6DISS

South America Workshop

WALC 2006 (Quito, Ecuador – 26-28 July '06)

IPv6 Security

Pedro Lorga (lorga@fccn.pt)

Miguel Baptista (miguel.baptista@fccn.pt)

Simon Muyal (muyal@renater.pt)

Laboratory Exercise: *IPv6 Security*

Objectives

In this laboratory exercise you will complete the following tasks:

- *Configure filters for IPv6 traffic;*
- *Access Control to VTY lines;*
- *IPv6 Firewall*
- *Analyzing and troubleshooting problems*

Visual Objective

You will use OSPF scenario you previously saved for this exercise (all routers in Area 0).

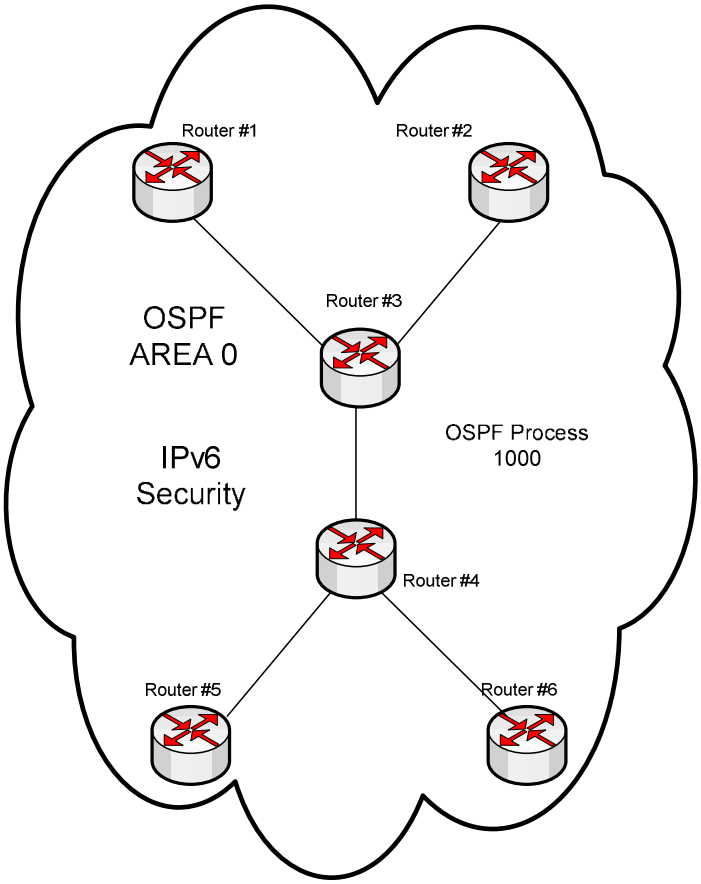

Figure 1 – IPv6 security scenario

Setup/Scenario

There will be 4 students per router. The distribution will be the same as in the OSPFv3 exercise.

