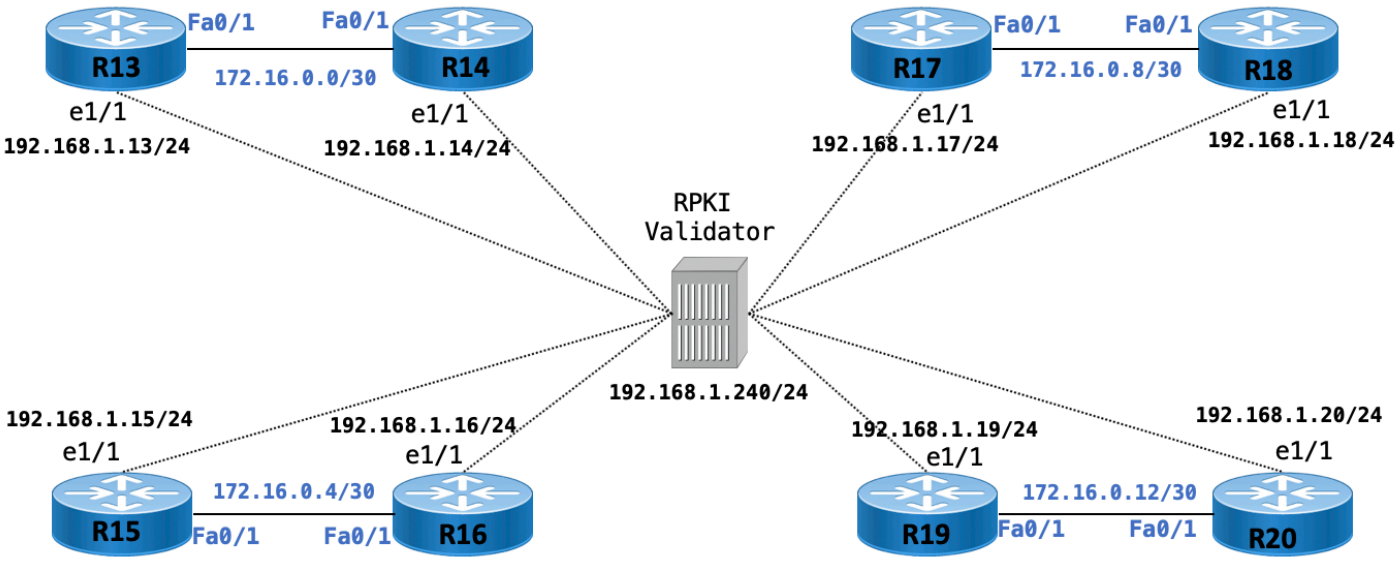
LAB: RPKI

Lab Setup

Topology

The topology below has 8 routers (`R13, R14, ...R20`), each with a unique ASN (`AS135533 - AS135540`).



Address plan & ROA table

Router	AS Number	f0/1 (to eBGP peers)	e1/1 (to RPKI-validator)	Valid ROA
R13	AS135533	172.16.0.1/30	192.168.30.13/24	61.45.248.0/24
R14	AS135534	172.16.0.2/30	192.168.30.14/24	61.45.249.0/24
R15	AS135535	172.16.0.5/30	192.168.30.15/24	61.45.250.0/24
R16	AS135536	172.16.0.6/30	192.168.30.16/24	61.45.251.0/24
R17	AS135537	172.16.0.9/30	192.168.30.17/24	61.45.252.0/24
R18	AS135538	172.16.0.10/30	192.168.30.18/24	61.45.253.0/24
R19	AS135539	172.16.0.13/30	192.168.30.19/24	61.45.254.0/24
R20	AS135540	172.16.0.14/30	192.168.30.20/24	61.45.255.0/24

Lab Notes

- For this lab, we will use RIPE's RPKI validator, which has already been installed and configured by the instructor as shown in the topology. The validator's IP address is `192.168.1.240`
- To simplify the configuration, the routers will establish eBGP session in pairs as shown below:

```
R13<-->R14
R15<-->R16
R17<-->R18
R19<-->R20
```

