Route Origin Validation - A MANRS Approach

Md. Zobair Khan
MANRS Ambassador
Acknowledgement

• This paper is made taking notes, diagram, configurations from MANRS, APNIC training materials & blog, NLnet Labs, Github & Dr. Philip Smith along with the operational experience of the author.
MANRS

• Mutually Agreed Norms for Routing Security (MANRS) is a global initiative, supported by the Internet Society, that provides crucial fixes to reduce the most common routing threats.
MANRS

• MANRS outlines four simple but concrete actions that network operators should take:
  • Filtering
  • Anti-spoofing
  • Coordination
  • Global Validation
Global Validation

- Publish your data, so others can validate routing information on a global scale
- Automated information validation needs arrangements
- Securing global routing information is done by RPKI
Resource Public Key Infrastructure (RPKI)

• RPKI allows holders of Internet number resources to make verifiable statements about how they intend to use their resources.

• RPKI is a way to define data in an out-of-band system such that the information that are exchanged by BGP can be validated to be correct.

• RPKI is used to make Internet routing more secure.
Importance of RPKI

• Secured Routing Table

• Dynamic LOA checking

• Maintaining a Dynamic Chain of Trust

• Digitally Signed Resources Certificate (X.509 Certificates-RFC5280)

• Helps to Stop Route Hijack
Route Origin Authorizations (ROA)

• Using the RPKI system, the legitimate holder of a block of IP addresses can use their resource certificate to make an authoritative, signed statement about which autonomous system is authorized to originate their prefix in BGP.

• These statements are called Route Origin Authorizations (ROAs).
Route Origin Validation (ROV)

- RPKI system tries to closely mimic what route objects in the IRR intend to do, but then in a more trustworthy manner.

- This process is called route origin validation (ROV)
Validity

• Valid - Resources found in database which is called Validated ROA Payload (VRP).

• Invalid – Resources found but partial/whole information doesn’t match with database.

• Not Found - The prefix in this announcement is not covered by a VRP.
RPKI - Chain Of Trust

RIR Root CA
- APNIC
- LACNIC
- ARIN
- RIPE NCC
- AFRINIC

NIR Child CA
- Some APNIC & LACNIC countries only
- NIR
- LIR
- LIR
- LIR
- LIR
- LIR

LIR Child CA
- ISP
- ISP

(Optional) ISP Child CA
Ecosystem
Working Steps

• Creating ROA for owned resources for RPKI

• Implementing Validator relying software for ROV

• Enforcing policies for based on Validation
Creating ROA

- Go to Resources > Route Management and select Create route
Creating ROA

• Mention your prefix with ASN & desired subnet & Submit

Example for IPv4

Relying Party Software

- NLnet Labs Routinator
  https://github.com/NLnetLabs/routinator

- LACNIC/NIC Mexico validator (FORT)
  https://github.com/NICMx/FORT-validator

- Cloudflare validator (OctoRPKI)
  https://github.com/cloudflare/cfrpki

- RPKI-Client
  https://github.com/kristapsdz/rpki-client
Routinator

- Routinator is free, open source RPKI Relying Party software written by NLnet Labs in the Rust programming language.

- Routinator connects to the Trust Anchors of the five Regional Internet Registries (RIRs) — APNIC, AFRINIC, ARIN, LACNIC and RIPE NCC — downloads all of the certificates and ROAs in the various repositories, verifies the signatures and makes the result available for use in the BGP workflow.

- The validated cache can be fed directly into RPKI-capable routers via the RPKI to Router Protocol (RPKI-RTR), described in RFC 8210.

Installation (Routinator)

Step by step installation process is given in Appendix-1 at the end of this slide.
OctoRPKI

- Developed and used by Cloudflare
- This application periodically refreshes the data provided by the RIRs and the delegated organizations.
- No development after December 2020.
- RTR session is handled by GoRTR – a separate module.
Installation (OctoRPKI)

Step by step installation process is given in Appendix-1 at the end of this slide.
FORT Validator

• FORT validator is an MIT-licensed RPKI Relying Party, this is a tool offered as part of the FORT project.

• FORT is a joint initiative by LACNIC, the Regional Internet Registry for Latin America and the Caribbean, and NIC.MX.

• RTR Configuration is done with json file.
Installation (FORT)

Start fort as a service.

Check the status. It will take some time to build the first validated cache.

