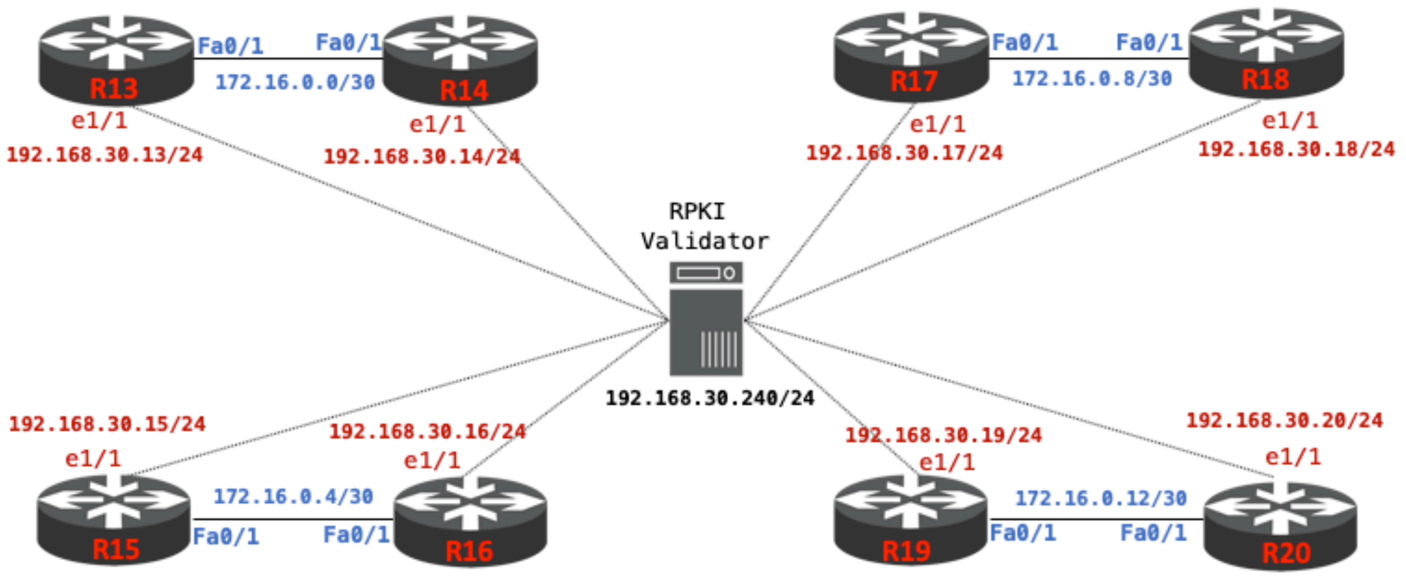
Route Origin Validation Lab

Part-3: Configure Routers to filter against ROAs

Topology

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



Address plan & ROA table

Router	AS#	fa0/1 (to eBGP peers)	e1/1 (to Validator)	ROA/route
R13	135533	172.16.0.1/30	192.168.30.13/24	61.45.248.0/24
R14	135534	172.16.0.2/30	192.168.30.14/24	61.45.249.0/24
R15	135535	172.16.0.5/30	192.168.30.15/24	61.45.250.0/24
R16	135536	172.16.0.6/30	192.168.30.16/24	61.45.251.0/24
R17	135537	172.16.0.9/30	192.168.30.17/24	61.45.252.0/24
R18	135538	172.16.0.10/30	192.168.30.18/24	61.45.253.0/24
R19	135539	172.16.0.13/30	192.168.30.19/24	61.45.254.0/24
R20	135540	172.16.0.14/30	192.168.30.20/24	61.45.255.0/24

