0/0/12 according to the lab diagram. You do not need to configure additional physical interfaces in the lab exercises. Switches are also interconnected over the OoB Management Network via switches' me0 interfaces.

You gain access to the switches via three different ways:

- Console port, CLI
- Management interface, me0, CLI
- Management interface, me0, J-Web

Access to each of these user interfaces is explained in lab access instructions sent to you via e-mail. Make sure you do not change the IP address of the me0.0 logical interface. This would cut your telnet/HTTP session after committing the changes. If that should happen log in via the Console port and issue **rollback 1** command followed by **commit** command. Also be careful not to change the hostname of the switch. That will make it hard for you to follow the lab instructions in this Lab guide where we normally refer the pod switches as **POD\_NAME-S1** and **POD\_NAME-S2**. Pods are labeled **A, B, C** and **D** thus the individual switches are labeled **A-S1 & A-S2, B-S1 & B-S2, C-S1 & C-S2** and **D-S1 & D-S2**.

When given instructions to configure a switch, the instructions normally refer to both of your pod's switches. In special cases you only need to configure one of the two switches. In these sections the specific switch will be indicated in the lab exercise by name **S1** or **S2**.

Note that your lab login (password *lab123*) grants you all permissions needed to complete this lab; but some restrictions have been made to prevent loss of connectivity to the devices. Please be careful, and have fun!

The time needed to complete each of the 4 lab exercise sets may vary very much between each individual student. Some of you might finish the tasks in less than the scheduled two hours. Some of you might not have time to complete them. If you get stuck in an exercise or do not fully understand the meaning of a task use the Detailed Lab Guide where all commands and outputs as well the answers for all questions are presented for you.

# Lab 4

---

## Routing Policy and Firewall Filters (Detailed)

### Overview

---

In this lab you configure and monitor Layer 3 routing features using the JUNOS Software command-line interface (CLI). You will enable and verify OSPF routing including the redistribution of static routes into OSPF through a routing policy. You will also familiarize yourself with the use of firewall filters.

This lab is available in two formats: a high-level format that is designed to make you think through each step and a detailed format that offers step-by-step instructions complete with sample output from most commands.

By completing this lab, you will perform the following tasks:

- Enable OSPF.
- Create static routes to redistribute into OSPF.
- Configure a firewall filter.

## Part 1: Enabling OSPF

---

### Step 1.1

Before starting this Lab you need to ensure the switches are in ready-to-go state for Lab 4. Load both switches with the Lab4-start file from /JSS directory using following command:

```
{master:0}[edit]
lab@D-S1# load merge /JSS/Lab4-start
```

```
{master:0}[edit]
lab@D-S1# commit
```

We will use OSPF as our routing protocol. Enable OSPF on all VLAN interfaces and the lo0 interface. The area number is 0. Do not forget to commit your changes.

```
edit protocols ospf]
lab@A-S1# show
area 0.0.0.0 {
    interface lo0.0;
    interface vlan.1;
    interface vlan.2;
    interface vlan.3;
    interface vlan.4;
}
```

### Step 1.2

Question: Do you see any neighbors behind the OSPF interfaces? What is the state of the interfaces?

---

---

Answer: Yes, the prerequisite is that both switches has been configured to run OSPF on loopback and the logical VLAN interfaces in OSPF area 0. Interface state depends on when it has been activated to OSPF operations, either DR or BDR.

Question: Check your switches' OSPF neighbors. What is the state of the OSPF adjacency with the neighbors?

---

---

Answer: Interfaces that have neighbors should also display these neighbors and the state of the adjacency. Example here below shows each of the vlan interfaces to have full adjacency with their respective neighbors.

To check your own configuration and the results of it, use **show ospf interface**. To see if you are exchanging information with your neighbors, use **show ospf neighbor**.

```
lab@A-S1> show ospf interface
```

Interface	State	Area	DR ID	BDR ID	Nbrs
lo0.0	DR	0.0.0.0	192.168.1.1	0.0.0.0 0	
vlan.1	BDR	0.0.0.0	192.168.1.2	192.168.1.1	1
vlan.2	DR	0.0.0.0	192.168.1.1	192.168.1.2	1
vlan.3	DR	0.0.0.0	192.168.1.1	192.168.1.2	1
vlan.4	DR	0.0.0.0	192.168.1.1	192.168.1.2	1

```
lab@A-S1> show ospf neighbor
```

Address	Interface	State	ID	Pri	Dead
10.1.1.2	vlan.1	Full	192.168.1.2	128	31
10.1.2.2	vlan.2	Full	192.168.1.2	128	33
10.1.3.2	vlan.3	Full	192.168.1.2	128	34
10.1.4.2	vlan.4	Full	192.168.1.2	128	32

### Step 1.3

Question: Are the switches learning any routes via OSPF?