Step by step installation process is given in Appendix-1 at the end of this slide.
RPKI-Client

• rpki-client is a FREE, easy-to-use implementation of the Resource Public Key Infrastructure (RPKI)

• The program queries the RPKI repository system and outputs Validated ROA Payloads in the configuration format of either OpenBGPD or BIRD, but also as CSV or JSON objects for consumption by other routing stacks.

• rpki-client was primarily developed as part of the OpenBSD Project and gets released as a base component of OpenBSD every six months.
Installation (RPKI-Client)

If everything goes smoothly, you will be able to see the ROA summary.

`bgpctl show sets`

You can apply RPKI information to reject invalids on import and export

`deny quick from ebgp ovs invalid`
`deny quick to ebgp ovs invalid`

Step by step installation process is given in Appendix-1 at the end of this slide.
Hardware OS

**Juniper** - Junos version 12.2 and newer.

**Cisco** - IOS release 15.2 and newer, as well as Cisco IOS/XR since release 4.3.2.

**Nokia** - SR OS 12.0.R4 and newer, running on the 7210 SAS, 7250 IXR, 7750 SR, 7950 XRS and the VSR.

**Arista** - EOS 4.24.0F and newer

**MikroTik** - 7.0beta7 and newer

**Huawei** - VRP 8.150 and newer.
Router Configuration

```
router bgp [ASN]
rpki server [SERVER IP]
transport tcp port 3323
refresh-time 120

address-family ipv4 unicast
bgp origin-as validation signal ibgp

address-family ipv6 unicast
bgp origin-as validation signal ibgp
```
Decision

• Since now the validation states are visible to you, you can decide what to do with invalids

• You can –
  – Use them with low preference
  – Or drop them
Policy

route-policy RPKI
  if validation-state is invalid then
    set local-preference 50
  else
    if validation-state is valid then
      set local-preference 200
    else
      pass
    endif
  endif
endif
end-policy

route-policy RPKI
  if validation-state is invalid then
    drop
  else
    if validation-state is valid then
      set local-preference 200
    else
      pass
    endif
  endif
endif
end-policy
FAH Deployment

• Fiber@Home is one of the largest Internet Gateway in Bangladesh.
• We have implemented Routinator as our validator software.
• We are still in observation period and not dropping the invalids yet.
• Communicating with our clients to make them ready for the dropping of invalids.
• A proud member of MANRS and getting all the benefits from being in the elite platform of Routing Security.
• Securing global routing information is done by RPKI.
One Last Thing

• ARIN has announced a surprised maintenance of 30 min in July 2021. Announcement was made on 2\textsuperscript{nd} June 2021.
• They are doing it to make sure that their RPKI infrastructure is running with peak performance.
• \url{https://www.arin.net/announcements/20210602-rpki/}

• Also Job Snijders has shared some valuable insights on this maintenance work.
• \url{https://lists.arin.net/pipermail/arin-tech-discuss/2021-June/000932.html}

• Our learning –
  – When you deploy RPKI, you need to keep an eye on your RPKI infrastructure performance.
  – You need to make sure that any maintenance work is informed to your clients way ahead.
  – Also you need to be aware of the events taking places in RIRs regarding their RPKI Infrastructure.
Reference

- https://www.manrs.org/about/
- https://blog.apnic.net/2019/10/28/how-to-installing-an-rpki-validator/
- https://github.com/cloudflare/cfrpki#ctorpki
- https://www.rpki-client.org/
Appendix-1

- Step by Step Installation of Routinator
- Step by Step Installation of OctoRPKI
- Step by Step Installation of Fort
- Step by Step Installation of RPKI-Client
Appendix-1

• Step by Step Installation of Routinator
Installation (Routinator)

Add the line below that corresponds to your operating system to your `/etc/apt/sources.list` or `/etc/apt/sources.list.d/`

```
deb [arch=amd64] https://packages.nlnetlabs.nl/linux/debian/ stretch main
deb [arch=amd64] https://packages.nlnetlabs.nl/linux/debian/ buster main
deb [arch=amd64] https://packages.nlnetlabs.nl/linux/ubuntu/ xenial main
deb [arch=amd64] https://packages.nlnetlabs.nl/linux/ubuntu/ bionic main
deb [arch=amd64] https://packages.nlnetlabs.nl/linux/ubuntu/ focal main
```
Installation (Routinator)