Lab Notes

- For this lab, the RPKI Validator (we are running Routinator) has been installed and configured by the instructor as shown in the topology. The validator's IP address is `192.168.30.240` and is listening on port: `3323`
- 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. Telnet (from the jumphost) to your assigned router as shown below:

```
telnet 192.168.30.254 2013 [R13]
telnet 192.168.30.254 2014 [R14]
telnet 192.168.30.254 2015 [R15]
telnet 192.168.30.254 2016 [R16]
telnet 192.168.30.254 2017 [R17]
telnet 192.168.30.254 2018 [R18]
telnet 192.168.30.254 2019 [R19]
telnet 192.168.30.254 2020 [R20]
```

2. If you see the following message during router bootup, enter `no` :

```
Would you like to enter the initial configuration dialog? [yes/no]:
```

3. You also might see the following service configuration messages when the IOS boots:

```
%Error opening tftp://192.168.30.254/network-confg (Timed out)
%Error opening tftp://192.168.30.254/cisconet.cfg (Timed out)
%Error opening tftp://192.168.30.254/router-confg (Timed out)
%Error opening tftp://192.168.30.254/ciscortr.cfg (Timed out)
```

   - Please disable this inbuilt feature and save the config to prevent it from happening during the next boot up:

     ```
     no service config
     do wr
     ```

   NOTE: Since we are running the lab on dynamips, if you need to reload your router, DO NOT issue the reload command (please ask your instructor)!

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

```
hostname R13
no logging console
!
interface ethernet1/1
 description link to RPKI-Validator
 ip address 192.168.30.13 255.255.255.0
 no shutdown
```

5. Verify connectivity between the router and the Validator

```
ping 192.168.30.240
```

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
 no shutdown
```

7. Verify connectivity to your eBGP peer (talk to your neighbor if there is no reachability). Example for **R19** to check its physical connection to **R20**:

```
ping 172.16.0.14
```

8. Configure eBGP with your neighbor (make sure its the correct neighbor). Example below for **R13**'s eBGP session with **R14**:

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

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

```
sh bgp ipv4 unicast summary
```

   - **Note:** You will not see any prefixes received from your neighbor yet.

10. Announce the correct prefix (based on the address plan table above) to your neighbor. Example below is for **R15**:

```
ip route 61.45.250.0 255.255.255.0 null 0
!
router bgp 135535
 address-family ipv4 unicast
 network 61.45.250.0 mask 255.255.255.0
```

11. Check/Verify routes learned from your neighbor. Example, for **R19** to verify received routes from its neighbor **R20**:

    ```
    sh bgp ipv4 unicast neighbors 172.16.0.14 routes
    ```

12. Verify the BGP table:

    ```
    sh bgp ipv4 unicast
    ```

13. Verify the routing table for BGP learned routes

    ```
    sh ip route bgp
    ```

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

    ```
    router bgp 135533
      bgp rpki server tcp 192.168.30.240 port 3323 refresh 600
    ```

    NOTE: *Since the router will now pull the validated ROA cache using the RTR protocol from the Validator, it might take a while.*

    - The `refresh` rate specifies how often the router (RTR client) will query the RTR server

15. Verify the RTR session with the Validator

    ```
    sh ip bgp rpki servers
    ```

    OR

    ```
    sh bgp ipv4 unicast rpki servers
    ```

    - The output should look like something below:

      ```
      BGP SOVC neighbor is 192.168.30.240/3323 connected to port 3323
      Flags 192, Refresh time is 900, Serial number is 0, Session ID is 15578
      InQ has 0 messages, OutQ has 0 messages, formatted msg 1
      Session IO flags 3, Session flags 4000
      Neighbor Statistics:
          Prefixes 39736
          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