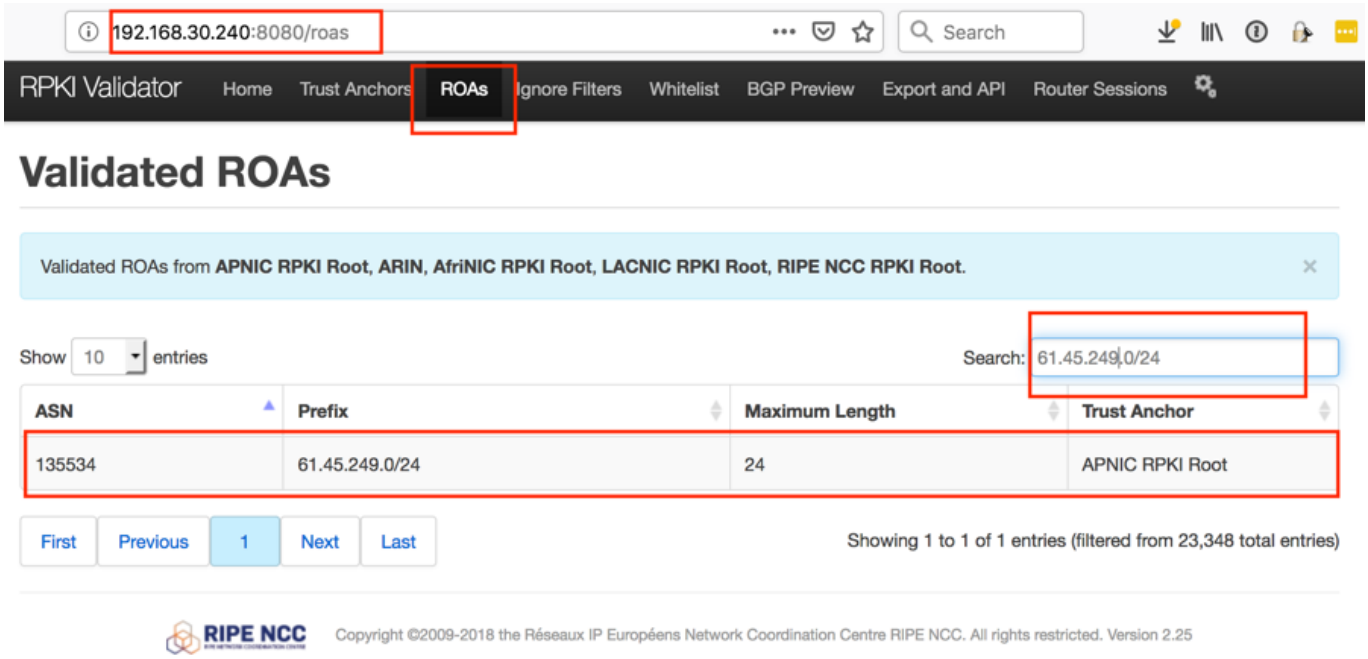
- Each router also has a connection to the RPKI validator to allow RTR (**rpki-to-router**) sessions.
- ROAs have already been created for each of the prefixes with corresponding origin AS numbers (`AS135533 - AS135540`) as shown in the table above.

Lab Exercise

1. On your browser, open the RPKI Validator dashboard

```
192.168.1.240:8080
```

2. Get yourself familiar with the different tabs on the dashboard. For example, you can view all the valid (existing) ROAs pulled by the validator from the global RPKI repository, as shown below:



3. Login to your group's router using telnet as shown below:

Router	Command		
R13	telnet	192.168.1.254	2013
R14	telnet	192.168.1.254	2014
R15	telnet	192.168.1.254	2015
R16	telnet	192.168.1.254	2016
R17	telnet	192.168.1.254	2017
R18	telnet	192.168.1.254	2018
R19	telnet	192.168.1.254	2019
R20	telnet	192.168.1.254	2020

4. Configure the host name and the interface to the validator (example for R13 below). Refer the address plan table:

```
hostname R13
interface Ethernet1/1
 description link to RPKI-validator
 ip address 192.168.30.13 255.255.255.0
 duplex full
 no shutdown
```

5. Verify connectivity between the router and the validator

```
ping 192.168.1.240
```

6. Configure the interface connecting to your eBGP peer (example for R13 below). Refer the address plan table:

```
interface fa0/1
 description link to R14
 ip address 172.16.0.1 255.255.255.252
 duplex full
 no shutdown
```

7. Verify connectivity to your peer (talk to your neighbor):

```
ping 172.16.0.2
```

8. Configure eBGP with your neighbor (make sure its the correct neighbor). Example below for R13's

eBGP with R14:

```
router bgp 135533
 address-family ipv4
  neighbor 172.16.0.2 remote-as 135534
  neighbor 172.16.0.2 activate
```

