RPKI
Resource Public Key Infrastructure
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?
<table>
<thead>
<tr>
<th>AS Path</th>
<th>2001:DB8::/32</th>
<th>65551</th>
<th>65550</th>
<th>65549</th>
<th>i</th>
<th>VALID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2001:DB8::/32</td>
<td>65552</td>
<td>65536</td>
<td>i</td>
<td></td>
<td>INVALID</td>
</tr>
</tbody>
</table>

Send a packet to 2001:DB8::1

I have 2001:DB8::/32

I have 2001:DB8::/32

I have 2001:DB8::/32
RPKI Deployment

Phase 2
Path Validation

Phase 1
Origin Validation

I have 2001:DB8::/32

Send a packet to 2001:DB8::1

65551
65550
65549

65552
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
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
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
APNIC
ARIN
APNIC
LACNIC
NIR
NIR
ISP
ISP
ISP
ISP
ISP
ISP

Trust Anchor Certificate

Issued Certificates match allocation actions

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

Validated RPKI Cache  RPKI to RTR protocol
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

Prefix: 10.0.0.0/16
ASN: 65420

<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>/18</td>
</tr>
<tr>
<td>VALID</td>
<td>AS65420</td>
<td>10.0.128.0/17</td>
<td>/18</td>
</tr>
<tr>
<td>INVALID</td>
<td>AS65421</td>
<td>10.0.0.0/16</td>
<td>/18</td>
</tr>
<tr>
<td>INVALID</td>
<td>AS65420</td>
<td>10.0.10.0/24</td>
<td>/18</td>
</tr>
<tr>
<td>UNKNOWN</td>
<td>AS65430</td>
<td>10.0.0.0/8</td>
<td>/18</td>
</tr>
</tbody>
</table>
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

- `{bgp4}` Routers validate updates from other BGP peers
- `{rtr}` Caches feeds routers using RTR protocol with ROA information
- `{rsync}` Caches retrieves and cryptographically validates certificates & ROAs from repositories
Phase I - Publishing ROA

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

- Show available prefix for which you can create ROA

### BGP Route Validity

<table>
<thead>
<tr>
<th>Origin AS</th>
<th>Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>45192</td>
<td>2001:df2:ee01::/48</td>
</tr>
<tr>
<td>45192</td>
<td>202.125.97.0/24</td>
</tr>
<tr>
<td>131107</td>
<td>2001:df2:ee00::/48</td>
</tr>
<tr>
<td>131107</td>
<td>202.125.96.0/24</td>
</tr>
<tr>
<td>135533</td>
<td>61.45.248.0/24</td>
</tr>
<tr>
<td>135540</td>
<td>61.45.248.0/24</td>
</tr>
</tbody>
</table>
Phase I - Publishing ROA

ROA Configuration

<table>
<thead>
<tr>
<th>Origin ASN</th>
<th>Prefix</th>
<th>Max Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>131107</td>
<td>202.125.96.0/24</td>
<td>24</td>
</tr>
<tr>
<td>131107</td>
<td>2001:df2:ee00::/48</td>
<td>48</td>
</tr>
</tbody>
</table>

Certified Resources

- 61.45.248.0/21
- 202.125.96.0/23
- 203.30.127.0/24
- 2001:DF0:A::/48
- 2001:DF2:EE00::/47
- 2406:6400::/32
Phase I - Check your ROA

```
# whois -h whois.bgpmon.net 2001:df2:ee00::/48

Prefix: 2001:df2:ee00::/48
Prefix description: APNICTRAINING-DC
Country code: AU
Origin AS: 131107
Origin AS Name: ASN for APNICTRAINING LAB DC
RPKI status: ROA validation successful
First seen: 2016-06-30
Last seen: 2017-01-03
Seen by #peers: 160
```
Phase I - Check your ROA

# whois -h whois.bgpmon.net " --roa 131107 2001:df2:ee00::/48"

0 - Valid
------------------------
ROA Details
------------------------
Origin ASN:       AS131107
Not valid Before: 2016-09-07 02:10:04
Not valid After:  2020-07-30 00:00:00  Expires in 3y208d1h39m28.7999999821186s
Trust Anchor:    rpki.apnic.net
Prefixes:         2001:df2:ee00::/48 (max length /48)                  202.125.96.0/24
(max length /24)
Phase II - RPKI Validator

• Two options:

A. RIPE NCC RPKI Validator
   • https://www.ripe.net/manage-ips-and-asns/resource-management/certification/tools-and-resources

B. Dragon Research Labs RPKI Toolkit
   • https://github.com/dragonresearch/rpki.net
Phase II - RPKI Validator

A. RIPE NCC RPKI Validator

• Download RPKI Validator


• Installation

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

A. RIPE NCC RPKI Validator

http://rpki-validator.apnictraining.net:8080/
Phase II - RPKI Validator

B. Dragon Research Labs RPKI Toolkit

• Installation process in Ubuntu Xenial 16.04
  • https://github.com/dragonresearch/rpki.net/blob/master/doc/quickstart/xenial-rp.md

• Installation

```
# wget -q -O /etc/apt/sources.list.d/rpki.list
https://download.rpki.net/APTng/rpki.xenial.list
# wget -q -O /etc/apt/trusted.gpg.d/rpki.asc  https://download.rpki.net/APTng/apt-gpg-key.asc
# apt update
# apt install rpki-rp
```
Phase II - RPKI Validator

