Schedule

09:00 - 09:30  Coffee, Tea
11:00 - 11:15  Break
13:00 - 14:00  Lunch
15:30 - 15:45  Break
17:30  End
Introductions

• Name
• Number in the list
• Experience
  - BGP Routing
  - RIPE Database and Routing Registry
  - Resource Certification
• Goals
Overview

- Internet Routing Insecurity
- BGP and Routing Basics
- Introduction to the Routing Registry
  - Routing Policy Specification Language (RPSL)
  - RPSL in Practice
  - Tools and Automation
- Introduction to the Resource Certification
  - RPKI: Setting it up
  - RPKI: Using it. Relying Party’s side. Validation
  - RPKI: Router Integration
Internet Routing Insecurity

Section 1
The Importance of the Internet

Internet has taken on an important role and facilitates nearly every aspect of modern life

- Communication
- Publishing
- Support
- Research
- Personal
- Commercial
- Governmental
- Internet of Things
Border Gateway Protocol 101

- Individual networks (Autonomous Systems) identified by number (ASN) interconnect and announce prefixes to each other

- No central “core”
- No “chain of trust” in IP allocation / assignment
- No association between ASN and IP
The State of The Global Routing

• Largely a trust-based system
  - Maximum prefix lists
  - Static prefix lists
  - IRR sourced
  - Often unfiltered
  - Often unauthenticated

• Auditing is almost impossible
Routing Incidents Types

- **Misconfiguration**
  - No malicious intentions
  - Software bugs

- **Malicious**
  - Competition
  - Claiming “unused” space

- **Targeted Traffic Misdirection**
  - Collect and/or temper with data
Routing Incidents Mitigation

Is that ASN authorised to originate that address range?

- A network should only originate its own prefix
  - How do we verify?
  - How do we avoid false advertisement?

- A transit network should filter customer prefix
  - Check customer prefix and ASN delegation
  - Transitive trust
Origin Validation

• Organisation gets their resources from the RIR
  - Allocated resource is in RIR whois database

• Organisation notifies its upstream of the prefix to be announced
  - Usually email or phone

• Upstream must check the RIR whois database before accepting prefix
  - Need to be able to authoritatively prove who owns a prefix and which ASN may announce it
External Origin Validation Tools

• Internet Routing Registry
  - Public database viewable and parsable by anyone
  - Needs validation for publishing information

• Resource Public Key Infrastructure
  - Framework for automation
  - Integration with routers
End Goal: BGP Security (BGPsec)

- Extension to BGP
- Currently an IETF Internet draft
- Implemented via a new optional non-transitive BGP path attribute that contains a digital signature

Features:
- BGP Prefix Origin Validation (using RPKI)
- BGP Path Validation
BGP and Routing Basics
Section 2
Border Gateway Protocol (BGP)

- The routing protocol of the Internet
- Routing between AS-es
- Uses AS Paths
AS-Path Prevents Loops
Control and Forwarding Planes

Routing Table

Routing Protocol

best paths

IP Packets

Forwarding Table

Routing Protocol

CONTROL

FORWARDING

Routing Protocol

IP Packets
# A Route and its Attributes

<table>
<thead>
<tr>
<th>Prefix (NLRI)</th>
<th>next hop</th>
<th>MED</th>
<th>origin</th>
<th>weight</th>
<th>Local-pref</th>
<th>AS-path</th>
<th>communities</th>
</tr>
</thead>
<tbody>
<tr>
<td>66.2.9.0/23</td>
<td>95.3.12.68</td>
<td>500</td>
<td>IGP</td>
<td>200</td>
<td>100</td>
<td>756</td>
<td>164 33</td>
</tr>
</tbody>
</table>
Route Propagation
Route Attributes Limited To

Router: weight

Local AS: local-pref
Update:
     Next-hop AS-Path

local AS + neighbour:
MED

not limited:
origin communities
Update Messages

- Withdrawn prefixes
- New prefixes
  - with attributes
- Also Keep-alive messages
Routing Tables in a Router

- Updates from peers
  - Adj-RIB-in
    - Entered manually
      - Adj-RIB-in
      - Redistribution from other protocols
    - Best path calculation
      - Routing-Table
      - Output Policy Engine
    - Updates to peers
      - Adj-RIB-out
      - FIB
      - Other protocols
      - Static prefixes
# Adj-RIB-In

