Securing Internet Routing: RPKI

SANOG 28
01 - 09 August, 2016
Mumbai, India
Fakrul Alam

Senior Training Officer

Fakrul is responsible for the development and delivery of technical training to the APNIC community and works closely with network operating members in the Asia Pacific region. His specialist training areas include Routing & Switching, Network Architecture, Network Security & Management and Network Forensics.

Prior to joining APNIC, Fakrul worked for several organizations which includes IXP, ISP, Financial Institutes. He has strong knowledge of, and operational experience in building and deploying scalable, reliable network infrastructure.

email: fakrul@apnic.net
After graduating from China’s Huazhong University of Science and Technology in 2007 with a degree in electronics engineering, Bei (whose nickname is Jessica) joined Huawei as a network training officer.

Over the next six years, she provided Huawei technical training on LAN/WAN systems, broadband access, IP core and IP mobile backhaul networks as well as working on technical training course design and the development of IP training materials. At the Huawei training center in China she provided technical training to engineers and administrators from more than 15 nations including Vietnam, Papua New Guinea, Thailand, Pakistan and Bangladesh.

Email: jwei@apnic.net
Purpose of RPKI

• RPKI replaces IRR or lives side by side?
  – Side by side: different advantages
  – Security, almost real time, simple interface: RPKI

• Purpose of RPKI
  – Is that ASN authorized to originate that address range?
Send a packet to 2001:DB8::1

I have 2001:DB8::/32

Valid

Invalid
RPKI Deployment

Phase 1
Origin Validation

Phase 2
Path Validation

Send a packet to 2001:DB8::1

I have 2001:DB8::/32

65552

65551

65550

65549

Phase 1
Origin Validation
Internet Registry (IR) / RIR

- Maintains Internet Resources such as IP addresses and ASNs, and publish the registration information
  - Allocations for Local Internet Registries
  - Assignments for end-users

- APNIC is the Regional Internet Registry (RIR) in the Asia Pacific region
  - National Internet Registry (NIR) exists in several economies
The Eco-System

Internet Assigned Numbers Authority

Regional IR (RIR)

National IR (NIR)

Internet Service Provider

End User
Goals of RPKI

• Able to authoritatively prove who owns an IP Prefix and what AS(s) may Announce It
  – Reducing routing leaks
  – Attaching digital certificates to network resources (AS Number & IP Address)

• Prefix Ownership Follows the Allocation Hierarchy IANA, RIRs, ISPs, …
Advantage of RPKI

• Useable toolset
  – No installation required
  – Easy to configure manual overrides

• Tight integration with routers
  – Supported routers have awareness of RPKI validity states

• Stepping stone for AS-Path Validation
  – Prevent Attacks on BGP
RPKI Implementation

• Two RPKI implementation type
  – **Delegated**: Each participating node becomes a CA and runs their own RPKI repository, delegated by the parent CA.
  – **Hosted**: The RIR runs the CA functionality for interested participants.
Two Components

• Certificate Authority (CA)
  – Internet Registries (RIR, NIR, Large LIR)
  – Issue certificates for customers
  – Allow customers to use the CA’s GUI to issue ROAs for their prefixes

• Relying Party (RP)
  – Software which gathers data from CAs
Issuing Party

- Internet Registries (RIR, NIR, Large LIRs)
- Acts as a Certificate Authority and issues certificates for customers
- Provides a web interface to issue ROAs for customer prefixes
- Publishes the ROA records

MyAPNIC GUI → APNIC RPKI Engine → publication → Repository rpki.apnic.net
Relying Party (RP)

Software which gathers data from CAs
Also called RP cache or validator
RPKI Building Blocks

1. Trust Anchors (RIR’s)
2. Route Origination Authorizations (ROA)
3. Validators
1. PKI & Trust Anchors
Public Key Concept

- **Private key**: This key must be known only by its owner.
- **Public key**: This key is known to everyone (it is public)
- **Relation between both keys**: What one key encrypts, the other one decrypts, and vice versa. That means that if you encrypt something with my public key (which you would know, because it's public :-), I would need my private key to decrypt the message.
- Same alike http with SSL aka https
RPKI Profile

