Module 7 RPKI Lab

Objective: The focus of this lab is to configure 8 routers to do the RPKI validation. Participants will do necessary interface configuration for BGP and RPKI related configuration.

Prerequisites: Intermediate routing concept (BGP), Cisco router CLI, Telnet/SSH software etc.

The following will be the topology and IP address plan used for the labs.

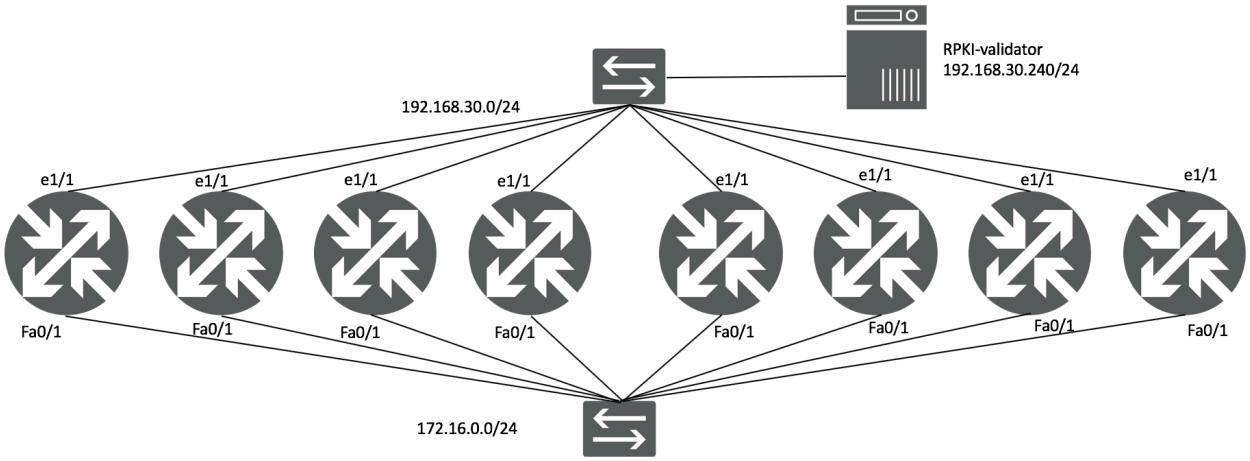


Figure 1 – RPKI Lab Topology

Lab Notes

This workshop is intended to be run on a real cisco routers or Dynamips server with the above lab topologies set up. The routers are using both IPv4 and IPv6 supported IOS software, the version of IOS is 15.2(4)S3. There is one RPKI-validator has been installed by the instructors.

In the topology above, there are 8 routers (*R13 – R20*), each of them represents one ISP, there are in different ASes (*AS135533 – AS135540*). These 8 routers will peer to each other via eBGP, they will learn the routes from each other. All of them are connected to the RPKI validator (*whose IP address is 192.168.30.240*)too, the routers will establish the session with validator. There are ROA already configured on MyAPNIC related to the AS numbers (*AS135533 – AS135540*). Then we can test on routers how RPKI validator works to do validation for the path received from BGP peers.

Table-1 Address Planning

Router	**AS Number**	f0/1 Connected with BGP peers	e1/1 Connected with RPKI-validator	**Legal Prefix on ROA**
R13	AS135533	172.16.0.13/24	192.168.30.13/24	61.45.248.0/24
R14	AS135534	172.16.0.14/24	192.168.30.14/24	61.45.249.0/24
R15	AS135535	172.16.0.15/24	192.168.30.15/24	61.45.250.0/24

R16	AS135536	172.16.0.16/24	192.168.30.16/24	61.45.251.0/24
R17	AS135537	172.16.0.17/24	192.168.30.17/24	61.45.252.0/24
R18	AS135538	172.16.0.18/24	192.168.30.18/24	61.45.253.0/24
R19	AS135539	172.16.0.19/24	192.168.30.19/24	61.45.254.0/24
R20	AS135540	172.16.0.20/24	192.168.30.20/24	61.45.255.0/24

In the class, each group will configure one router to achieve following tasks:

1. All the routers set up eBGP with each other and announce prefixes
2. All the routers establish TCP session with RPKI-validator
3. Check the results of routes with legal origin AS and illegal origin AS based on ROA

RPKI Exercise Steps

Prepared by instructor:

RPKI Validator Installation:

RIPE NCC Validator (https://www.ripe.net/manage-ips-and-asns/resource-management/certification/tools-and-resources)

The ip address of the validator used in this lab is 192.168.30.240.