9. Make sure the eBGP session is up with your neighbor

```
show bgp ipv4 unicast summary
```

10. Announce the correct prefix (based on the address plan table above) to your neighbor. Example below is for R13:

```
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
```

11. Check/Verify routes learned from your neighbor. Example, for R13 to check what it recived from its neighbor R14:

```
sh bgp ipv4 unicast neighbors 172.16.0.2
```

12. Look at the BGP table:

```
sh bgp ipv4 unicast
```

13. Look at the routing table

```
sh ip route
```

14. Setup RTR (rpki-to-router) session with the RPKI validator. Example for R13:

```
router bgp 135533
 address-family ipv4
  bgp rpki server tcp 192.168.1.240 port 8282 refresh 600
```

15. Verify the RTR session with the validator

```
show ip bgp rpki server
```

The output should look like something below:

```
BGP SOVC neighbor is 192.168.1.240/8282 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
……
```

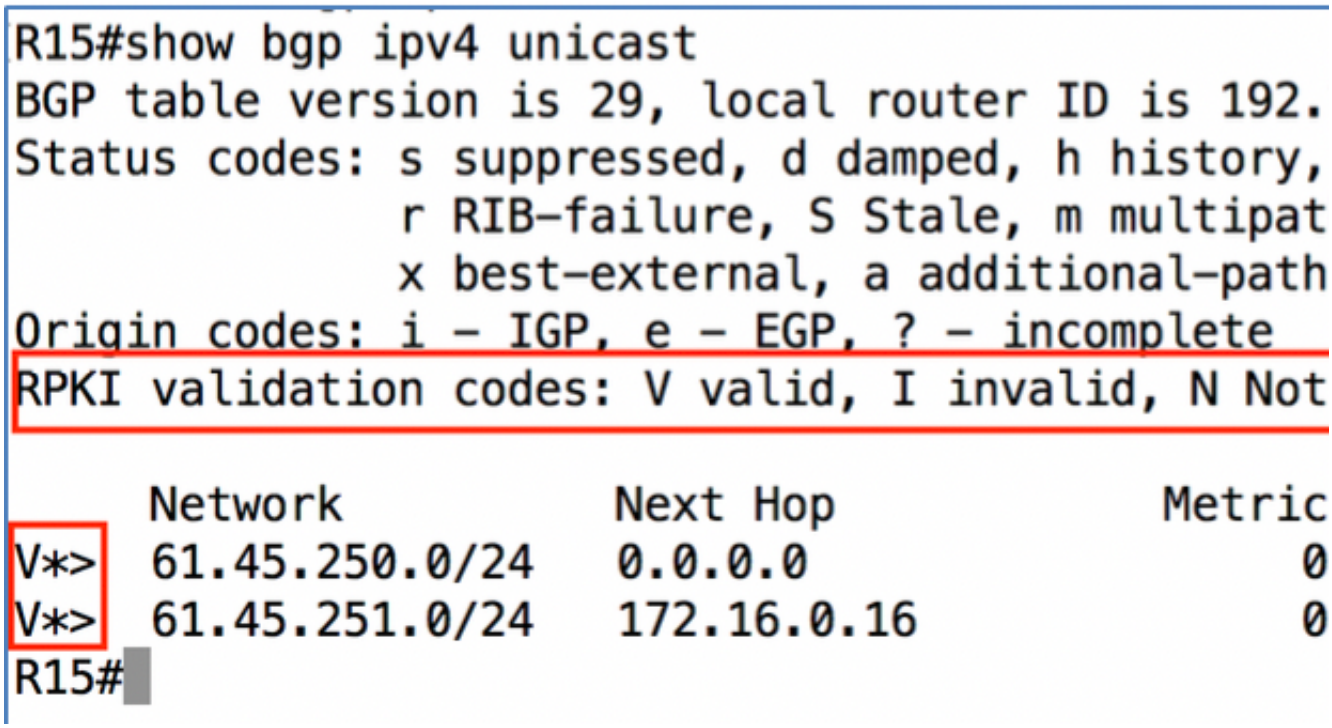
16. Look at all the valid ROAs learned from the validator

```
show bgp ipv4 unicast rpki table
```

17. Now check the BGP table again to see how the route's you learned from your neighbor is tagged with the RPKI validation state:

```
show bgp ipv4 unicast
```

** Since the prefixes have their corresponding ROAs already created, you should see all of them as valid (V). Example below for R15:



End of Lab