X.509 Certificates 3779 EXT

Certificates are X.509 certificates that conform to the PKIX profile [PKIX]. They also contain an extension field that lists a collection of IP resources (IPv4 addresses, IPv6 addresses and AS Numbers) [RFC3779]
Trust Anchor

Resource Allocation Hierarchy

- AFRINIC
- RIPE NCC
- ARIN
- APNIC
- LACNIC

Trust Anchor Certificate

Source: http://isoc.org/wp/ietfjournal/?p=2438
RPKI Chain of Trust

• The RIRs hold a self-signed root certificate for all the resources that they have in the registry
  – They are the trust anchor for the system
• That root certificate is used to sign a certificate that lists your resources
• You can issue child certificates for those resources to your customers
  – When making assignments or sub allocations
2. ROA
Route Origin Authorizations
Route Origination Authorizations (ROA)

- A ROA is a **digitally signed object** that provides a means of **verifying** that an **IP address block holder** has **authorized** an **Autonomous System (AS)** to originate routes to one or more **prefixes** within the address block.

- With a **ROA**, the **resource holder is attesting** that the **origin AS number** is **authorized** to **announce** the **prefix(es)**. The attestation can be verified cryptographically using RPKI.
Route Origination Authorizations (ROA)

• Next to the prefix and the ASN which is allowed to announce it, the ROA contains:
  – A minimum prefix length
  – A maximum prefix length
  – An expiry date
  – Origin ASN

• Multiple ROAs can exist for the same prefix

• ROAs can overlap
3. Validators
Origin Validation

• Router gets ROA information from the RPKI Cache
  – RPKI verification is done by the RPKI Cache

• The BGP process will check each announcement with the ROA information and label the prefix
Result of Check

- **Valid** – Indicates that the prefix and AS pair are found in the database.

- **Invalid** – Indicates that the prefix is found, but either the corresponding AS received from the EBGP peer is not the AS that appears in the database, or the prefix length in the BGP update message is longer than the maximum length permitted in the database.

- **Not Found / Unknown** – Indicates that the prefix is not among the prefixes or prefix ranges in the database.

Valid > Unknown > Invalid
**ROA Example**

<table>
<thead>
<tr>
<th>ROA</th>
<th>Origin AS</th>
<th>Prefix</th>
<th>Max Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>VALID</td>
<td>AS65420</td>
<td>10.0.0.0/16</td>
<td></td>
</tr>
<tr>
<td>VALID</td>
<td>AS65420</td>
<td>10.0.128.0/17</td>
<td></td>
</tr>
<tr>
<td>INVALID</td>
<td>AS65421</td>
<td>10.0.0.0/16</td>
<td></td>
</tr>
<tr>
<td>INVALID</td>
<td>AS65420</td>
<td>10.0.10.0/24</td>
<td></td>
</tr>
<tr>
<td>UNKNOWN</td>
<td>AS65430</td>
<td>10.0.0.0/8</td>
<td></td>
</tr>
</tbody>
</table>

Prefix: 10.0.0.0/16
ASN: 65420
Local Policy

• You can define your policy based on the outcomes
  – Do nothing
  – Just logging
  – Label BGP communities
  – Modify preference values
  – Rejecting the announcement
In summary

• As an announcer/LIR
  – You choose if you want certification
  – You choose if you want to create ROAs
  – You choose AS, max length

• As a Relying Party
  – You can choose if you use the validator
  – You can override the lists of valid ROAs in the cache, adding or removing valid ROAs locally
  – You can choose to make any routing decisions based on the results of the BGP Verification (valid/invalid/unknown)
RPKI Caveats

- When RTR session goes down, the RPKI status will be not found for all the bgp route after a while
  - Invalid => not found
  - we need several RTR sessions or care your filtering policy

- In case of the router reload, which one is faster, receiving ROAs or receiving BGP routes?
  - If receiving BGP is match faster than ROA, the router propagate the invalid route to others
  - We need to put our Cache validator within our IGP scope
RPKI Further Reading

• RFC 5280: X.509 PKI Certificates
• RFC 3779: Extensions for IP Addresses and ASNs
• RFC 6481-6493: Resource Public Key Infrastructure
RPKI Configuration
RPKI Configuration