Participants configuration:
Router configuration

1. Interface configuration
2. BGP peering configuration
3. BGP route-map configuration
4. BGP RPKI server configuration
5. BGP prefix announcement
6. Verify the routes

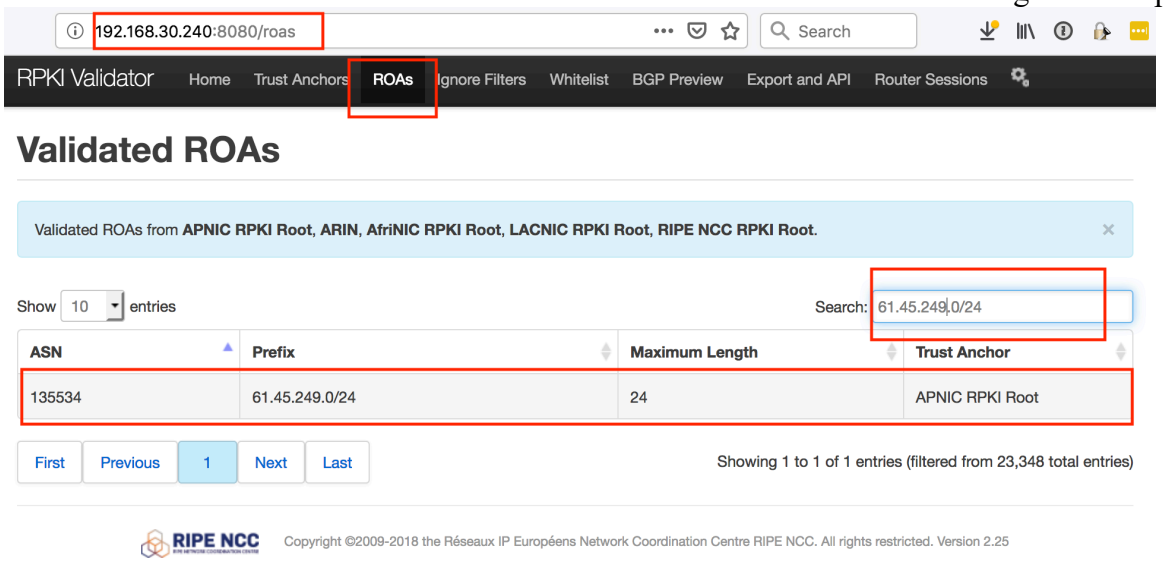
Lab Exercise

1. **Check the RPKI validator reachability from your laptop:**

   The RPKI validator has already been installed, please open this page in browser:
   192.168.30.240:8080
   Be familiar with the contents on the page.

2. **Check ROA results from the validator, you can also check these from whois database.**

   Example of checking the address block 61.45.249.0/24 which should be announced from AS135534 Router14 in this lab. The information will match Table-1.



3. Log into your router through telnet, the telnet port number of each router is as following. Windows OS users please use Putty or SecuCRT to telnet via port.

Router	Telnet with Port Number
R13	telnet 192.168.30.240 2013
R14	telnet 192.168.30.240 2014
R15	telnet 192.168.30.240 2015
R16	telnet 192.168.30.240 2016
R17	telnet 192.168.30.240 2017
R18	telnet 192.168.30.240 2018
R19	telnet 192.168.30.240 2019
R20	telnet 192.168.30.240 2020

4. Configure interfaces

4.1 Configure interface connected to RPKI validator

Example configuration on R13

```
enable
config t
hostname R13
interface Ethernet1/1
description To RPKI-validator
no ip redirects
no ip unreachables
no clns route-cache
ip address 192.168.30.13 255.255.255.0
duplex full
no shutdown
end
wr
```

Interface connectivity with RPKI-validator verification:

```
ping 192.168.30.240
```

4.2 Configure interface for eBGP peer

Example configuration on R13

```
config t
interface fa0/1
description To eBGP Peers
no ip redirects
no ip unreachables
no clns route-cache
ip address 172.16.0.13 255.255.255.0
duplex full
no shutdown
end
wr
```

Interface connectivity with other routers:

```
ping 172.16.0.14
```

And ping other routers IP addresses (172.16.0.13 ~ 172.16.0.20)

5. Configure eBGP peers between the 8 routers

Each router will set up an eBGP neighbour with a neighbor, we design like following 4 pairs:

