APNIC RPKI LAB

About this Lab
This Lab has been designed for you to configure DNS with DNSSEC on a Linux/Lubuntu server.

These are the ASN and IP assignment for each router:

<table>
<thead>
<tr>
<th>Router</th>
<th>AS Number</th>
<th>ge1 interface IP</th>
<th>Public IP prefix with ROA</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>AS135534</td>
<td>192.168.1.201/24</td>
<td>61.45.249.0/24</td>
</tr>
<tr>
<td>R2</td>
<td>AS135535</td>
<td>192.168.1.202/24</td>
<td>61.45.250.0/24</td>
</tr>
<tr>
<td>R3</td>
<td>AS135536</td>
<td>192.168.1.203/24</td>
<td>61.45.251.0/24</td>
</tr>
</tbody>
</table>

This shows the IP assignment for the P2P links:

<table>
<thead>
<tr>
<th>Links</th>
<th>IP block to use</th>
<th>Interface IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1-R3</td>
<td>10.0.13.0/24</td>
<td>R1 - 10.0.13.1, R3 - 10.0.13.3</td>
</tr>
<tr>
<td>R1-R2</td>
<td>10.0.12.0/24</td>
<td>R1 - 10.0.12.1, R2 - 10.0.12.2</td>
</tr>
<tr>
<td>R2-R3</td>
<td>10.0.23.0/24</td>
<td>R1 - 10.0.12.2, R2 - 10.0.12.3</td>
</tr>
</tbody>
</table>

Part 1 - RPKI Validator

Step 1 - Installing RIPE NCC Validator

Select H0_Host_Ubuntu from the Resources. Open a Terminal. A link should be available in the desktop.

Most of the following commands can be run From the Terminal window and sudo access is required.

```bash
user: host
password: root
```

To change to root user, use the command and input the password

```bash
sudo su -
```

Download the RIPE-NCC RPKI Validator (use wget in CLI below or use browser)

```bash
wget https://lirportal.ripe.net/certification/content/static/validator/rpki-validator-app-2.25-dist.tar.gz
```

Download and install the package

```bash
apt install default-jre
tar -zxvf rpki-validator-app-2.25-dist.tar.gz
cd rpki-validator-app-2.25
```
Install the ARIN TAL (optional)

```sh
cd rpki-validator-app-2.25
vi conf/rpki-validator.conf
```

Look for `rtr.port` and replace the value with 8323

```sh
rtr.port = 8323
```

Note that you may use any port. We have changed it in this lab so you know how to update the configuration.

Now stop and start the router

```
./rpki-validator.sh stop
./rpki-validator.sh start
```

**Step 2 - Installing Routinator**

Install curl and Rust.

```sh
apt-get install curl
apt-get install build-essential
curl https://sh.rustup.rs -ssSf | sh
```

Don't forget to use `sudo` at the start of the commands if not running as root user.

Install the routinator

```
sudo -c ~/.cargo/env
cargo install routinator
```

Initiate the routinator. The extra parameters are means you agree to the ARIN Relying Party Agreement (RPA)

```
routinator init --accept-arin-rpa
```

Run the routinator as an RTR server listening on port 3323.
routinator server --rtr 127.0.0.1:3323

To listen to a specific port
routinator server --rtr 192.168.1.100:3323

Or try `--help` to check all available options.

**Part 2 - Configuring RPKI on a Cisco Router**

**Step 1 - Interface and BGP Configuration for R1**

Select Router1_Cisco from the resources tab.

Configure the router interfaces. Check the lab topology.

```conf
t
int gi1
no ip unreachables
no ip redirects
no clns route-cache
ip address 192.168.1.201 255.255.255.0
no shutdown
int gi2
ip address 10.0.12.1 255.255.255.0
no shut
int gi3
ip address 10.0.13.1 255.255.255.0
no shut
exit
exit
```

Configure eBGP between R1 and R2

```conf
t
router bgp 135534
neighbor 10.0.12.2 remote-as 135535
neighbor 10.0.12.2 description peer with R3
neighbor 10.0.12.2 activate
exit
```

Verify
```show bgp ipv4 unicast summary```

Advertise your valid prefix

```conf
t```
router bgp 135533  
address-family ipv4 unicast
network 61.45.249.0 mask 255.255.255.0
exit
ip route 61.45.249.0 255.255.255.0 null0

**Step 2 - Interface and BGP Configuration for R2**

Now repeat the above steps for Router2. Make sure to use the correct IP address and ASN assignment as mentioned in the first page of the instruction

```
conf t
int gi1
no ip unreachables
no ip redirects
no clns route-cache
ip address 192.168.1.202 255.255.255.0
no shutdown
int gi2
ip address 10.0.21.1 255.255.255.0
no shut
int gi3
ip address 10.0.23.1 255.255.255.0
no shut
exit
exit
```

Configure eBGP with R1

```
conf t
router bgp 135534
neighbor 10.0.12.1 remote-as 135535
neighbor 10.0.12.1 description peer with R1
neighbor 10.0.12.1 activate
exit
```

Verify

```
show bgp ipv4 unicast summary
```

Advertise your valid prefix

```
conf t
router bgp 135533
address-family ipv4 unicast
network 61.45.250.0 mask 255.255.255.0
exit
```
Check that the peering is up and both routers are receiving one prefix from each other.

**Step 3 - RPKI-RTR Protocol**

Connect to the RPKI server. Before proceeding, make sure that the RPKI server has been configured in Part 1.

Now login to R1 and R2, and add the following configuration:

```conf
t
router bgp 135534
bgp rpki server tcp 192.168.1.100 port 8323 refresh 600
end
wr
```

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

Repeat the same config for R2.

```conf
t
router bgp 135535
bgp rpki server tcp 192.168.1.100 port 8323 refresh 600
end
wr
```

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

The server must be reachable from the router.

**Step 3 - Verify RPKI validation**

Now you may check if a prefix is validating.

```
sh ip bgp 61.45.249.0/24
```

And look for the term `RPKI State Valid`. This means the ip block has a ROA and being announced by the correct ASN.

You may also show all the routes, and a mark for RPKI validity is shown on the left of the prefix.

```
sh ip bgp
```
Step 4 - Creating rules (optional)

Create rules based on RPKI state.

First create a route map to match the RPKI states. Then set a local preference value depending on the state.

```conf
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
```

Then apply the route map to bgp

```conf
router bgp 135533
address-family ipv4
neighbor 10.0.X.3 route-map rpki-loc-pref in
end
wr
```

Step 5 - Propagating RPKI States via iBGP (optional)

Next configure Router 2 as iBGP instead, then add the following for RPKI.

```conf
address-family ipv4 unicast
neighbor x.x.x.x send-community extended
neighbor x.x.x.x announce rpki state
```

You have reached the end of this lab.