• Resources:
  – AS: 131107 [APNICTRAINING-DC]
  – IPv4: 202.125.96.0/24
  – IPv6: 2001:df2:ee00::/48

• Process
  – Create ROA
  – Setup cache validation server
  – Validate the ROA
Implementation Scenario

ASBR

Intranet

Internet

Trust Anchors

Trust Anchors

Sync with RPKI Trust Anchors

RPKI to RTR protocol

Data Center

RPKI Cache Validator

Trust Anchors

Trust Anchors
Phase I - Publishing ROA

- Login to your MyAPNIC portal
- Required valid certificate
- Go to Resources > Certification Tab
Phase I - Publishing ROA

1. Click on the "Certification" tab.

2. Select the option: "I want to operate in the MyAPNIC RPKI portal."

3. Click on "I accept. Create my Certification Authority."
Phase I - Publishing ROA

• Show available prefix for which you can create ROA

<table>
<thead>
<tr>
<th>Origin AS</th>
<th>Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>131107</td>
<td>2001:df2:ee00::/48</td>
</tr>
<tr>
<td>131107</td>
<td>202.125.96.0/24</td>
</tr>
</tbody>
</table>
Phase I - Publishing ROA

ROA Configuration

- Create ROA for smaller block.

1. Write your ASN
2. Your IP Block
3. Subnet
4. Click Add

Certified Resources

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>61.45.248.0/23</td>
<td></td>
</tr>
<tr>
<td>61.45.251.0/24</td>
<td></td>
</tr>
<tr>
<td>61.45.253.0/24</td>
<td></td>
</tr>
<tr>
<td>203.176.189.0/24</td>
<td></td>
</tr>
<tr>
<td>2001:DF0:A::/48</td>
<td></td>
</tr>
<tr>
<td>2406:6400::/32</td>
<td></td>
</tr>
</tbody>
</table>
Phase I - Check your ROA

```
~ D/A/R RPSL-DEMO master whois -h whois.bgpmon.net 202.125.96.0/24
% This is the BGPmon.net whois Service
% You can use this whois gateway to retrieve information
% about an IP address or prefix
% We support both IPv4 and IPv6 address.
%
% For more information visit:
% https://portal.bgpmon.net/bgpmonapi.php

Prefix: 202.125.96.0/24
Prefix description: APNICTRAINING-DC
Country code: MN
Origin AS: 131107
Origin AS Name: ASN for APNICTRAINING LAB DC
RPKI status: ROA validation successful
First seen: 2016-06-21
Last seen: 2016-08-03
Seen by #peers: 248
```
Phase I - Check your ROA

whois -h whois.bgpmon.net " --roa 131107 202.125.96.0/24"

0 - Valid

-------------------
ROA Details
-------------------
Origin ASN: AS131107
Not valid Before: 2016-06-22 05:00:07
Not valid After: 2020-07-30 00:00:00 Expires in 3y360d1h32m3.79999998211861s
Trust Anchor: rpki.apnic.net
Prefixes: 202.125.96.0/24 (max length /24)
Phase II - RPKI Validator

• Download RPKI Validator

Tools and Resources

Here you can find an overview of all information and tools for the Resource Certification (RPKI) service.

RIPE NCC RPKI Validator 2.21 (Updated 3 November 2015)

This application allows operators to download and validate the global RPKI data set for use in their BGP decision making process and router configuration.

System requirements: a UNIX-like OS, Java 7, rsync and 2GB free memory.
To install, simply unpack the archive and run 'rpki-validator.sh' from the base folder.

For more information, view the release notes. You can also contribute to the project on GitHub.
Phase II - RPKI Validator

# tar -zxvf rpki-validator-app-2.21-dist.tar.gz
# cd rpki-validator-app-2.21
# ./rpki-validator.sh start
Phase II - RPKI Validator

http://ip_address:8080

Quick Overview of BGP Origin Validation

Trust anchors are the entry points used for validation in any Public Key Infrastructure (PKI) system. This RPKI Validator is preconfigured with the trust anchors for APNIC, ARIN, LACNIC, and RIR, including the RPKI repository. In order to obtain the trust anchor for the ARIN RPKI repository, you will first have to accept their Relying Party Agreement. Please refer to the README.txt for details on how to add trust anchors to this application.