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Next Hop</th>
<th>MED</th>
<th>Origin</th>
<th>Weight</th>
<th>Local Pref</th>
<th>AS-Path</th>
<th>Communities</th>
</tr>
</thead>
<tbody>
<tr>
<td>66.249.0.0/16</td>
<td>92.65.185.42</td>
<td>0</td>
<td>IGP</td>
<td>0</td>
<td>100</td>
<td>203 89 151</td>
<td></td>
</tr>
<tr>
<td>66.249.0.0/16</td>
<td>98.3.23.146</td>
<td>0</td>
<td>IGP</td>
<td>0</td>
<td>100</td>
<td>34 151</td>
<td>34:102 34:123</td>
</tr>
<tr>
<td>66.249.0.0/16</td>
<td>91.67.47.102</td>
<td>100</td>
<td>IGP</td>
<td>0</td>
<td>100</td>
<td>456 1436 151</td>
<td>456:30 1436:78</td>
</tr>
<tr>
<td>66.249.0.0/20</td>
<td>95.23.129.30</td>
<td>0</td>
<td>IGP</td>
<td>100</td>
<td>40</td>
<td>2344 151</td>
<td></td>
</tr>
<tr>
<td>198.45.16.0/21</td>
<td>81.23.45.2</td>
<td>500</td>
<td>IGP</td>
<td>0</td>
<td>100</td>
<td>3456 2119 8289</td>
<td></td>
</tr>
<tr>
<td>198.45.16.0/21</td>
<td>84.5.167.85</td>
<td>0</td>
<td>IGP</td>
<td>0</td>
<td>80</td>
<td>4561 2356 8289</td>
<td>4561:180 2356:90</td>
</tr>
<tr>
<td>198.45.16.0/20</td>
<td>82.46.10.182</td>
<td>40</td>
<td>IGP</td>
<td>0</td>
<td>200</td>
<td>341 8289</td>
<td></td>
</tr>
<tr>
<td>213.4.78.0/23</td>
<td>85.196.44.23</td>
<td>0</td>
<td>IGP</td>
<td>0</td>
<td>20</td>
<td>7895 1299</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
## BGP Entries in the Routing-Table

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Next Hop</th>
<th>MED</th>
<th>Origin</th>
<th>Weight</th>
<th>Local Pref</th>
<th>AS-Path</th>
<th>Communities</th>
</tr>
</thead>
<tbody>
<tr>
<td>66.249.0.0/16</td>
<td>98.3.23.146</td>
<td>0</td>
<td>IGP</td>
<td>0</td>
<td>100</td>
<td>34 151</td>
<td>34:102 34:123</td>
</tr>
<tr>
<td>66.249.0.0/20</td>
<td>95.23.129.30</td>
<td>0</td>
<td>IGP</td>
<td>100</td>
<td>40</td>
<td>2344 151</td>
<td></td>
</tr>
<tr>
<td>198.45.16.0/21</td>
<td>81.23.45.2</td>
<td>500</td>
<td>IGP</td>
<td>0</td>
<td>100</td>
<td>3456 2119 8289</td>
<td></td>
</tr>
<tr>
<td>198.45.16.0/20</td>
<td>82.46.10.182</td>
<td>40</td>
<td>IGP</td>
<td>0</td>
<td>200</td>
<td>341 8289</td>
<td></td>
</tr>
<tr>
<td>213.4.78.0/23</td>
<td>85.196.44.23</td>
<td>0</td>
<td>IGP</td>
<td>0</td>
<td>20</td>
<td>7895 1299</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Routing Security
# FIB - Forwarding Table

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>66.249.0.0/16</td>
<td>2</td>
</tr>
<tr>
<td>66.249.0.0/20</td>
<td>4</td>
</tr>
<tr>
<td>198.45.16.0/21</td>
<td>1</td>
</tr>
<tr>
<td>198.45.16.0/20</td>
<td>3</td>
</tr>
<tr>
<td>213.4.78.0/23</td>
<td>5</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Best Path Calculation

- Drop if own AS in AS-Path
- Prefer path with highest Weight
- Highest Local Preference
- Shortest AS-Path
- Lowest MED
Best Path Calculation - Tiebreakers

- Path with shortest next hop metric (minimum IGP cost)
- Oldest received path
- Path from lowest neighbour address
Administrative Distance

Routing Table

- Connected Interface: 0
- Static Route: 1
- eBGP: 20
- IGP: 90-120
- iBGP: 200
- Unknown: 255

FIB
## More Specific Wins

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Next Hop</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>66.249.0.0/16</td>
<td>98.3.23.146</td>
<td>2</td>
</tr>
<tr>
<td>66.249.0.0/20</td>
<td>95.23.129.30</td>
<td>4</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Traffic to 66.249.7.35?