Then run the following commands to add the public key and update the repository list:

```
wget -qO- https://packages.nlnetlabs.nl/aptkey.asc | sudo apt-key add -
sudo apt update
```
You can then install Routinator by running this command

```
sudo apt install routinator
```

```bash
nano@ROV:~$ sudo apt install routinator
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  routinator
0 upgraded, 1 newly installed, 0 to remove and 197 not upgraded.
Need to get 3,287 kB of archives.
After this operation, 10.1 MB of additional disk space will be used.
Get:1 https://packages.ninetlabs.nl/linux/ubuntu xenial/main amd64 routinator amd64 0.9.0-1xenial [3,287 kB]
Fetched 3,287 kB in 2s (1,438 kB/s)
Selecting previously unselected package routinator.
(Reading database ... 80683 files and directories currently installed.)
Preparing to unpack .../routinator_0.9.0-1xenial_amd64.deb ...
Unpacking routinator (0.9.0-1xenial) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up routinator (0.9.0-1xenial) ...
Adding system user 'routinator' (UID 113) ...
Adding new group 'routinator' (GID 120) ...
Adding new user 'routinator' (UID 113) with group 'routinator' ...
Creating home directory '/var/lib/routinator/' ...
routinator.service is a disabled or a static unit, not starting it.
```
Installation (Routinator)

Initialise, enable and start Routinator by running these commands.

```bash
sudo routinator-init  # (This command should be given in /home/user/ directory)
sudo systemctl enable --now routinator
```

```bash
nano@ROV:~$ sudo routinator-init
Running command as user routinator: routinator --config /etc/routinator/routinator.conf init
before we can install the ARIN TAL, you must have read
and agree to the ARIN Relying Party Agreement (RPA).
It is available at
https://www.arin.net/resources/manage/rpki/rpa.pdf

If you agree to the RPA, please run the command
again with the --accept-arin-rpa option.
Fatal error. Exiting.
```

```bash
nano@ROV:~$ sudo routinator-init --accept-arin-rpa
Running command as user routinator: routinator --config /etc/routinator/routinator.conf init --accept-arin-rpa
Created local repository directory /var/lib/routinator/rpki-cache
Installed 5 TALs in /var/lib/routinator/tals
```

```bash
nano@ROV:~$ sudo systemctl enable --now routinator
Created symlink from /etc/systemd/system/multi-user.target.wants/routinator.service to /lib/systemd/system/routinator.service.
nano@ROV:~$
```
Installation (Routinator)

By default, Routinator will start the RTR server on port 3323 and the HTTP server on port 8323 which can be changed in `/etc/routinator/routinator.conf`. You can check the status of Routinator with `sudo systemctl status routinator` and view the logs with `sudo journalctl --unit=routinator`
Installation (Routinator)

routinator -v vrps

routinator server --rtr SERVER-IP:3323 --http SERVER-IP:9556 --refresh=900 &
Installation (Routinator)
Appendix-1

• Step by Step Installation of OctoRPKI
Installation (OctoRPKI)

First, go to the Releases tab, download the latest version matching your platform.
Installation (OctoRPKI)

To install the Linux deb package and start it:

```bash
$ sudo dpkg -i octorpki_1.1.4_amd64.deb
$ sudo systemctl start octorpki
```
Installation (OctoRPKI)

mkdir tals

cd tals

wget https://www.arin.net/resources/manage/rpki/arin-rfc7730.tal -O arin.tal

cd ..
Installation (OctoRPKI)

Look into the tals directory to check if you have got all the 5 tals or not.

Run the validator by issuing the following command.

```
nohup octorpki -output.sign=false > out 2> err &
```
Installation (OctoRPKI)

If the validator runs smoothly, you will see the ROAs in web format by querying a json file.

The validated cache output uses default port 8081.

GoRTR metrics uses default port 8080.

You can build RTR session on your desired port.
Installation (OctoRPKI)