R13 ⇔ R14
R15 ⇔ R16
R17 ⇔ R18
R19 ⇔ R20

Example configuration on R13, R13 will form neighbour relationship with R14, similar configuration on other routers:

```
config t
router bgp 135533
address-family ipv4
neighbor 172.16.0.14 remote-as 135534
neighbor 172.16.0.14 activate
end
wr
```

Example configuration on R16, R16 will form neighbour relationship with R15:

```
config t
router bgp 135536
address-family ipv4
neighbor 172.16.0.15 remote-as 135535
```

```
neighbor 172.16.0.15 activate
end
wr
```

Check the eBGP neighbor relationship on the router by using following commands:

```
show bgp ipv4 unicast summary
```

Each router should have 1 eBGP neighbour established.

6. Configure BGP route-map related to RPKI and apply to eBGP neighbors inbound

Example configuration on R13

```
config t
route-map rpki-loc-pref permit 10
match rpki invalid
set local-preference 90
route-map rpki-loc-pref permit 20
match rpki not-found
set local-preference 100
route-map rpki-loc-pref permit 30
match rpki valid
set local-preference 110
!
router bgp 135533
address-family ipv4
neighbor 172.16.0.14 route-map rpki-loc-pref in
end
wr
```

After we have applied the route-map for RPKI, run following command to soft reconfig BGP peer inbound.

```
clear bgp ipv4 unicast * in
```

7. Configure BGP to validate prefixes based on origin AS

Configure the RPKI-validator server information on routers.

Example configuration on R13:

```
conf t
router bgp 135533
address-family ipv4
```

```
bgp rpki server tcp 192.168.30.240 port 8282 refresh 600
end
wr
```

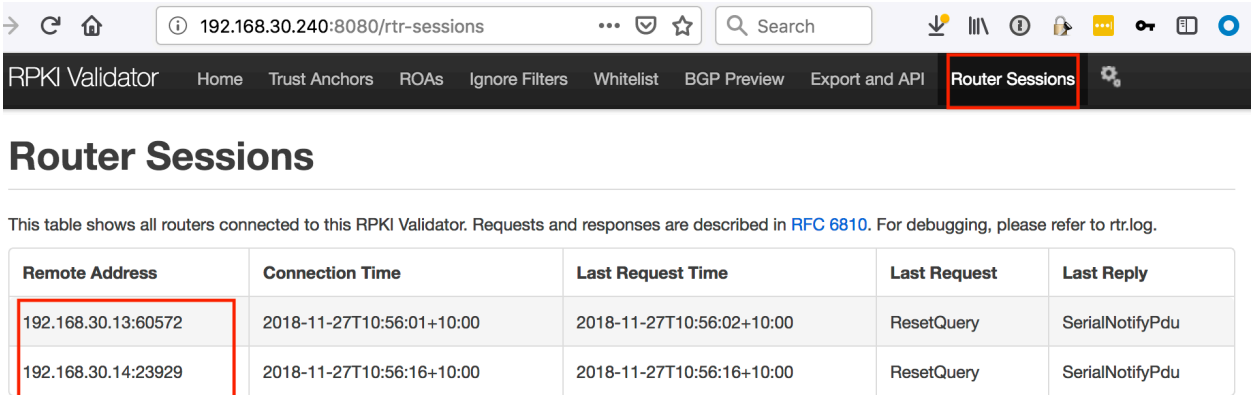
Check the RPKI server information on router:

```
show ip bgp rpki server
show ip bgp rpki table
```

You should be able to see some information as following on the router:

```
R13#show ip bgp rpki server
BGP SOVC neighbor is 192.168.30.240/8282 connected to port 8282
Flags 64, Refresh time is 600, Serial number is 21, Session ID is 17969
InQ has 0 messages, OutQ has 0 messages, formatted msg 1
Session IO flags 3, Session flags 4008
 Neighbor Statistics:
  Prefixes 54261
  Connection attempts: 1
  Connection failures: 0
  Errors sent: 0
  Errors received: 0
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
……
```

We can also see similar information from the RPKI-validator portal on page "Router Sessions":



8. Announce prefix from each AS

In this step, each router will announce prefixes, we will check the routes with different states: "valid", "invalid", "Not-found"

8.1 Valid state ----- Announce legal prefix

Each router announces their legal prefixes based on ROA, the legal prefixes refer to **Table-1** on Page 1-2 in this lab guide.

```
conf t
ip route 61.45.248.0 255.255.255.0 null 0
router bgp 135533
address-family ipv4 unicast
network 61.45.248.0 mask 255.255.255.0
end
wr
```