```
sh bgp ipv4 unicast rpki table
```

- Should output a list of ROAs (origin-AS, max-length) like below:

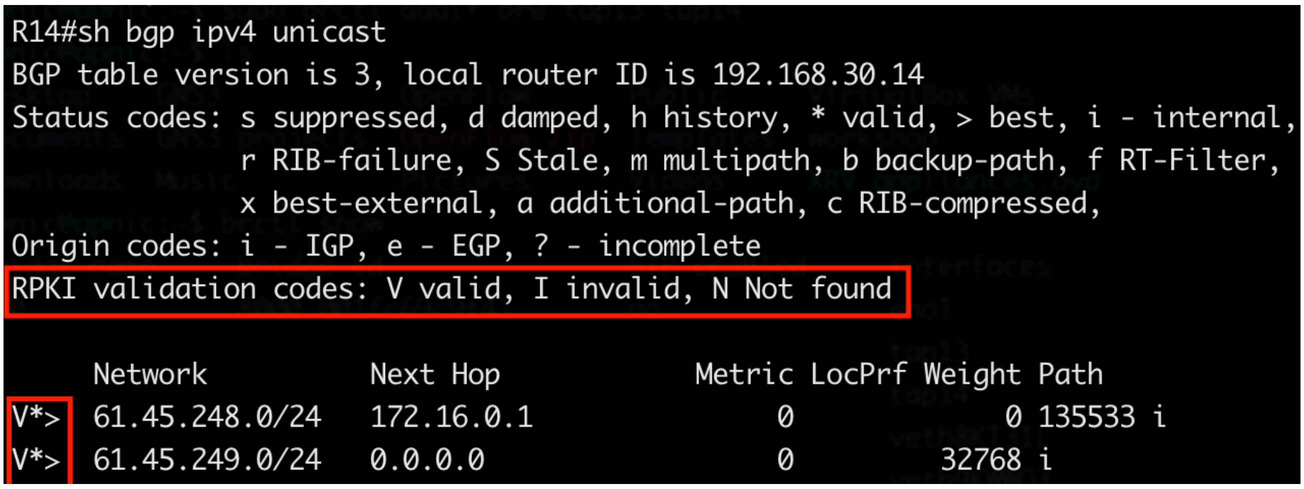
```
65373 BGP sovc network entries using 5752824 bytes of memory
69579 BGP sovc record entries using 1391580 bytes of memory

Network               Maxlen  Origin-AS  Source  Neighbor
1.0.0.0/24            24      13335      0       192.168.30.240/3323
1.1.1.0/24            24      13335      0       192.168.30.240/3323
1.9.0.0/16            24      4788       0       192.168.30.240/3323
1.9.12.0/24           24      65037      0       192.168.30.240/3323
1.9.21.0/24           24      24514      0       192.168.30.240/3323
1.9.23.0/24           24      65120      0       192.168.30.240/3323
1.9.31.0/24           24      65077      0       192.168.30.240/3323
1.9.65.0/24           24      24514      0       192.168.30.240/3323
1.34.0.0/15           24      3462       0       192.168.30.240/3323
1.36.0.0/19           19      4760       0       192.168.30.240/3323
```

17. Now check the BGP table again to see how the routes learned from your neighbors are tagged with the RPKI validation states of **Valid**, **Invalid** or **Not Found**:

```
show bgp ipv4 unicast
```

- Since we have created the ROAs corresponding to the prefixes used in this lab, you should see all of them tagged as valid (**V**). Example below for **R14**:



```
R14#sh bgp ipv4 unicast
BGP table version is 3, 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   172.16.0.1               0             0 135533 i
V*>  61.45.249.0/24   0.0.0.0                  0         32768 i
```

- Also verify the routing table (you should see the valid routes in the routing table)

