LAB: route filtering (cisco routers)

Lab Environment:

The workshop wifi:

SSID: workshop

PASS: iij/2497

Routers - Cisco routers (Dynamips):

Hostname: RXX.local

IPv6: fd00:2497:1::10:XX

IPv4: 10.0.10.XX

For telnet (vty) access:

username: workshop

password: iij/2497

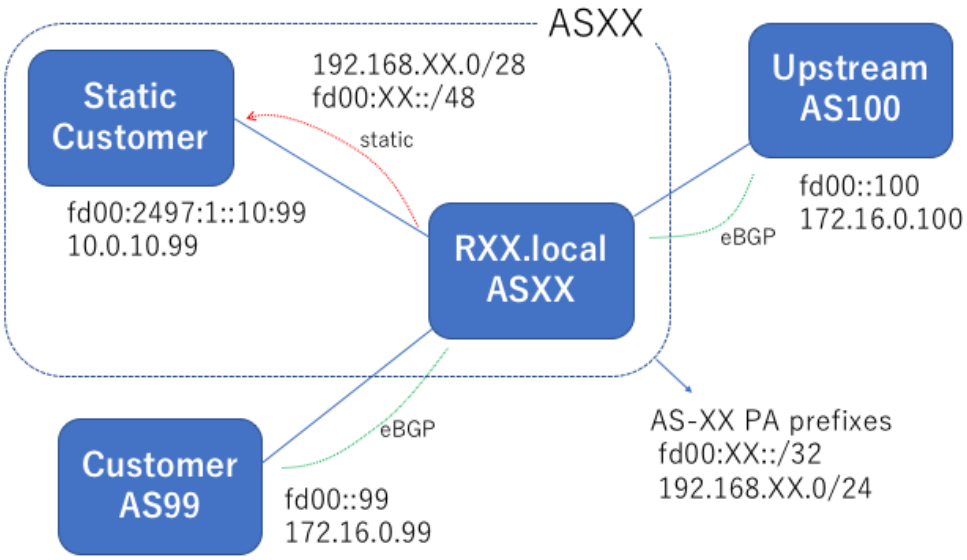
Number Resources

AS#: XX

IPv6 PA: fd00:XX::/32

IPv4 PA: 192.168.XX.0/24

Note: XX is your group id



Configure static routes for your static customer

telnet to your RXX router. If you don't have a telnet client, ssh to nsXX host first and then telnet to your router.

$ telnet RXX.local

Enter the configuration mode by typing 'configure terminal', and configure static route for your static customer. That should be fd00:XX::/48 to fd00:2497:1::10:99 and 192.168.XX.0/28 to 10.0.10.99.

RXX# configure terminal

RXX(config)# ipv6 route fd00:XX::/48 Ether0/0 fd00:2497:1::10:99

RXX(config)# ip route 192.168.XX.0 255.255.255.240 Ether0/0 10.0.10.99

RXX(config)# end

Check your routing and reachability.

RXX# show ipv6 route

RXX# show ip route

RXX# ping fd00:XX::1

RXX# ping 192.168.XX.1

Note: XX is your group id

Originate your own PA prefixes

1. configure static null routes to have stable routing of your PA prefixes
2. configure bgp process, and originate BGP routes

Enter the configuration mode by typing 'configure terminal', and configure static null route of your PA blocks

RXX# configure terminal

RXX(config)# ipv6 route fd00:XX::/32 null 0

RXX(config)# ip route 192.168.XX.0 255.255.255.0 null 0

RXX(config)# end

Check the static routes

RXX# show ipv6 route

RXX# show ip route

Enter the configuration mode, and configure BGP process to originate BGP routes from the router

RXX(config)# router bgp XX
RXX(config-router)# no bgp default ipv4-unicast
RXX(config-router)# address-family ipv6
RXX(config-router-af)# network fd00:XX::/32
RXX(config-router-af)# address-family ipv4
RXX(config-router-af)# network 192.168.XX.0 mask 255.255.255.0
RXX(config)# end

Check your BGP routes

RXX# show bgp ipv6 unicast route
RXX# show ip bgp

prefix-based outbound filtering

First configure prefix-based filtering rules to announce your PA blocks to the upstream

RXX(config)# ipv6 prefix-list OUT6 permit fd00:XX::/32
RXX(config)# ip prefix-list OUT4 permit 192.168.XX.0/24

Configure eBGP session with upstream

RXX(config)# router bgp XX
RXX(config-router)# neighbor fd00::100 remote-as 100
RXX(config-router)# neighbor 172.16.0.100 remote-as 100
RXX(config-router)# address-family ipv6
RXX(config-router-af)# neighbor fd00::100 activate
RXX(config-router-af)# neighbor fd00::100 prefix-list OUT6 out
RXX(config-router-af)# address-family ipv4
RXX(config-router-af)# neighbor 172.16.0.100 activate
RXX(config-router-af)# neighbor 172.16.0.100 prefix-list OUT4 out
RXX(config-router-af)# end

Check your BGP routes

RXX# show bgp ipv6 unicast

RXX# show ip bgp

Configure eBGP session with downstream

The customer, AS-99 has own PA blocks 192.168.99.0/24 and fd00:99::/32, and you need to provide transit for them. In other words, you need to announce the prefixes to upstream. So now you update your prefix-filters, OUT4 and OUT6 to allow BGP customer's prefixes.