---

---

Answer: Yes. Results may vary depending the configuration. The example below displays 2 routes learned via OSPF.

To see the result of the OSPF route exchange, issue the **show route protocol ospf** command.

```
lab@A-S1> show route protocol ospf
```

```
inet.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both
```

```
10.200.1.0/24      *[OSPF/150] 00:05:16, metric 100, tag 0
                   to 10.1.1.2 via vlan.1
                   > to 10.1.2.2 via vlan.2
                   to 10.1.3.2 via vlan.3
                   to 10.1.4.2 via vlan.4
```

```
10.200.2.0/24      *[OSPF/150] 00:05:16, metric 100, tag 0
                   > to 10.1.1.2 via vlan.1
                   to 10.1.2.2 via vlan.2
                   to 10.1.3.2 via vlan.3
                   to 10.1.4.2 via vlan.4
```

```
...
```

### Step 1.4

All protocol configuration options are in the same part of the configuration hierarchy. To illustrate this concept, make the VLAN v4 interface passive in OSPF. Change the VLAN v2 interface into a point-to-point interface-type to optimize the neighborship.

```
lab@A-S1# show protocols ospf | display set
```

```
set protocols ospf area 0.0.0.0 interface lo0.0
```

```
set protocols ospf area 0.0.0.0 interface vlan.1
```

```
set protocols ospf area 0.0.0.0 interface vlan.2 interface-type p2p
```

```
set protocols ospf area 0.0.0.0 interface vlan.3
set protocols ospf area 0.0.0.0 interface vlan.4 passive
```

## Step 1.5

Question: Can you verify that the changes are active; p2p and passive interface?

---

Answer: Yes. Both changes are active in the example output here below. Use the **show ospf interface detail** command to display the detailed OSPF interface configuration and operations information.

```
lab@A-S1# run show ospf interface detail
Interface      State   Area      DR ID          BDR ID          Nbrs
vlan.2         PtToPt 0.0.0.0    0.0.0.0        0.0.0.0         1
Type: P2P, Address: 10.1.2.1, Mask: 255.255.255.0, MTU: 1500, Cost: 1
  Adj count: 1
  Hello: 10, Dead: 40, ReXmit: 5, Not Stub
  Auth type: None
  Topology default (ID 0) -> Cost: 0
vlan.4         DROther 0.0.0.0    0.0.0.0        0.0.0.0         0
Type: LAN, Address: 10.1.4.1, Mask: 255.255.255.0, MTU: 1500, Cost: 1
  Adj count: 0, Passive
  Hello: 10, Dead: 40, ReXmit: 5, Not Stub
  Auth type: None
  Topology default (ID 0) -> Passive, Cost: 0
```

## Part 2: Creating Static Routes to Redistribute into OSPF

---

### Step 2.1

Create the four static routes assigned to your device (see diagram) with a next hop of `reject`. The reason for these routes is to have some routes to redistribute into OSPF.

```
[edit routing-options]
lab@A-S1# show
static {
  route 0.0.0.0/0 {
    next-hop 192.168.227.1;
    no-readvertise;
  }
  route 10.100.1.0/24 reject;
  route 10.100.2.0/24 reject;
  route 10.100.3.0/24 reject;
  route 10.100.4.0/24 reject;
}
```

### Step 2.2

Question: Can you see the static routes in your switch's routing table `inet.0`?

---

---

Answer: Yes. The **show route protocol static** command should show all four routes as active (\*).

Question: What is the preference value and the next-hop of these static routes?

---

---

Answer: 5 and reject. Use the **show route protocol static** command to display the static routing information.

```
lab@A-S1# run show route protocol static
inet.0: 22 destinations, 22 routes (22 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[Static/5] 07:15:31
                   > to 192.168.227.1 via me0.0
10.100.1.0/24     *[Static/5] 00:01:40
                   Reject
10.100.2.0/24     *[Static/5] 00:01:40
                   Reject
10.100.3.0/24     *[Static/5] 00:01:40
                   Reject
10.100.4.0/24     *[Static/5] 00:01:40
                   Reject
```

### Step 2.3

Write a policy to allow the static route redistribution into OSPF. OSPF should accept the first two routes with a metric of 100, the other two routes should have a metric of 200. Apply the policy as an export policy to OSPF. (Remember that the Tab key gives you variable completion!)

```
[edit policy-options policy-statement Advertise_Statics_to_OSPF]
lab@A-S1# show
term 1 {
  from {
    protocol static;
    route-filter 10.100.1.0/24 exact;
    route-filter 10.100.2.0/24 exact;
  }
  then {
    metric 100;
    accept;
  }
}
term 2 {
  from {
    protocol static;
    route-filter 10.100.3.0/24 exact;
    route-filter 10.100.4.0/24 exact;
  }
}
```

```

    then {
        metric 200;
        accept;
    }
}

[edit protocols ospf]
lab@A-S1# set export Advertise_Statics_to_OSPF

```

## Step 2.4

Question: Are the static routes been distributed into OSPF?