```
sh ip route bgp
```

18. Let us now try to announce some **Invalid** routes (hijack someone's routes).

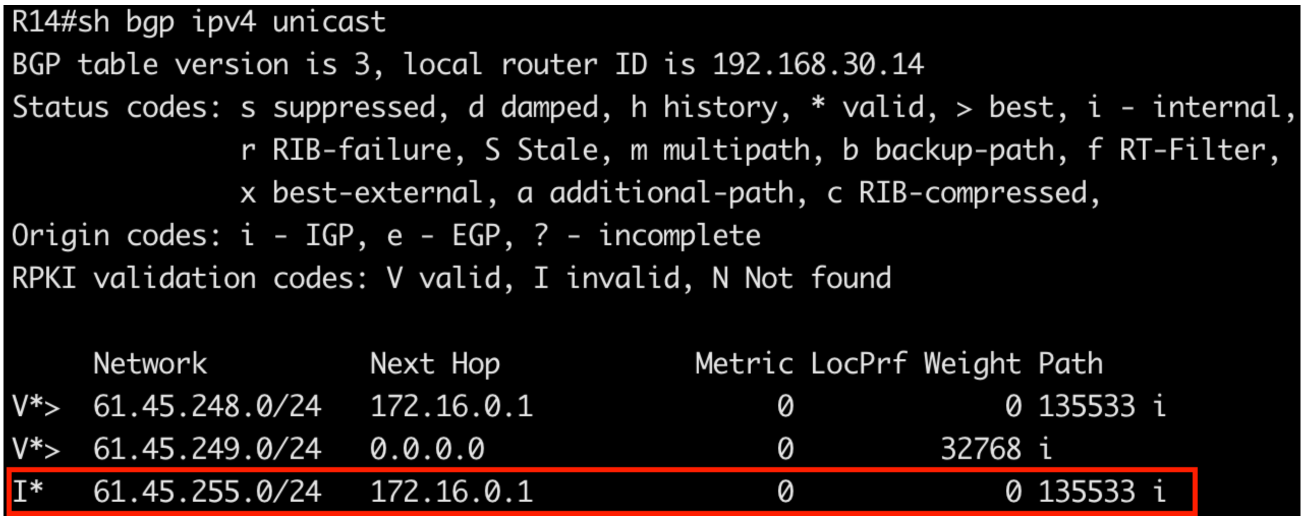
   - Go ahead and announce routes (refer the ip address plan) that belong to other ASNs. In the example below, **R13** in **AS135533** is announcing **R20**'s prefix (**AS135540**):

     ```
     ip route 61.45.255.0 255.255.255.0 null 0
     !
     router bgp 135533
      address-family ipv4 unicast
       network 61.45.255.0 mask 255.255.255.0
     ```

   - Verify the BGP table on **R14** (R13's eBGP neighbor).

     ```
     sh bgp ipv4 unicast
     ```

   - You will see that the route `61.45.255.0` learned from its neighbor **R13** has been tagged as **Invalid** (**I**). Discuss within your group why it is Invalid?



```
R14#sh bgp ipv4 unicast
BGP table version is 3, 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   172.16.0.1               0             0 135533 i
V*>  61.45.249.0/24   0.0.0.0                  0         32768 i
I*   61.45.255.0/24   172.16.0.1               0             0 135533 i
```

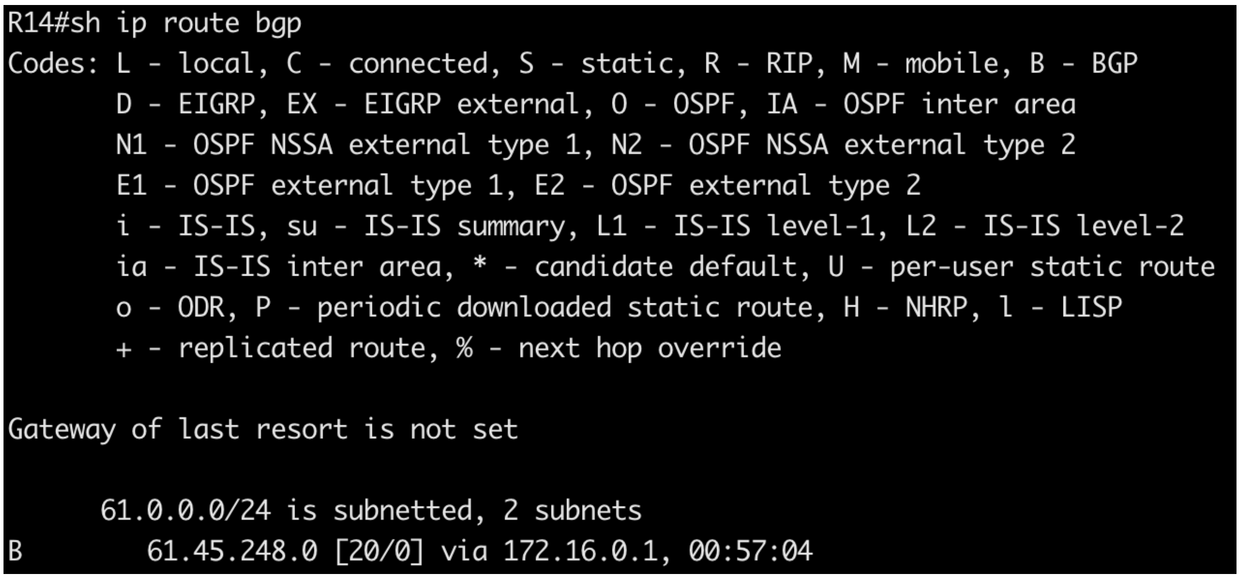
   - Now, look at the routing table:

     ```
     sh ip route bgp
     ```

     OR

     ```
     sh ip route
     ```

     - You will notice that the Invalid route has **NOT** be been inserted in the routing table.



