Route Origin Validation Lab

Part-1: Installing RPKI Validator (OctoRPKI)

VM Details

```
[192.168.30.13]
......
[192.168.30.20]
```

Login Details

- Username `apnic` and password `training`.

Preinstalled packages

To save time, the following essential package(s) have been preinstalled on your machines:

- `rsync`

Lab Setup

For this lab, we will use OctoRPKI from Cloudflare as the RPKI validator.

1. Login to your server (SSH from the jumphost to your machine using the `username` and `password` given above), where `X` is your VM number:

   ```
   ssh apnic@192.168.30.X
   ```

2. Update the repository

   ```
   sudo apt update && sudo apt upgrade
   ```

3. Download and install the validator:

   ```
   wget https://github.com/cloudflare/cfrpki/releases/download/v1.1.4/octorpki_1.1.4_amd64.deb
   dpkg -i octorpki_1.1.4_amd64.deb
   ```

4. Download the standard TALs for each RIR from the repo (except ARIN, which needs to be downloaded from here).

   - Note that by downloading ARIN's TAL, you agree to be bound by ARIN's Relying Party Agreement (RPA):

     ```
     mkdir tals
     cd tals
     wget https://raw.githubusercontent.com/cloudflare/cfrpki/master/cmd/octorpki/tals/afrinic.tal
     wget https://raw.githubusercontent.com/cloudflare/cfrpki/master/cmd/octorpki/tals/apnic.tal
     wget https://raw.githubusercontent.com/cloudflare/cfrpki/master/cmd/octorpki/tals/lacnic.tal
     wget https://raw.githubusercontent.com/cloudflare/cfrpki/master/cmd/octorpki/tals/ripe.tal
     wget https://www.arin.net/resources/manage/rpki/arin-rfc7730.tal -O arin.tal
     cd ..
     ```

5. Run the validator:

   ```
   nohup octorpki -output.sign=false > out 2> err &
   ```

6. Use the following command to retrieve the validated ROA payloads (produces a list of ASNs and prefixes). If this command produces the string "File not ready yet", then the validator is still working through the initial synchronisation process, which generally takes a few minutes. By default, the server will resynchronise its state every 20 minutes.

   ```
   curl localhost:8080/output.json
   ```

   - You can also access it through the web interface (`<validator-name/validator-address>:8080/output.json`)

Now the validator is ready to feed the validated cache to an rpki-rtr server, which in turn handles requests from BGP-speaking routers through the RTR (RPKI-to-Router) protocol.

Part-2: RTR session

Validator side

GoRTR is Cloudflare's rpki-rtr server component, which allows RPKI-enabled routers to connect to it and fetch the validated cache (ROA cache).

1. Download and install the rpki-rtr server:

```
wget https://github.com/cloudflare/gortr/releases/download/0.11.4/gortr_0.11.4_amd64.deb
dpkg -i gortr_0.11.4_amd64.deb
```

2. Run the server, listening for rpki-rtr requests on port `8282`, where `X` is your VM number:

```
nohup gortr -bind=192.168.30.X:8282 -metrics.addr=:8081 -verify=false -cache=http://localhost:8080/output.json > out 2> err &
```