RXX(config)# ipv6 prefix-list OUT6 permit fd00:99::/32

RXX(config)# ip prefix-list OUT4 permit 192.168.99.0/24

RXX(config)# end

Check if you configure it properly

RXX# show ipv6 prefix-list OUT6

RXX# show ip prefix-list OUT4

Now configure eBGP session

RXX(config)# router bgp XX

RXX(config-router)# neighbor fd00::99 remote-as 99

RXX(config-router)# neighbor 172.16.0.99 remote-as 99

RXX(config-router)# address-family ipv6

RXX(config-router-af)# neighbor fd00::99 activate

RXX(config-router-af)# address-family ipv4

RXX(config-router-af)# neighbor 172.16.0.99 activate

RXX(config-router-af)# end

Check your BGP routes

RXX# show bgp ipv6 unicast

RXX# show ip bgp

Check your BGP announcement at your upsteam router at fd00:2497:1::10:100

Exercise 1:

Probably your customer is announcing unnecessary de-aggregated prefixes to you. It's better to have an inbound filtering to avoid routing troubles.

RXX(config)# ipv6 prefix-list AS99-IPv6 permit fd00:99::/32
RXX(config)# ip prefix-list AS99-IPv4 permit 192.168.99.0/24
RXX(config)# router bgp XX
RXX(config-router)# address-family ipv6
RXX(config-router-af)# neighbor fd00::99 prefix-list AS99-IPv6 in
RXX(config-router-af)# address-family ipv4
RXX(config-router-af)# neighbor 172.16.0.99 prefix-list AS99-IPv4 in
RXX(config-router-af)# end

Now you need to soft reset the BGP session as BGP filtering policy is applied only when the router receives BGP UPDATE from its neighbor.

RXX# clear bgp ipv6 unicast fd00::99 soft in
RXX# clear bgp ipv4 unicast 172.16.0.99 soft in

Check your BGP routes

RXX# show bgp ipv6 unicast
RXX# show ip bgp

Exercise 2:

Configure uRPF check on your static customer facing interface - Ether0/0

ipv6 verify unicast source reachable-via rx
ip verify unicast source reachable-via rx

Note: you need to enable CEF on your router to have uRPF on cisco routers

Exercise 3:

Consider inbound filtering for your upstream.

You shouldn't accept any sub-prefixes of your IP blocks from your eBGP neighbors.

ip prefix-list PROTECT4 deny 192.168.XX.0/24 ge 25
ip prefix-list PROTECT4 permit 0.0.0.0/0 le 24
ipv6 prefix-list PROTECT6 deny fd00:XX::/32 ge 33
ipv6 prefix-list PROTECT6 permit 0::0/0 le 48

Exercise 4:

Configure VTY access-list to allow access only from your host

access-list 10 permit host 10.0.0.XX
access-list 10 deny any log
ipv6 access-list vty
permit ipv6 host FD00:2497:1::XX any
deny ipv6 any any log
line vty 0 4
access-class 10 in
ipv6 access-class vty in

Exercise 5:

Consider additional route filtering based on BGP community.
Note: Tag your BGP routes by BGP community. Based on that you can pick prefixes that are needed to announce to your eBGP neighbors. At the same time, you need to consider to overwrite some BGP communities from your eBGP neighbors to protect your routing policy.

ip bgp-community new-format
ip community-list 100 deny XX:10
*ip community-list 100 permit :**
ip community-list standard TRANSIT permit XX:10

```
route-map TRANSIT permit 10
 match community TRANSIT
!
route-map TRANSIT deny 20
!
route-map TAG permit 10
 set local-preference 110
 set community XX:10 additive

route-map PEER permit 10
 set comm-list 100 delete

router bgp XX
address-family ipv4
 network 192.168.XX.0 route-map TAG
 neighbor 192.168.0.99 route-map TAG in
 neighbor 192.168.0.100 route-map PEER in
 neighbor 192.168.0.100 route-map TRANSIT out

address-family ipv6
 network fd00:XX::/32 route-map TAG
 neighbor fd00::99 route-map TAG in
 neighbor fd00::100 route-map PEER in
 neighbor fd00::100 route-map TRANSIT out
```