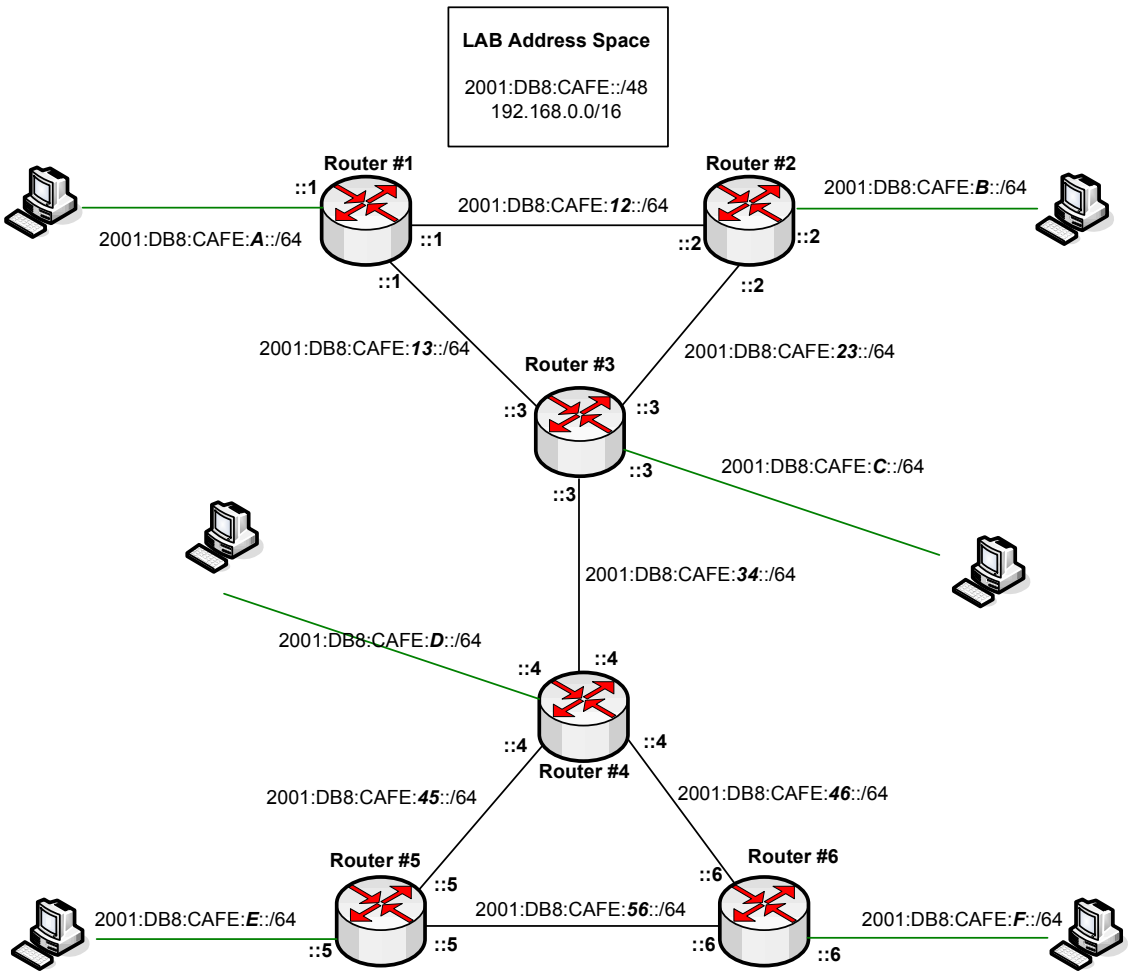

Figure 2 – Scenario Topology

Preparing the LAB

If your router is not yet configured, load the configuration from the flash memory:

```
RouterX# copy flash:iniv6-OSPF running-config
RouterX# wr
```

Task 1: *IPv6 traffic filters*

Note that the routers you are using in these exercises have IOS that support the mentioned features. At home remember to check them, as you may not have all functionalities.

Several different methods of configuring access-lists have been implemented in Cisco IOS over the past years. In this module we will use the latest approach. The way used to configure access-list in the routers in this module is:

```
Router3# conf t
Router3#(config)# ipv6 access-list <name_access_list>
Router3#( config-ipv6-acl)# <deny/permit> …
```

The **permit** clause is:

```
permit protocol {source-ipv6-prefix/prefix-length | any | host
source-ipv6-address} [operator [port-number]] {destination-ipv6-
prefix/prefix-length | any | host destination-ipv6-address}
[operator [port-number]] [dest-option-type [doh-number | doh-
type]] [dscp value] [flow-label value] [fragments] [log] [log-
input] [mobility] [mobility-type [mh-number | mh-type]] [reflect
name [timeout value]] [routing] [routing-type routing-number]
[sequence value] [time-range name]
```