```
R14#sh ip route bgp
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      61.0.0.0/24 is subnetted, 2 subnets
B        61.45.248.0 [20/0] via 172.16.0.1, 00:57:04
```

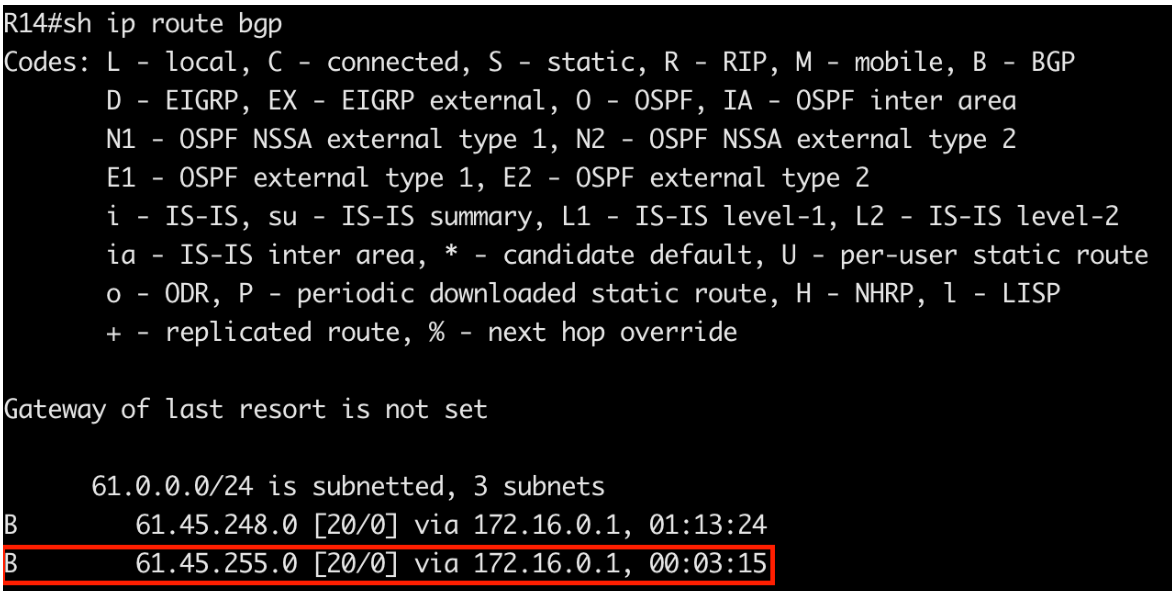
- ***NOTE: The default Cisco IOS (IOS-XE) behaviour is not to include Invalid routes for best path selection!***
- If you donot want to drop Invalids with Cisco IOS, you need to explicitly tell BGP to include invalids for best path selection (under respective address families) as shown below for **R14**:

