

IPv6 Security Lab

APRICOT14 – Manila, Philippines

February 2009

LAB1: Configuring Telnet Vty Access for IPv6

1. Enable IPv6 and configure interface IP addresses

IOS

```
router# config terminal
router(config)# ipv6 unicast-routing
router(config)# interface <interface>
router(config-if)# ipv6 enable
router(config-if)# ipv6 address <ipv6 address>
```

JUNOS

```
lab@router# set interfaces <interface> unit 0 family inet6
address <ipv6 address>
```

2. Check interface status and neighbor caches to see what it looks like.

IOS

```
router# show ipv6 interface
router# show ipv6 neighbors
```

JUNOS

```
lab@Router> show interface
lab@Router> show ipv6 neighbors
```

3. Configure a filter to allow only the trusted hosts to have Telnet access. Note that all tries are logged to have an audit trail of all access to the router.

IOS

```
router(config)# ipv6 access-list v6_telnet-filter
router(config-ipv6-acl)# permit host <ipv6 address>
```

JUNOS

```
lab@Router# set firewall family inet6 filter v6_telnet-  
filter term ALLOW_MY_HOSTS from source-address <ipv6  
address>  
lab@Router# set firewall family inet6 filter v6_telnet-  
filter term ALLOW_MY_HOSTS from next-header tcp  
lab@Router# set firewall family inet6 filter v6_telnet-  
filter term ALLOW_MY_HOSTS from destination-port telnet  
lab@Router# set firewall family inet6 filter v6_telnet-  
filter term ALLOW_MY_HOSTS then accept
```

4. Apply filter to vty ports

IOS

```
router(config)# line vty 0 4  
router(config-line)# ipv6 access-class v6_telnet-filter in
```

JUNOS

```
lab@Router# set interfaces lo0 unit 0 family inet6 filter  
input v6_telnet-filter
```

Test to make sure that only telnet from the configured host can have access to the router. Use the debug command to see if you can capture the telnet packets and see the clear-text passwords.

LAB2: Configuring SSH Vty Access for IPv6

1. Generate the router key with the command: *crypto key generate rsa*
Note that the command is NOT performed in configuration mode.

IOS

```
router# crypto key generate rsa
```

Pick key length of 1024 bytes. Also, note that the domain must also be configured before this command will work.

JUNOS

No equivalent needed

2. Create the filter to allow SSH access. Create a new filter which can be tested and then later the old one can be removed:

IOS

```
router(config)# ipv6 access-list v6_ssh-filter
router(config-ipv6-acl)# permit host <ipv6 address> host
<ipv6 address>
```

JUNOS

```
set firewall family inet6 filter v6_telnet-filter term
ALLOW_MY_HOSTS from destination-port ssh
```

3. Modify vty access command to allow ssh:

IOS

```
router(config)# line vty 0 4
router(config-line)# ipv6 access-class v6_ssh-filter in
router(config-line)# exec-timeout 15 0
router(config-line)# transport input ssh
```

Test to make sure that ssh from the permitted host can get access to the router. Use the debug command to see if you can capture the telnet packets and see the encrypted passwords.

JUNOS

```
lab@Router# set system services ssh
```

LAB3: Configuring IPv6 Traffic Filters

1. Configure simple traffic filter to announce only your subnet (BCP38)

IOS

```
router(config)# ipv6 access-list ipv6-netannounce
router(config-ipv6-acl)# permit ipv6 <v6 subnet> any
router(config-ipv6-acl)# deny ipv6 any any log
router(config-ipv6-acl)# exit
router(config)# interface <interface>
router(config-if)# ipv6 traffic-filter ipv6-netannounce out
```

JUNOS

```
lab@Router# set firewall family inet6 filter ipv6-  
netannounce term MY_SUBNET from source-address 2001::0/30  
lab@Router# set firewall family inet6 filter ipv6-  
netannounce term MY_SUBNET then accept  
lab@Router# set firewall family inet6 filter ipv6-  
netannounce term DENY_ALL then log  
lab@Router# set firewall family inet6 filter ipv6-  
netannounce term DENY_ALL then discard  
lab@Router# set interfaces ge-0/0/1.0 family inet6 filter  
output ipv6-netannounce
```

2. Filter RH Type0 packets

IOS

Note: need 12.4(2) T or higher for the following configuration to work.

```
router(config)# ipv6 access-list deny-sourcerouted  
router(config-ipv6-acl)# deny ipv6 any any routing-type 0  
router(config-ipv6-acl)# permit ipv6 any any  
router(config-ipv6-acl)# exit  
router(config)# interface Ethernet0  
router(config-if)# ipv6 source-route  
router(config-if)# ipv6 traffic-filter deny-sourcerouted in
```

For versions prior to 12.4(2) T, the following configuration will work but it filters all RH Type packets

```
router(config)# no ipv6 source route
```

JUNOS

```
set firewall family inet6 filter DENY_SOURCEROUTED term  
DENY from next-header hop-by-hop  
set firewall family inet6 filter DENY_SOURCEROUTED term  
DENY from next-header routing  
set firewall family inet6 filter DENY_SOURCEROUTED term  
DENY then discard  
set firewall family inet6 filter DENY_SOURCEROUTED term  
PERMIT_ALL then accept  
set interfaces <interface> unit 0 family inet6 filter  
input DENY_SOURCEROUTED
```

Default behavior is to disable source-routing

2. Inbound packet filter for initial ipv6 testing

IOS

```
router(config)# ipv6 access-list v6starter
router(config-ipv6-acl)# permit icmp any any <ipv6 subnet>
                        echo-reply log-input
router(config-ipv6-acl)# permit icmp any any <ipv6 subnet>
                        echo-request log-input
router(config-ipv6-acl)# permit icmp any any <ipv6 subnet>
                        time-exceeded log-input
router(config-ipv6-acl)# permit icmp any any <ipv6 subnet>
                        packet-too-big log-input
router(config-ipv6-acl)# permit icmp any any <ipv6 subnet>
                        parameter-problem log-input
router(config-ipv6-acl)# permit ipv6 any host
                        <specific host> log-input
router(config-ipv6-acl)# deny ipv6 any any log-input
router(config-ipv6-acl)# exit
router(config)# interface <interface>
router(config-int)# ipv6 traffic-filter v6starter in
```

Note that the 'log-input' is more to check what ipv6 traffic is coming in from the outside. Send some ipv6 pings and see if can see traffic from a 'show log'.

JUNOS

```
set firewall family inet6 filter IPV6_STARTER term ICMP
from icmp-type echo-reply
set firewall family inet6 filter IPV6_STARTER term ICMP
from icmp-type parameter-problem
set firewall family inet6 filter IPV6_STARTER term ICMP
from icmp-type echo-request
set firewall family inet6 filter IPV6_STARTER term ICMP
from icmp-type packet-too-big
set firewall family inet6 filter IPV6_STARTER term ICMP
from icmp-type time-exceeded
set firewall family inet6 filter IPV6_STARTER term ICMP
then log
set firewall family inet6 filter IPV6_STARTER term ICMP
then accept
set firewall family inet6 filter IPV6_STARTER term DENY_ALL
then log
set firewall family inet6 filter IPV6_STARTER term DENY_ALL
```

```
then discard
set interfaces <interface> unit 0 family inet6 filter
input IPV6_STARTER
```