• B. Dragon Research Labs RPKI Toolkit

http://rpki-dragonresearch.apnictraining.net/rcynic/

rcynic summary 2017-01-03T01:07:37Z

Overview Repositories Problems All Details

Grand totals for all repositories

<table>
<thead>
<tr>
<th>Tainted by stale CRL</th>
<th>Object accepted</th>
<th>Manifest interval overruns certificate</th>
</tr>
</thead>
<tbody>
<tr>
<td>None .cer</td>
<td>55061</td>
<td></td>
</tr>
<tr>
<td>None .cer</td>
<td>55061</td>
<td></td>
</tr>
<tr>
<td>None .gbr</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>None .mft</td>
<td>55060</td>
<td>1</td>
</tr>
<tr>
<td>None .roa</td>
<td>55060</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>25</td>
<td>23503</td>
</tr>
</tbody>
</table>

Overview for repository rpki.apnic.net

<table>
<thead>
<tr>
<th>Tainted by stale CRL</th>
<th>Object accepted</th>
<th>Manifest interval over</th>
</tr>
</thead>
<tbody>
<tr>
<td>None .cer</td>
<td>752</td>
<td></td>
</tr>
<tr>
<td>None .cer</td>
<td>748</td>
<td></td>
</tr>
<tr>
<td>None .mft</td>
<td>748</td>
<td></td>
</tr>
<tr>
<td>None .roa</td>
<td>492</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>2740</td>
<td></td>
</tr>
</tbody>
</table>
Phase III - Router Configuration (JunOS)

A. JunOS:

1. Establish session with RPKI Validator

   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 202.125.96.254

2. Configure policy to tag ROA

   set policy-options policy-statement ROUTE-VALIDATION term valid from protocol bgp
   set policy-options policy-statement ROUTE-VALIDATION term valid from validation-database 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 valid then accept
   set policy-options policy-statement ROUTE-VALIDATION term invalid from protocol bgp
   set policy-options policy-statement ROUTE-VALIDATION term invalid from validation-database 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
   set policy-options policy-statement ROUTE-VALIDATION term unknown from validation-database 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

3. Push policy to the BGP neighbour

   set protocols bgp import ROUTE-VALIDATION

http://pastebin.com/50bmnv9F
Phase III - Router Configuration (IOS)

B. IOS:

1. Establish session with RPKI Validator
   
   ```
   router bgp 131107
   bgp log-neighbor-changes
   bgp rpk1 server tcp 202.125.96.46 port 8282 refresh 120
   ```

2. Configure policy to tag ROA
   
   ```
   route-map ROUTE-VALIDATION permit 18
   match rpk1 invalid
   set local-preference 90
   
   route-map ROUTE-VALIDATION permit 20
   match rpk1 not-found
   set local-preference 100
   
   route-map ROUTE-VALIDATION permit 30
   match rpk1 valid
   set local-preference 110
   ```

3. Push policy to the BGP neighbour
   
   ```
   router bgp 54500
   bgp log-neighbor-changes
   :other neighbour related configuration
   neighbor 10.1.1.2 route-map ROUTE-VALIDATION in
   ```

http://pastebin.com/p3OnWu0R
Phase III - Router Configuration (GoBGP)

C. GoBGP

1. Establish session with RPKI Validator

```
[[rpki-servers]]
[rpki-servers.config]
address = "202.125.96.46"
port = 8282
```

2. Configure policy to tag ROA

```
[[policy-defined]]
name = "AS45392-IMPORT-RPKI"
[policy-defined.statements]
  name = "valid-statement"
    [policy-defined.statements.conditions.bgp-conditions]
      rpki-validation-result = "valid"
        [policy-defined.statements.actions.bgp-actions]
          set-local-pref = 110
```

```
[[policy-defined.statements]]
name = "invalid-statement"
[policy-defined.statements.conditions.bgp-conditions]
  rpki-validation-result = "invalid"
    [policy-defined.statements.actions.bgp-actions]
      set-local-pref = 90
```

3. Push policy to the BGP neighbour

```
[global.apply-policy.config]
import-policy-list = ["AS45392-IMPORT-RPKI"]
```

http://pastebin.com/DwQbdq7A
Check your prefix

- Junos

```
rpki-junos>show route protocol bgp 202.125.96.46/24

202.125.96.0/24  *[BGP/170] 3w5d 16:57:33, MED 0, localpref 110
        AS path: 3333 4608 131107 I, validation-state: verified
        > to 193.0.19.254 via xe-1/3/0.0
```
Check your prefix

• IOS

rpki-ios>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
Check your prefix

- GoBGP

```bash
fakrul@gobgp:~$ gobgp global rib 202.125.96.0/24
```

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>AS_PATH</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>202.125.96.0/24</td>
<td>202.12.29.113</td>
<td>4608 1221 4826 131107</td>
<td>00:13:29</td>
</tr>
<tr>
<td>V*&gt;</td>
<td></td>
<td>[{Origin: i} {Med: 0} {<strong>LocalPref: 110</strong>} {Communities: 4608:11101}]</td>
<td></td>
</tr>
</tbody>
</table>
## Commands

- **Check session status of cache validator server**

<table>
<thead>
<tr>
<th>OS</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>JunOS</td>
<td><code>show validation session detail</code></td>
</tr>
<tr>
<td>IOS</td>
<td><code>show bgp ipv4 unicast rpki servers</code></td>
</tr>
<tr>
<td>GoBGP</td>
<td><code>gobgp rpki server</code></td>
</tr>
</tbody>
</table>

- **Full validation database**

<table>
<thead>
<tr>
<th>OS</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>JunOS</td>
<td><code>show validation database</code></td>
</tr>
<tr>
<td>IOS</td>
<td><code>show bgp ipv4 unicast rpki table</code></td>
</tr>
<tr>
<td>GoBGP</td>
<td><code>gobgp rpki table</code></td>
</tr>
</tbody>
</table>
Caution!
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