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

```bash
wget https://github.com/cloudflare/gotr/releases/download/v0.14.7/gotr_0.14.7_amd64.deb
dpkg -i gotr_0.14.7_amd64.deb
```
Installation (OctoRPKI)

```
```

```
root@routinator-2:~#
root@routinator-2:~# nophup gortr -bind=163.47.159.22:8282 -metrics.addr=:8080 -verify=false -cache=http://163.47.159.22:8081/output.json > out 2> err &
[2] 6483
root@routinator-2:~#

root@routinator-2:~# ps ax | grep gortr
  6483 pts/0   Sl  0:06 gortr -bind=163.47.159.22:8282 -metrics.addr=:8080 -verify=false -cache=http://163.47.159.22:8081/output.json
  6584 pts/0   Sr+ 0:00 grep --color=auto gortr

root@routinator-2:~# ps ax | grep octo
  5690 pts/0   Sl  6:02 octorpi -output.sign=false
  6588 pts/0   Sr+ 0:00 grep --color=auto octo

root@routinator-2:~#
```

```
RP/0/RSP1/CPU0:C-AGG-1#sh bgp rpki server summary
Mon Jun  7 16:20:55.602 BST

Hostname/Address | Transport | State | Time | ROAS (IPv4/IPv6)
-----------------|-----------|-------|------|----------------
163.47.159.22    | TCP:8282  | ESTAB | 00:02:07 | 214077/39923

RP/0/RSP1/CPU0:C-AGG-1#
```
Appendix-1

• Step by Step Installation of Fort
Installation (FORT)

It has some dependencies. So need to resolve those first.

```
sudo apt install openssl rsync libjansson-dev -y
```

Version 1.5.0 is the latest official release. To fetch or review it, visit the GitHub release

```
sudo wget https://github.com/NICMx/FORT-validator/releases/download/v1.5.0/fort_1.5.0-1_amd64.deb
sudo apt install ./fort_1.5.0-1_amd64.deb
```

Installation (FORT)

You will find an example config.json file inside `/etc/fort/example`.

Rename the actual config.json file to something else and copy the config file from example directory to `/etc/fort/`

In this way it will be much more easier to configure without destroying the json format.

Change the below parameters if required.

Tal
Local-repository
Address
Port

Rest of the fields should be fine with default values.
Installation (FORT)

Get the tals including Arin.

```
sudo fort --init-tals --tal=/etc/fort/tal
```

```bash
nano@routinator-2:~$ cd /etc/fort/tal
nano@routinator-2:/etc/fort/tal$
Please download and read ARIN Relying Party Agreement (RPA) from https://www.arin.net/resources/manage/rpki/rpa.pdf. Once you've read it and if you agree ARIN RPA, type 'yes' to proceed with ARIN's TAL download:

```
yes
Successfully fetched '/etc/fort/tal/arin.tal'!
Successfully fetched '/etc/fort/tal/apnic.tal'!
Successfully fetched '/etc/fort/tal/afrinic.tal'!
Successfully fetched '/etc/fort/tal/ripe.tal'!
Successfully fetched '/etc/fort/tal/lacnic.tal'!
```
Installation (FORT)

Start fort as a service.

Check the status. It will take some time to build the first validated cache.
Appendix-1

• Step by Step Installation of RPKI-Client
Installation (RPKI-Client)

First, you'll need a recent OpenSSL library (version 1.1.1c and above) on your operating system.

Next, you'll need the /var/cache/rpki-client directory in place. It must be writable by the operator of rpki-client.

```
mkdir /var/cache/rpki-client
```

The default output directory is /var/db/rpki-client, which must also be writable (if not overridden).

```
mkdir /var/db/rpki-client
```
Installation (RPKI-Client)

Get the software from git hub

`gh repo clone kristapsdz/rpki-client`

Edit the make file and put 0 in the value of RPKI_PRIVDROP. If the value is 0, it doesn’t use the Privilege Drop user.
Installation (RPKI-Client)

Get the TALs.

fetch https://rpki.afrinic.net/tal/afrinic.tal
fetch https://www.arin.net/resources/manage/rpki/arin-rfc7730.tal
fetch https://www.lacnic.net/innovaportal/file/4983/1/lacnic.tal
fetch https://tal.apnic.net/apnic.tal
fetch https://tal.rpki.ripe.net/ripe-ncc.tal
Installation (RPKI-Client)

You will be needing open rsync for cache refresh operation.

```
pkg install openrsync
```

Go to rpki-client directory and Install rpki-client and run it with mentioning the cache location.

```
./configure
make
make install
./rpki-client --v --d /var/cache/rpki-client/ /var/db/rpki-client/
```
Installation (RPKI-Client)

If everything goes smoothly, you will be able to see the ROA summury.
```
bgpctl show sets
```

You can apply RPKI information to reject invalids on import and export
```
deny quick from ebgp ovs invalid
deny quick to ebgp ovs invalid
```
Query !!

Join MANRS : manrs.org/join/

Thanks ...