- Interface 4
Introduction to the Routing Registry

Section 3
Why Routing Registry?

To be able to answer the question:

Is that ASN authorised to originate that address range?
Internet Routing Registry

- Number of public databases that contain routing policy information which mirror each other:
  - RIPE, APNIC, RADB, JPIRR, Level3, …
  - http://www.irr.net

- RIPE NCC operates the RIPE Routing Registry
  - Part of the RIPE Database
  - Part of the Internet Routing Registry
RIPE Database Objects

- **inetnum** ➔ IPv4 address range
- **inet6num** ➔ IPv6 address range
- **aut-num** ➔ single AS number and routing policy
- **route, route6** ➔ glue between IP address range and an AS number announcing it
- **person** ➔ contact info for other objects
- **role** ➔ group of person objects
- **maintainer** ➔ protects all other objects
Registering Routes

**inet6num:** 2001:db8::/32

- **tech-c:** LA789-RIPE
- **admin-c:** JD1-RIPE
- **mnt-by:** RIPE-NCC-HM-MNT
- **mnt-routes:** LIR-MNT

**aut-num:** AS64512

- **as-name:** GREEN-AS
- **tech-c:** LA789-RIPE
- **admin-c:** JD1-RIPE
- **mnt-by:** LIR-MNT

**route6:** 2001:db8::/32

- **tech-c:** LA789-RIPE
- **admin-c:** JD1-RIPE
- **origin:** AS64512
- **mnt-by:** LIR-MNT

Add passwords

Session passwords

0 stored password(s)

12lir

✔️
Registering Routes

**inet6num:** 2001:db8::/32
- **tech-c:** LA789-RIPE
- **admin-c:** JD1-RIPE
- **mnt-by:** RIPE-NCC-HM-MNT
- **mnt-routes:** LIR-MNT

**aut-num:** AS64512
- **as-name:** GREEN-AS
- **tech-c:** LA789-RIPE
- **admin-c:** JD1-RIPE
- **mnt-by:** AS-MNT

**route6:** 2001:db8::/32
- **tech-c:** LA789-RIPE
- **admin-c:** JD1-RIPE
- **origin:** AS64512
- **mnt-by:** END-MNT

Add passwords
- 12lir
- as999
- end72
Registering Routes

inet6num: 2001:db8::/32

tech-c: LA789-RIPE
admin-c: JD1-RIPE
mnt-by: RIPE-NCC-HM-MNT
mnt-routes: LIR-MNT

aut-num: AS64512

as-name: GREEN-AS
tech-c: LA789-RIPE
admin-c: JD1-RIPE
mnt-by: AS-MNT

route6: 2001:db8::/32

tech-c: LA789-RIPE
admin-c: JD1-RIPE
origin: AS64512
mnt-by: AS-MNT

Add passwords

Session passwords
0 stored password(s)
as999
Registering Routes

**inet6num:** 2001:db8::/32
- **tech-c:** LA789-RIPE
- **admin-c:** JD1-RIPE
- **mnt-by:** RIPE-NCC-HM-MNT
- **mnt-routes:** LIR-MNT

**aut-num:** AS64512
- **as-name:** GREEN-AS
- **tech-c:** LA789-RIPE
- **admin-c:** JD1-RIPE
- **mnt-by:** AS-MNT
- **mnt-routes:** LIR-MNT

Add passwords

**Session passwords**
0 stored password(s) [?] 12lir [+]
Registering Routes

- Creating route object
  - Sharing passwords
  - Adding other users’ maintainers to your objects