Check the routes on router:

`sh bgp ipv4 unicast neighbors [router 13……router20] routes` [To check prefixes learn from BGP peers]
`sh ip route [R13, R14, R15, R16, R17, R18, R19, R20]` [To check prefixes in routing table]

Here is the result on R15:

```
R15#show bgp ipv4 unicast
BGP table version is 29, local router ID is 192.168.30.15
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
V*>  61.45.250.0/24   0.0.0.0                  0         32768 i
V*>  61.45.251.0/24   172.16.0.16              0    110      0 135536 i
R15#
```

8.2 Invalid state --- Announce prefixes with invalid origination AS which are other routers' prefixes

For example, if AS135533 announce AS135536 prefix (61.45.251.0/24)

```
conf t
ip route 61.45.251.0 255.255.255.0 null 0
router bgp 135533
address-family ipv4 unicast
network 61.45.251.0 mask 255.255.255.0
end
wr
```

Each group's router announce your neighbour group's prefix.

Check the routes on router:

`sh bgp ipv4 unicast neighbors [router 13……router20] routes` [To check prefixes learn from iBGP peers]
`sh ip route [R13, R14, R15, R16, R17, R18, R19, R20]` [To check prefixes in routing table]

Result on R14, when it has received an illegal route from R13:

```
R14#show bgp ipv4 unicast
BGP table version is 6, local router ID is 192.168.30.14
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
V*>  61.45.248.0/24   172.16.0.13              0    110      0 135533 i
V*>  61.45.249.0/24   0.0.0.0                  0         32768 i
I*   61.45.251.0/24   172.16.0.13              0     90      0 135533 i
```

8.3 Announce private addresses which are not in the ROA database

For example, if AS135533 announce the private addresses.

```
conf t
ip route 10.13.0.0 255.255.255.0 null 0
router bgp 135533
address-family ipv4 unicast
network 10.13.0.0 mask 255.255.255.0
end
wr
```

Check the routes on router:

`sh bgp ipv4 unicast neighbors [router 13……router20] routes` [To check prefixes learn from iBGP peers]
`sh ip route [R13, R14, R15, R16, R17, R18, R19, R20]` [To check prefixes in routing table]

Example result on R14:

```
R14#show bgp ipv4 unicast
BGP table version is 7, local router ID is 192.168.30.14
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
N*>  10.13.0.0/24     172.16.0.13              0    100      0 135533 i
V*>  61.45.248.0/24   172.16.0.13              0    110      0 135533 i
V*>  61.45.249.0/24   0.0.0.0                  0         32768 i
I*   61.45.251.0/24   172.16.0.13              0     90      0 135533 i
```

9. Additional information: (Optional) (refer to https://www.ripe.net/manage-ips-and-asns/resource-management/certification/router-configuration)

Related to IOS version what we are using in the lab, there are two ways you can modify the default BGP best path selection process when using RPKI validation states:

- You can completely disable the validation of prefixes on the router. This is done by configuring the **bgp bestpath prefix-validate disable** command. You might want to do this for configuration testing. The router will still connect to the RPKI Validator and download the validation information but will not use the information.
- You can allow an invalid prefix to be used as the BGP best path, even if valid prefixes are available. This is the default behaviour. The command to allow a BGP best path to be an invalid prefix, as determined by the BGP Origin AS Validation feature, is the **bgp bestpath prefix-validate allow-invalid** command. The prefix validation state will still be assigned to paths and will still be communicated to iBGP neighbours that have been configured to receive RPKI state information. You can use a route map to set a local preference, metric or other property based on the validation state.

During BGP best path selection, the default behaviour, if neither of the above options is configured, is that the system will prefer prefixes in the following order:

- Those with a validation state of valid.
- Those with a validation state of not found.
- Those with a validation state of invalid **(which, by default, will not be installed in the routing table)**.

These preferences override metric, local preference and other choices made during the bestpath computation. The standard bestpath decision tree applies only if the validation state of the two paths is the same.

If both commands are configured, the **bgp bestpath prefix-validate disable** command will prevent the validation state from being assigned to paths, so the **bgp bestpath prefix-validate allow-invalid** command will have no effect.

These configurations can be in either router configuration mode or in address family configuration mode for the IPv4 unicast or IPv6 unicast address families.