Configured Trust Anchors

Router Sessions

This table shows all routers connected to this RPKI Validator. Requests and responses are described in RFC 6810. For debugging, please refer to /r/log.
Phase III - Router Configuration

1. Establish session with RPKI Validator

### Junos

```bash
set routing-options validation group RPKI session 202.125.96.46 refresh-time 120
set routing-options validation group RPKI session 202.125.96.46 hold-time 180
set routing-options validation group RPKI session 202.125.96.46 port 8282
set routing-options validation group RPKI session 202.125.96.46 local-address 103.21.75.1
```

### IOS

```bash
router bgp 64500
    bgp log-neighbor-changes
    bgp rpki server tcp 202.125.96.46 port 8282 refresh 120
```
Phase III - Router Configuration

2. Configure policy to tag ROA

<table>
<thead>
<tr>
<th>Junos</th>
</tr>
</thead>
</table>
| set policy-options policy-statement ROUTE-VALIDATION term valid from protocol bgp  
  valid |
| set policy-options policy-statement ROUTE-VALIDATION term valid then local-preference 110 |
| set policy-options policy-statement ROUTE-VALIDATION term valid then validation-state valid |
| set policy-options policy-statement ROUTE-VALIDATION term invalid from protocol bgp invalid |
| set policy-options policy-statement ROUTE-VALIDATION term invalid then local-preference 90 |
| set policy-options policy-statement ROUTE-VALIDATION term invalid then validation-state invalid |
| set policy-options policy-statement ROUTE-VALIDATION term invalid then accept |
| set policy-options policy-statement ROUTE-VALIDATION term unknown from protocol bgp unknown |
| set policy-options policy-statement ROUTE-VALIDATION term unknown then local-preference 100 |
| set policy-options policy-statement ROUTE-VALIDATION term unknown then validation-state unknown |
| set policy-options policy-statement ROUTE-VALIDATION term unknown then accept |
Phase III - Router Configuration

2. Configure policy to tag ROA

```plaintext
IOS

! route-map ROUTE-VALIDATION permit 10
  match rpki invalid
  set local-preference 90

! route-map ROUTE-VALIDATION permit 20
  match rpki not-found
  set local-preference 100

! route-map ROUTE-VALIDATION permit 30
  match rpki valid
  set local-preference 110
```
Phase III - Router Configuration

3. Push policy to the BGP neighbour

**Junos**

| set protocols bgp import ROUTE-VALIDATION |

**IOS**

<table>
<thead>
<tr>
<th>router bgp 64500</th>
</tr>
</thead>
<tbody>
<tr>
<td>bgp log-neighbor-changes</td>
</tr>
<tr>
<td>!other neighbour related configuration</td>
</tr>
<tr>
<td>neighbor 10.1.1.2 route-map ROUTE-VALIDATION in</td>
</tr>
</tbody>
</table>
# Check your prefix

## Junos

```
rpki-rtr>show route protocol bgp 202.125.96.46/24
202.125.96.0/24    *[BGP/170] 3w5d 16:57:33, MED 0, localpref 100
                       AS path: 3333 4608 131107 I, validation-state: verified
                                           > to 193.0.19.254 via xe-1/3/0.0
```

## IOS

```
rpki-rtr>show ip bgp 202.125.96.0/24
BGP routing table entry for 202.125.96.0/24, version 70470025
Paths: (2 available, best #2, table default)
   Not advertised to any peer
   Refresh Epoch 1
       3333 1273 4637 1221 4608 131107
       193.0.19.254 from 193.0.3.5 (193.0.0.56)
       Origin IGP, localpref 110, valid, external
       Community: 83449328 83450313
       path 287058B8 RPKI State valid
```
## Commands

<table>
<thead>
<tr>
<th>Command (Junos)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show validation session detail</td>
<td>Check session status of cache validator server</td>
</tr>
<tr>
<td>show validation statistics</td>
<td>Statistics on valid/invalid prefixes</td>
</tr>
<tr>
<td>show validation database</td>
<td>Full validation database</td>
</tr>
<tr>
<td>show route protocol bgp validation-state valid/invalid/unknown</td>
<td>Find valid/invalid/unknown routes</td>
</tr>
</tbody>
</table>
!Caution!