- New approach
  - For any missing authorisation, object is queued and notification is sent to the maintainer

mntner: LIR-MNT
auth: MD5-PW $1$car0J
upd-to: lir@example.com
Registering Routes

**inet6num:** 2001:db8::/32
- **tech-c:** LA789-RIPE
- **admin-c:** JD1-RIPE
- **mnt-by:** RIPE-NCC-HM-MNT
- **mnt-routes:** LIR-MNT

**route6:** 2001:db8::/32
- **tech-c:** LA789-RIPE
- **admin-c:** JD1-RIPE
- **origin:** AS64512
- **mnt-by:** LIR-MNT

**aut-num:** AS64512
- **tech-c:** LA789-RIPE
- **admin-c:** JD1-RIPE
- **mnt-by:** RIPE-NCC-HM-MNT
- **mnt-by:** AS-MNT

**mntner:** AS-MNT
- **auth:** MD5-PW $1$car0J
- **upd-to:** lir@example.com
What is a Routing Policy?

• What prefixes do you announce?
• Who are your neighbours?
  - Peers, transits and customers
• Which prefixes do you accept from them?
• What are your preferences?
aut-num Object and Routing Policy

<table>
<thead>
<tr>
<th>aut-num:</th>
<th>AS64512</th>
</tr>
</thead>
<tbody>
<tr>
<td>descr:</td>
<td>RIPE NCC Training Services</td>
</tr>
<tr>
<td>as-name:</td>
<td>GREEN-AS</td>
</tr>
<tr>
<td>tech-c:</td>
<td>LA789-RIPE</td>
</tr>
<tr>
<td>admin-c:</td>
<td>JD1-RIPE</td>
</tr>
<tr>
<td>import:</td>
<td>from AS64444 accept ANY</td>
</tr>
<tr>
<td>import:</td>
<td>from AS64488 accept ANY</td>
</tr>
<tr>
<td>export:</td>
<td>to AS64444 announce AS64512</td>
</tr>
<tr>
<td>export:</td>
<td>to AS64488 announce AS64512</td>
</tr>
<tr>
<td>mnt-by:</td>
<td>LIR-MNT</td>
</tr>
<tr>
<td>source:</td>
<td>RIPE</td>
</tr>
</tbody>
</table>
Why Publish Your Routing Policy?

• Some transit providers and IXPs (Internet Exchange Points) require it
  - They build their filters based on the routing registry

• Contributes to routing security and stability
  - Let people know about your intentions

• Can help in troubleshooting
  - Which parties are involved?
RIPE Database

• Close relation between registry information and routing policy
  - The holder of the resources knows how they should be routed

• The Routing Policy Specification Language (RPSL) originates from a RIPE Document
  - Shares attributes with the RIPE Database
Routing Registries Challenges

- Accuracy and completeness
- Not every Routing Registry is linked directly to an Internet Registry
  - Offline verification of the resource holder is needed
- Different authorisation methods
- Mirrors are not always up to date
Create a route or a route6 Object

Exercise 1
Exercise 1

- Create a `route` object for your IPv4 allocation
- Create a `route6` object for your IPv6 allocation
- List your AS Number (`aut-num`) as the origin for both objects
Routing Policy Specification Language

Section 4
Routing Policy

• A routing policy describes how a network works
  - Who do you connect with
  - Which prefixes or routes do you announce
  - Which routes do you accept from others
  - What are your preferences

• In your router, this is your BGP configuration
  - neighbours
  - route-maps
  - prefix lists
  - localpref
RPSL

- Language used by the IRRs
- Not vendor-specific
- Documented in RFC 2622
  - and RFC 2650 “Using RPSL in practice”

- Can be translated into router configuration
Objects Involved

• **route** or **route6** object
  - Connects a prefix to an origin AS

• **aut-num** object
  - Registration record of an AS Number
  - Contains the routing policy

• **Sets**
  - Objects can be grouped in sets, i.e. as-set, route-set

• **Keywords**
  - “ANY” matches every route
Notation

- AS Numbers are written as ASxxx
- Prefixes are written in CIDR notation
  - i.e. 193.0.4.0/24
- Any value can be replaced by a list of values of the same type
  - AS1 can be replaced by “AS1 AS2 AS3”
- You can reference a set instead of a value
  - “…announce AS1” or “…announce as-myname”
Import and Export Attributes