```
router bgp 135534
 address-family ipv4 unicast
  bgp bestpath prefix-validate allow-invalid
```

  - Verify the routing table to see how BGP behaves with the above command:

  ```
  sh ip route bgp
  ```

  - The Invalid route now appears in the routing table of **R14** as shown below:



```
R14#sh ip route bgp
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      61.0.0.0/24 is subnetted, 3 subnets
B        61.45.248.0 [20/0] via 172.16.0.1, 01:13:24
B        61.45.255.0 [20/0] via 172.16.0.1, 00:03:15
```

19. Let us have a look at **Not Found** routes - routes for which there are no correspnding ROAs (neither valid or invalid, perhaps not created yet).

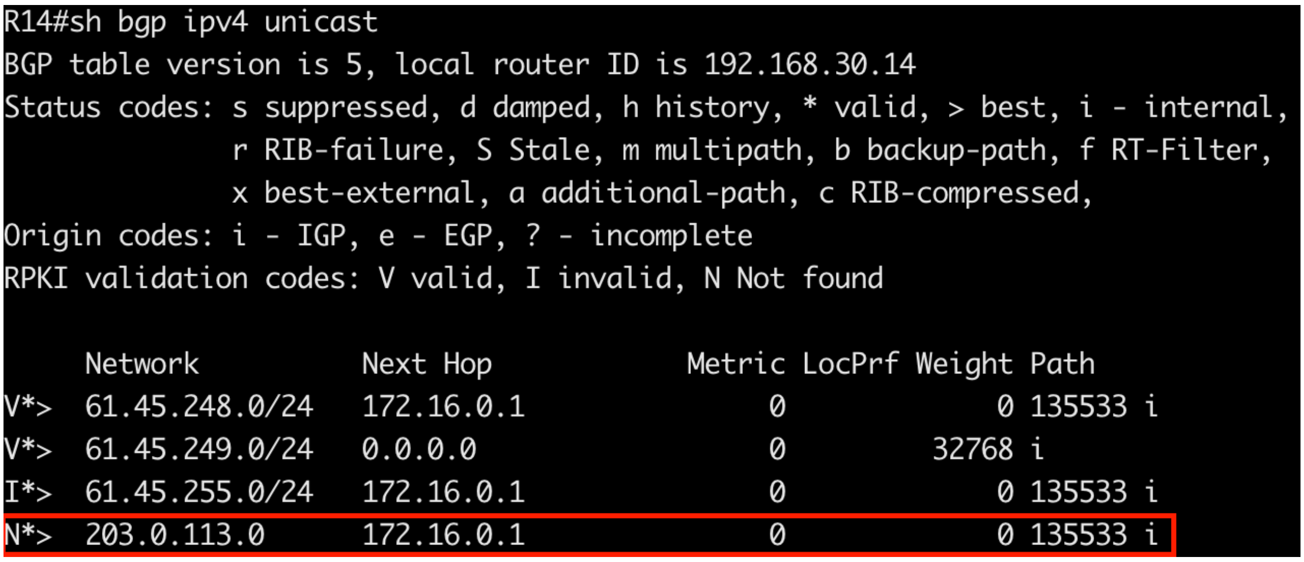
   - These make up more than 80% of the global routing table, which shows many networks haven't created ROAs for their prefix announcements! *(Waiting for stars to align??)*

   - Let us announce special use prefixes RFC5735, which should not have corresponding ROAs. Example below shows **R13** announcing the documentation prefix **203.0.113.0/24**