The **deny** clause is:

```
deny protocol {source-ipv6-prefix/prefix-length | any | host
source-ipv6-address} [operator [port-number]] {destination-ipv6-
prefix/prefix-length | any | host destination-ipv6-address}
[operator [port-number]] [dest-option-type [doh-number | doh-
type]] [dscp value] [flow-label value] [fragments] [log] [log-
input] [mobility] [mobility-type [mh-number | mh-type]]
[routing] [routing-type routing-number] [sequence value] [time-
range name] [undetermined-transport]
```

Step 1: Configure Access-lists

Configure an access-list to only permit incoming IPv6 traffic from your direct neighbor(s) (see figure 2). Insert them in the *permit* clause of your IPv6 access-list.

Step 2: Applying Access Lists to your interfaces

Apply the access-list to you interfaces in the *in* direction.

(**Tip**: ipv6 traffic-filter…)

Can they ping you now?

Check the OSPF process. Is it up?

(**Tip**: allow the link-local addresses also in your configuration)

Try to *ping* again.

Step 3: Checking your Access-lists

Explicitly deny any IPv6 ICMP on your access-list. Ask anyone (not directly connected to you) to ping your router. See in what access-list line are the requests being counted.

(**Tip**: show ipv6 access-list …)

Although you deny any ICMP, why can your neighbour ping your router and the others don't?

Step 4: Allowing a Port Number and a Protocol

Create a new access-list. Instead of permitting all IPv6, only allow the *telnet* port from your neighbour(s).

(**Tip**: permit ipv6 tcp ….)

Apply this access-list to the interface. Try to ping the router.

Add another line to the access-list, adding the protocol ICMP to the allowed conditions.

Can you ping now? Check your access-list and see what packets are matching your configuration.

Step 5: Controlling access to VTY Lines

As you know, you should restrict access to your routers, to only a small number of PCs or networks that you control.

Create an IPv6 access-list to only allow your direct neighbours to access your network. Use only the source IP address, don't use the protocol configuration.

(Tip: permit ipv6 …)

Now apply this access-list to your VTY.

(**Tip**: line vty 0 <ending_VTY

access-class …)

Step 6: IPv6 Firewall

The latest IOS allows you take advantage of some new features, like firewall. With it you can both inspect packets and use ACL, described previously. For you to see the output of this exercise, you will need a FTP server and client.

Configure your router to inspect FTP packets.

(**Tip**: ipv6 inspect name …)

Apply this rule to the interface you wish.

(**Tip**: ipv6 inspect …).

Now run the FTP from the PCs. Verify that the traffic runs through the interfaces you are inspecting. Check the output.

(**Tip**: show ipv6 inspect…)

Summary

After completing these exercises, you should be able to:

- *Configure and apply IPv6 access-lists*
- *Restrict access to VTY*
- *Configure Cisco Inspect Firewall*

Appendix

Step 1: Configure Access-lists

```
Router1# configure terminal
Router1(config)# ipv6 access-list v6test
Router1(config-ipv6-acl)# permit ipv6 host 2001:DB8:CAFE:3::1 any
Router1(config-ipv6-acl)# permit ipv6 host 2001:DB8:CAFE:13::3 any
Router1(config-ipv6-acl)# permit ipv6 host 2001:DB8:CAFE:34::3 any
Router1(config-ipv6-acl)# exit
```

Step 2: Applying Access Lists to your interfaces

In IPv4 the command to apply an access-list to an interface was *ip access-group <access-list_number_or_name> <in|out>*. In IPvt the command is *ipv6 traffic filter*.

```
Router1(config)# interface FastEthernet1
Router1(config-if)# ipv6 traffic-filter v6test in
```

If you notice, you don't peer with any neighbors now. The reason is that the updates are sent via link local addresses. You have to permit these packets to your router.

```
Router1(config)# ipv6 access-list v6test
Router1(config-ipv6-acl)#  permit ipv6 FE80::/16 any
```

Test again. You should be able to ping your direct neighbour.