---



---

Answer: Yes. The **show route protocol static** command should show all four routes as active (\*).

Question: What is the preference value and the next-hop of these static routes?

---



---

Answer: 5 and reject. Use the **show route protocol static** command to display the static routing information.

Follow this step for verification:

You can see static routes that were exported into OSPF with the **show ospf database external** command. Your redistributed routes show up with an asterisk (\*) next to them.

```

lab@A-S1# run show ospf database external
      OSPF AS SCOPE link state database
Type      ID          Adv Rtr      Seq          Age      Opt  Cksum  Len
Extern    *10.100.1.0  192.168.1.1 0x80000001  8        0x22 0x91e1 36
Extern    *10.100.2.0  192.168.1.1 0x80000001  8        0x22 0x86eb 36
Extern    *10.100.3.0  192.168.1.1 0x80000001  8        0x22 0x67a5 36
Extern    *10.100.4.0  192.168.1.1 0x80000001  8        0x22 0x5caf 36
Extern    10.200.1.0   192.168.1.2 0x8000000c  537      0x22 0xc042 36
Extern    10.200.2.0   192.168.1.2 0x8000000b  1954     0x22 0xb74b 36
Extern    10.200.3.0   192.168.1.2 0x8000000b  1467     0x22 0x9805 36
Extern    10.200.4.0   192.168.1.2 0x8000000b  991      0x22 0x8d0f 3 6

```

To see the correct metric of your routes, use the **show ospf database external detail** command.

```

lab@A-S1# run show ospf database external detail
      OSPF AS SCOPE link state database
Type      ID          Adv Rtr      Seq          Age      Opt  Cksum  Len
Extern    *10.100.1.0  192.168.1.1 0x80000001  294      0x22 0x91e1 36
      mask 255.255.255.0

```

```

Topology default (ID 0)
  Type: 2, Metric: 100, Fwd addr: 0.0.0.0, Tag: 0.0.0.0
Extern      *10.100.3.0 192.168.1.1      0x80000001 294      0x22 0x67a5  36
mask 255.255.255.0
Topology default (ID 0)
  Type: 2, Metric: 200, Fwd addr: 0.0.0.0, Tag: 0.0.0.0

```

## Part 3: Configuring a Firewall Filter

---

### Step 3.1

Using a firewall filter on the lo0 interface is an efficient way to filter all traffic to and from the RE. Configure a firewall filter to count and discard all pings to your own lo0 interface address. Also, count and discard all SSH traffic to any of your (VLAN) interface addresses.

```

lab@A-S1# show firewall | display set
set firewall family inet filter Lo0_FF term 1 from destination-address
192.168.1.1/32
set firewall family inet filter Lo0_FF term 1 from protocol icmp
set firewall family inet filter Lo0_FF term 1 from icmp-type echo-request
set firewall family inet filter Lo0_FF term 1 then count Ping_Block
set firewall family inet filter Lo0_FF term 1 then discard
set firewall family inet filter Lo0_FF term 2 from protocol tcp
set firewall family inet filter Lo0_FF term 2 from destination-port ssh
set firewall family inet filter Lo0_FF term 2 then count SSH_Block
set firewall family inet filter Lo0_FF term 2 then discard
[edit]

```

```
lab@A-S1# set interfaces lo0.0 family inet filter input Lo0_FF
```

Follow this step for verification:  
Generate some of the traffic that the firewall blocks. After testing, you can issue the **show firewall** command.

```

lab@A-S1> show firewall
Filter: Lo0_FF
Counters:
Name Bytes Packets
Ping_Block 550 5
SSH_Block 180 2

```

### Step 3.2

The firewall filter has an implicit discard any at the end, so the firewall filter in the previous step will block all other traffic including the routing protocols.

```
lab@A-S1> show ospf neighbor
```

To fix this issue, add a new term to the existing firewall filter that will allow all other traffic.

```

[edit firewall family inet filter Lo0_FF]
lab@A-S1# show term 3
then accept;

```

Follow this step for verification:  
Check if the OSPF neighbors are in the Full state again using the **show ospf neighbor** command. Remember that vlan.4 is passive.

```
lab@A-S1> show ospf neighbor  
Address Interface State ID Pri Dead  
10.1.1.2 vlan.1 Full 192.168.1.2 128 35  
10.1.2.2 vlan.2 Full 192.168.1.2 128 33  
10.1.3.2 vlan.3 Full 192.168.1.2 128 31
```