```
ip route 203.0.113.0 255.255.255.0 null 0
!
router bgp 135533
 address-family ipv4 unicast
  network 203.0.113.0 mask 255.255.255.0
```

     - For other routers, please feel free to use any prefixes listed in RFC5735 (please pick `/24s` from `198.18.0.0/15`). Note that if you and your eBGP peer both announce the same prefix, since it is locally originated, it will be marked as **Valid**. Make sure you communicate with your eBGP peer!

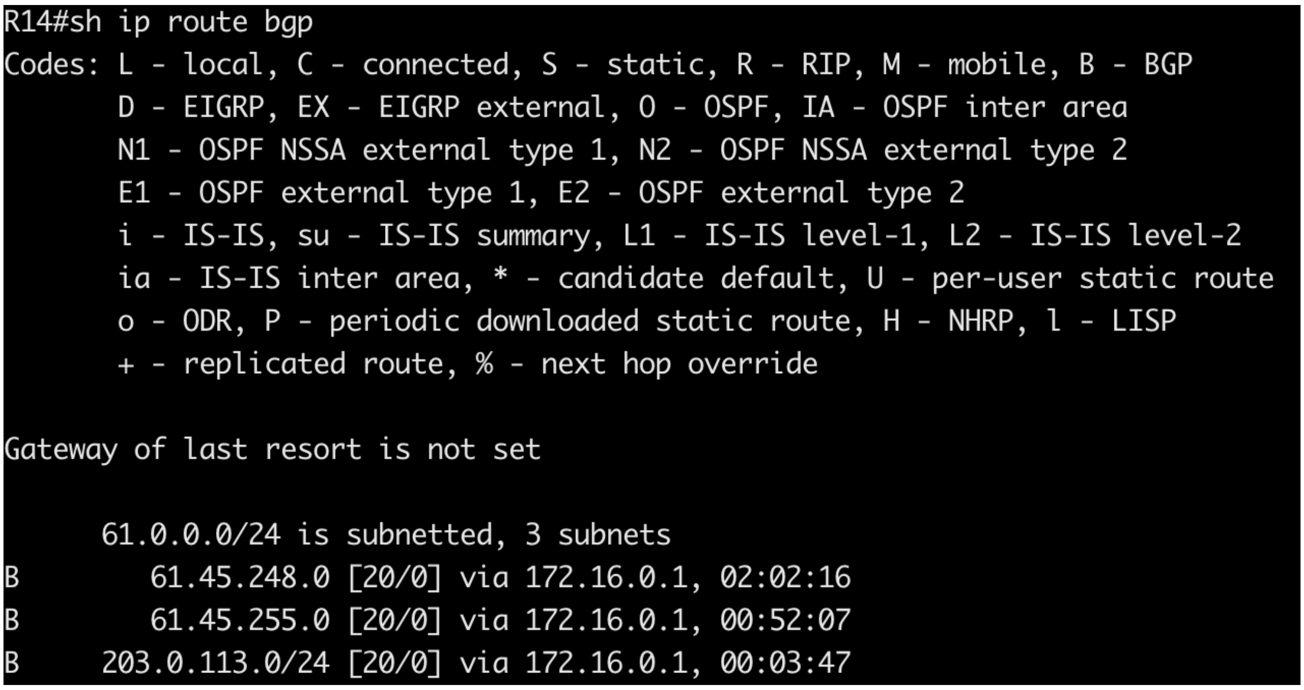
   - A look at **R14**'s BGP table:



```
R14#sh bgp ipv4 unicast
BGP table version is 5, 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   172.16.0.1               0             0 135533 i
V*>  61.45.249.0/24   0.0.0.0                  0         32768 i
I*>  61.45.255.0/24   172.16.0.1               0             0 135533 i
N*>  203.0.113.0      172.16.0.1               0             0 135533 i
```

   - And **R14**'s routing table shows the Not Found routes are included in the best path selection:



```
R14#sh ip route bgp
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      61.0.0.0/24 is subnetted, 3 subnets
B        61.45.248.0 [20/0] via 172.16.0.1, 02:02:16
B        61.45.255.0 [20/0] via 172.16.0.1, 00:52:07
B     203.0.113.0/24 [20/0] via 172.16.0.1, 00:03:47
```

20. If we do not want to drop Invalids, you can follow the recommendations in RFC7115 to prefer Valids over Not Found over Invalids:

   - Define a routing policy that prefers ***Valids > Not Founds > Invalids***