Exception to IOS Thread:
Frame pointer 0x7F3A8AA51EE0, PC = 0x8B04A0A

UNIX-EXT-SIGNAL: Segmentation fault(11), Process = BGP Router
-Traceback: 1#270a78a13c82800fb446b5d32a66d575 :400000+4A4ADA :380000-73A8568
000000+4980EA :400000+4A6ADD :400000+496ED5

Fastpath Thread backtrace:
-Traceback: 1#270a78a13c82800fb446b5d32a66d575 c:7F3B7C28C000+B0000

Auxiliary Thread backtrace:
-Traceback: 1#270a78a13c82800fb446b5d32a66d575 pthread:7F3B774E0000+A7C9

RAX = 0000000000000000 RBX = 00007F3A8A52A0A
RCX = 0000F30F00000000 RDX = 0000000000000000
RSI = 0000000000000000 RBP = 00007F3A8A51FE0
RSI = 0000000000000000 RDI = 00007F3A8A51FE0
R8 = 0000000000000000 R9 = 00007F3A8A51FE0
R10 = 0000000000000000 R11 = 0000000000000000
R12 = 0000000000000000 R13 = 0000000000000000
R14 = 0000000000000000 R15 = 0000000000000000
RPL = 0000000000000000 RIP = 0000000000000000
CS = 0033 FS = 0000 GS = 0000
ST0 = 0000 0000000000000000 ST1 = 0000 0000000000000000
ST2 = 0000 0000000000000000 ST3 = 0000000000000000
ST4 = 0000 0000000000000000 ST5 = 0000000000000000
ST6 = 0000 0000000000000000 ST7 = 0000000000000000
X86CW = 037F X86SW = 0000 X86TG = 0000 X86TP = 0000
X86IP = 0000000000000000 X86DP = 0000000000000000
XMM0 = A8F718A3A7F0000982598A3A7F0000

20:34 BDT Mon Ng.
BGP Table:
-Traceback: 1#270a78a13c82800fb446b5d32a66d575 :400000+4A4ADA :380000-73A8568
000000+4980EA :400000+4A6ADD :400000+496ED5

Fastpath Thread backtrace:
-Traceback: 1#270a78a13c82800fb446b5d32a66d575 c:7F3B7C28C000+B0000

Auxiliary Thread backtrace:
-Traceback: 1#270a78a13c82800fb446b5d32a66d575 pthread:7F3B774E0000+A7C9

RAX = 0000000000000000 RBX = 00007F3A8A52A0A
RCX = 0000F30F00000000 RDX = 0000000000000000
RSI = 0000000000000000 RBP = 00007F3A8A51FE0
RSI = 0000000000000000 RDI = 00007F3A8A51FE0
R8 = 0000000000000000 R9 = 00007F3A8A51FE0
R10 = 0000000000000000 R11 = 0000000000000000
R12 = 0000000000000000 R13 = 0000000000000000
R14 = 0000000000000000 R15 = 0000000000000000
RPL = 0000000000000000 RIP = 0000000000000000
CS = 0033 FS = 0000 GS = 0000
ST0 = 0000 0000000000000000 ST1 = 0000 0000000000000000
ST2 = 0000 0000000000000000 ST3 = 0000000000000000
ST4 = 0000 0000000000000000 ST5 = 0000000000000000
ST6 = 0000 0000000000000000 ST7 = 0000000000000000
X86CW = 037F X86SW = 0000 X86TG = 0000 X86TP = 0000
X86IP = 0000000000000000 X86DP = 0000000000000000
XMM0 = A8F718A3A7F0000982598A3A7F0000
Testbed

- **Cisco (hosted by the RIPE NCC)**
  - Public Cisco router: rpki-rtr.ripe.net
  - Telnet username: ripe / No password

- **Juniper (hosted by Kaia Global Networks)**
  - Public Juniper routers: 193.34.50.25, 193.34.50.26
  - Telnet username: rpki / Password: testbed
Configuration - Reference Link

• Cisco

• Juniper
Thanks