Step 3: Checking your Access-lists

To explicitly deny any IPv6 ICMP configure:

```
Router1(config)# ipv6 access-list v6test
Router1#(config-ipv6-acl)# deny icmp any any
```

To check the access-list counters do:

```
Router1# show ipv6 access-list v6test
IPv6 access list v6test
permit ipv6 host 2001:DB8:CAFE:3::1 any sequence 10
permit ipv6 host 2001:DB8:CAFE:13::3 any (7 matches) sequence 20
permit ipv6 host 2001:DB8:CAFE:34::3 any sequence 30
permit ipv6 FE00::/8 any (10 matches) sequence 40
deny icmp any any (5 matches) sequence 50
```

If you try to ping from router 3, you get a reply, but none from router 4. This happens because you configured the *allowance of all IPv6 packets* from your neighbour before you denied the ICMP. If you change the order of the lines, you won't get any answer to your pings from any neighbour.

Step 4: Allowing a Port Number and a Protocol

Create a new access-list. Eg:

```
Router1# configure terminal
Router1(config)# ipv6 access-list v6proto
Router1#(config-ipv6-acl)# permit tcp host 2001:DB8:CAFE:3::1
any eq telnet
Router1#(config-ipv6-acl)# permit tcp host 2001:DB8:CAFE:13::3
any eq telnet
Router1#(config-ipv6-acl)# permit tcp host 2001:DB8:CAFE:34::3
any eq telnet
Router1#(config-ipv6-acl)# permit ipv6 FE00::/8 any
```

Apply the access-list to the interface.

You can't ping the routers, because you are only allowing TCP and traffic to Port 23 (telnet). You can also use other ports and commands (gt – greater than, lt – less than, etc.). To permit ICMP:

```
Router1#(config-ipv6-acl)# permit icmp any any
```

Step 5: Controlling access to VTY lines

Create an access-list to the allowed machines/networks:

```
Router1(config)# ipv6 access-list v6lines
Router1#(config-ipv6-acl)# permit ipv6 host 2001:DB8:CAFE:3::1
any
Router1#(config-ipv6-acl)# permit ipv6 host 2001:DB8:CAFE:13::3
any
Router1#(config-ipv6-acl)# permit ipv6 host 2001:DB8:CAFE:34::3
any
Router1#(config-ipv6-acl)# exit
Router1(config)# line vty 0 15
Router1(config-line)# ipv6 access-class v6lines in
```

Test with telnet by different routers.

Step 6: IPv6 Firewall

To configure your router to inspect FTP packets do:

```
Router1(config)# ipv6 inspect name v6ftp-inspect ftp timeout 60
```

To apply this rule to the interface do:

```
Router1(config)# interface FastEthernet 1
Router1(config-if)# ipv6 inspect v6ftp-inspect in
```

Run the FTP between the two PCs. Check the output from the *inspect:* show

```
Router1# show inspect name v6ftp-inspect
```

To activate audit-trail:

```
Router1(config)# ipv6 inspect name v6ftp-inspect ftp audit-trail on
```

To activate alerts:

```
Router1(config)# ipv6 inspect name v6ftp-inspect ftp alert on
```

Some useful commands

Another feature, namely PAM – Port-to-application Mapping, existing in Cisco IOS, allows you to customize TCP or UDP port numbers for network services or applications.

Using the port information, PAM establishes a table of default port-to-application mapping information at the firewall. The information in the PAM table enables Context-based Access Control (CBAC) supported services to run on nonstandard ports.

PAM also supports host or subnet specific port mapping, which allows you to apply PAM to a single host or subnet using standard ACLs. Host or subnet specific port mapping is done using standard ACL.

Eg: create an access-list and then apply it:

```
Router1(config)# ipv6 port-map application-name port port [list acl-name]
```

To check the output from this command, type:

```
Router1# show ipv6 port-map […]
```