```
route-map ROUTE-VALIDATION permit 10
 match rpki valid
 set local-preference 200
!
route-map ROUTE-VALIDATION permit 20
 match rpki not-found
 set local-preference 100
!
route-map ROUTE-VALIDATION permit 30
 match rpki invalid
 set local-preference 50
```

   - Apply the route-map to inbound updates from your neighbor. Example below for **R20**:

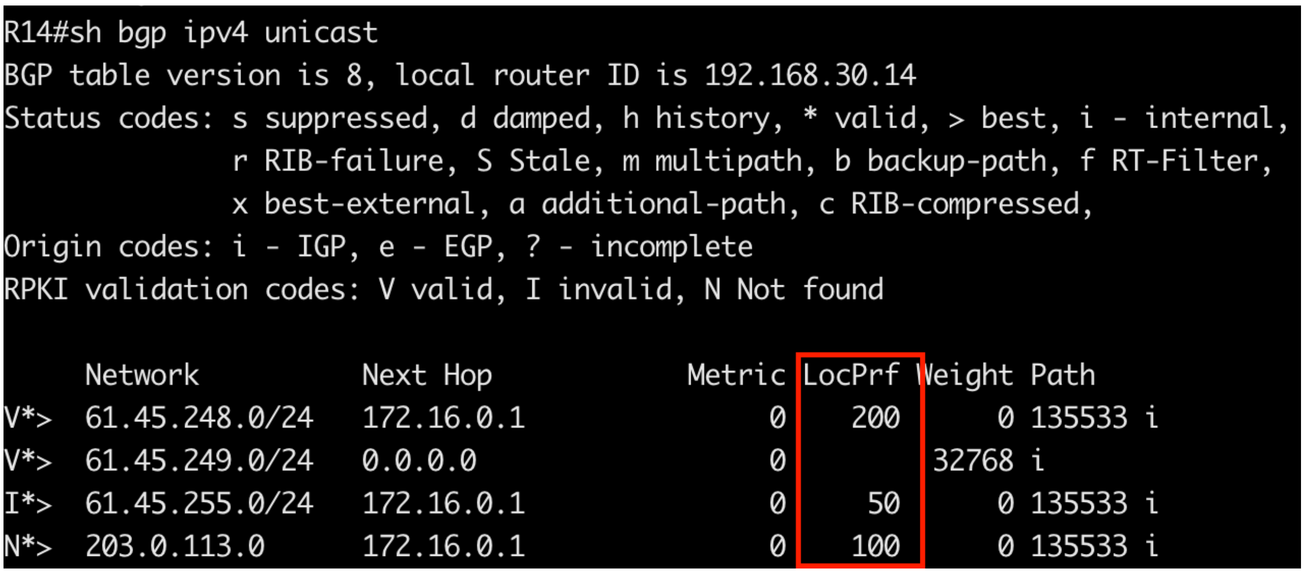
```
router bgp 135540
 address-family ipv4 unicast
  neighbor 172.16.0.13 route-map ROUTE-VALIDATION in
```

   - Refresh the routes learned from your neighbor (telling them to resend their routes without tearing down the BGP session). Example below for **R14**:

```
clear bgp ipv4 unicast 172.16.0.1 soft in
```

   - Now verify the BGP table (example **R14** below) to see the policy in action:

```
sh bgp ipv4 unicast
```



```
R14#sh bgp ipv4 unicast
BGP table version is 8, 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   172.16.0.1               0    200      0 135533 i
V*>  61.45.249.0/24   0.0.0.0                  0         32768 i
I*>  61.45.255.0/24   172.16.0.1               0     50      0 135533 i
N*>  203.0.113.0      172.16.0.1               0    100      0 135533 i
```