- You can document your routing policy in your aut-num object in the RIPE Database:
  - Import lines describe what routes you accept from a neighbour and what you do with them
  - Export lines describe which routes you announce to your neighbour
Traffic Direction vs Announcement

**aut-num:** AS1

**import:** from AS2 accept AS2

**export:** to AS2 announce AS1

AS1 accepting those prefixes **from** AS2 that originate in AS2 so that the **outbound** traffic for AS2 can go **towards** the AS2

AS1 announcing prefixes (originating in AS1) **to** AS2, so that the **incoming** traffic for AS1 can flow **away** from the AS2
Example: You Are Downstream

Internet

AS2

Transit provider

AS1

You

```
aut-num: AS1
import: from AS2 accept ANY
export: to AS2 announce AS1
```
Example: You Are Upstream

**Internet**

**AS1**

**You**

**AS3**

**Downstream customer**

**aut-num: AS1**

**import: from AS3 accept AS3**

**export: to AS3 announce ANY**
Example: Peering

**aut-num**: AS1
**import**: from AS4 accept AS4
**export**: to AS4 announce AS1
**Example: Summary**

- **AS1**
  - `aut-num: AS1`
  - `import: from AS2 accept ANY`
  - `import: from AS3 accept AS3`
  - `import: from AS4 accept AS4`
  - `export: to AS2 announce AS1 AS3`
  - `export: to AS3 announce ANY`
  - `export: to AS4 announce AS1 AS3`

- **AS2**
  - **Transit provider**

- **AS3**
  - **Downstream**

- **AS4**
  - **Peer**

- **Internet**
Building an aut-num Object

aut-num: AS2
import: from AS1 accept AS1
export: to AS1 announce AS2

aut-num: AS1
export: to AS2 announce AS1
import: from AS2 accept AS2
import: from AS3 accept ANY
export: to AS3 announce AS1

aut-num: AS3
export: to AS1 announce ANY
import: from AS1 accept AS1
RPSLng

- RPSL is older than IPv6, the defaults are IPv4
- IPv6 was added later using a different syntax
- You have to specify that it’s IPv6

```
mp-import:  afi ipv6.unicast from AS201 accept AS201
mp-export:  afi ipv6.unicast to AS201 announce ANY
```

- More information in RFC 4012 RPSLng
Retrieving Information from the IRR

Exercise 2
A Look at the Real World

• Have a look at AS 3333 in the RIPE Database
  - Which prefixes would you accept from AS 3333 if it was your customer?

• Remember to use the real database!

• Optionally verify the results using the tools at http://stat.ripe.net
RPSL in Practice

Section 5
### Example Routing Policy

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>aut-num:</strong></td>
<td>AS99</td>
</tr>
<tr>
<td><strong>as-name:</strong></td>
<td>SMALL-ISP-EU</td>
</tr>
<tr>
<td><strong>descr:</strong></td>
<td>My network</td>
</tr>
<tr>
<td><strong>remarks:</strong></td>
<td>*** Transit via 101 ***</td>
</tr>
<tr>
<td><strong>import:</strong></td>
<td>from AS101 accept ANY</td>
</tr>
<tr>
<td><strong>export:</strong></td>
<td>to AS101 announce AS99 AS201 AS202</td>
</tr>
<tr>
<td><strong>remarks:</strong></td>
<td>*** Transit via 102 ***</td>
</tr>
<tr>
<td><strong>import:</strong></td>
<td>from AS102 accept ANY</td>
</tr>
<tr>
<td><strong>export:</strong></td>
<td>to AS102 announce AS99 AS201 AS202</td>
</tr>
<tr>
<td><strong>remarks:</strong></td>
<td>*** AS201 is a customer ***</td>
</tr>
<tr>
<td><strong>import:</strong></td>
<td>from AS201 accept AS201</td>
</tr>
<tr>
<td><strong>export:</strong></td>
<td>to AS201 announce ANY</td>
</tr>
<tr>
<td><strong>remarks:</strong></td>
<td>*** AS202 is a customer ***</td>
</tr>
<tr>
<td><strong>import:</strong></td>
<td>from AS202 accept AS202</td>
</tr>
<tr>
<td><strong>export:</strong></td>
<td>to AS202 announce ANY</td>
</tr>
</tbody>
</table>
Using as-set

- Adding and removing customers can become time consuming
- Create a set to list them all at once

```
as-set:    AS-SMALLISP
descr:     Customers’ ASNs of a small ISP
members:   AS99
members:   AS201
members:   AS202
```

- And use that to describe your policy

```
export:    to AS101 announce AS-SMALLISP
export:    to AS102 announce AS-SMALLISP
```
Use Keywords for as-sets

- **as-set:** AS4:AS-CUSTOMERS
- **members:** AS7, AS5, AS8

- **aut-num:** AS4
- **export:** to AS3 announce AS4 AS4:AS-customers
- **export:** to AS4:AS-CUSTOMERS announce ANY
- **import:** from AS4:AS-CUSTOMERS accept PeerAS

- PeerAS means:
  - from AS5 accept AS5
  - from AS7 accept AS7
  - from AS8 accept AS8
Indicating Your Preferences

- BGP uses the "localpref" to influence which received routes you want to prefer
- In RPSL you can use the "pref" action on your import attributes
- Important: lower value means more preferred!

```
import: from AS101 action pref=20;
   accept ANY

import: from AS102 action pref=30;
   accept ANY
```
Describing AS Path Prepending

- AS Path prepending is used to influence other people’s preferences
- Prepending can also be notated in RPSL using another action statement:

```export: to AS102 action aspath.prepend (AS99, AS99); announce AS-SMALLISP```
Routing Security

Building an aut-num Object

Internet

AS5

aut-num: AS5

import: from AS1 accept AS1

export: to AS1 announce ANY

AS1

import: from AS4 action pref=80;

accept ANY

export: to AS4 announce AS1

import: from AS5 action pref=90;

accept ANY

import: from AS5 action pref=70;

accept AS5

export: to AS5

action aspath.prepend (AS1, AS1);

announce AS1

AS4

aut-num: AS4

import: from AS1 accept AS1

export: to AS1 announce ANY
MED (Multi Exit discriminator)

- **Multiple Exit Discriminator**
  - Differentiates connections to same peer
  - “Which inbound connection do I prefer?”
  - Doesn’t go beyond neighbour

- **Local Pref has precedence over MED**
  - To honour your neighbours MED:
  - Don’t set different prefs
Example: Using MED

```
export: to AS4
10.0.0.4 at 10.0.0.1
action med=1000;
announce AS99

export: to AS4
10.0.0.5 at 10.0.0.2
action med=2000;
announce AS99
```
Communities

- Optional tags
  - Can go through many peers
- Can be used for advanced filtering
- Not a routing parameter
- Enables customers to control their own routing policy
  - Publish your communities, and what you do with them
  - Filter incoming announcements accordingly
Example: Using Communities

• Set a community

```python
import: from AS6
action community = { 99: 100 };
accept AS6
```

• Append a community

```python
import: from AS7
action community.append(99:51);
accept AS7
```

```python
export: to AS3
action community = { 99: 100 };
announce ANY
```

• Delete a community

```python
import: from AS201 action community.delete(99:100); accept AS201
```
Example: Communities Filtering

**import:**
from AS21
accept AS6 AND
community.contains = (21:32)

**import:**
from AS17
accept community(68:2)

**import:**
from AS1:AS-CUSTOMERS
accept PeerAS AND
community.contains (202:3)

**export:**
to AS3
announce AS1:AS-CUST AND
community == {1:113}

**export:**
to AS1:AS-PEERS
announce ANY AND
community.contains (1:75)
AS Path Regular Expressions

• You can use regular expressions in your filters
  - they are always enclosed in “< >”
  - import: from AS201 accept <^AS201+$>

• Uses the standard posix notation
  - “^” start of path
  - “$” end of path
  - “*” zero or more
  - “+” one or more
  - “?” zero or one
Literal Prefixes

• Instead of AS Numbers you can use prefixes
  - import: from AS2121 accept {193.0.24.0/21}

• Operators can be used to define ranges
  - "^−" all more specifics excluding the prefix itself
  - "^+" all more specifics including the prefix itself
  - "^n" all routes of length n in this prefix
  - "^n-m" all routes of length n to length m
Using a route-set

- Groups literal prefixes
- Can include other route-sets and even ASNs

```
route-set: RS-BAR
descr: All ASNs of a small ISP
members: 5.0.0.0/8^+, 30.0.0.0/8^24-32
members: rs-foo^+
members: AS2
```

- And use that to describe/simplify your policy

```
export: to AS101 announce RS-BAR
```
Routing Security

**Default Routes**

- Next to import and export there can also be a default line to describe your default policy

```
export: to AS99 announce AS201
import: from AS202 accept AS202
export: to AS202 announce AS201
default: to AS99 action pref=150
```

- Instead of all routes, you can also announce a default route

```
export: to AS101 announce RSL-BAR
```
### The Simplified Object

<table>
<thead>
<tr>
<th>aut-num:</th>
<th>AS99</th>
</tr>
</thead>
<tbody>
<tr>
<td>as-name:</td>
<td>SMALL-ISP-EU</td>
</tr>
<tr>
<td>descr:</td>
<td>My network</td>
</tr>
<tr>
<td>remarks:</td>
<td>*** Announcements are grouped ***</td>
</tr>
<tr>
<td>import:</td>
<td>from AS101 accept ANY</td>
</tr>
<tr>
<td>export:</td>
<td>to AS101 announce AS-SMALLISP</td>
</tr>
<tr>
<td>import:</td>
<td>from AS102 accept ANY</td>
</tr>
<tr>
<td>export:</td>
<td>to AS102 announce AS-SMALLISP</td>
</tr>
<tr>
<td>remarks:</td>
<td>*** My Customers are grouped ***</td>
</tr>
<tr>
<td>import:</td>
<td>from AS99:Customers accept PEERAS</td>
</tr>
<tr>
<td>export:</td>
<td>to AS99:Customers announce ANY</td>
</tr>
</tbody>
</table>
Describing Your Routing Policy

Exercise 3
Modifying aut-num Object

- Take the scenario as presented

- In the TEST RIPE Database update your AS (aut-num), adding import, export, mp-import, mp-export attributes to describe your policy towards these neighbours
Making Life Easier

• There are a lot of tools around that use information in the Routing Registry

• Some can generate complete router configurations like the IRRToolset

• Most are open source tools
  - You can modify them to your needs
  - Some are not very well maintained
Example Tools

- IRRToolkit (written in C++)
  - http://irrtoolset.isc.org/

- Rpsltool (perl)
  - http://www.linux.it/~md/software

- IRR Power Tools (PHP)
  - http://sourceforge.net/projects/irrpt/

- BGPQ3 (C)

- Filtergen (Level 3)
  - whois -h filtergen.level3.net RIPE::ASxxx

- IRR Explorer (web)
  - http://irrexplorer.nlnog.net
Building Your Own

• A couple of things to keep in mind
  - The RIPE Database has limits on the number of queries you can do per day
  - Query flags or output format can change over time

• Instead of the whois interface, you can use the RESTful API for the RIPE Database
  - Uses XML or JSON for output
  - See https://ripe.net/developer
  - Also visit https://labs.ripe.net for more information
Getting the Complete Picture

• Automation relies on the IRR being complete
  - Not all resources are registered in an IRR
  - Not all information is correct

• Small mistakes can have a big impact

• Check your output before using it
  - Be prepared to make manual overrides

• Help others by documenting your policy
• You can compare the Routing Registry and the Internet routing table using http://stat.ripe.net
Using a Tool

Exercise 4
Using Filtergen

- Use a tool to retrieve the same information from the exercise 2

- "whois -h filtergen.level3.net RIPE::AS3333"
  - Syntax is "RIPE::" followed by the AS you want information about

- Do you get the same answers?
  - What is the result of AS-RIPENCC?
  - If you have time, try AS-TELIANET
Questions
Introduction the the RPKI

Section 7
Why RPKI?

To be able to answer the question:

Is that ASN authorised to originate that address range?
RPKI and IRR

• Why yet another system?
  - Lots of Routing Registries
  - Not all mirroring each other
  - Different levels of trustworthiness and authentication

• RPKI replaces IRR or lives side by side?
  - Side by side: different advantages
  - Security, almost real time, simple interface: RPKI
  - More info in: IRR
The Advantages 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
The announcers side
Section 8
Resource Certificates

- RIPE NCC issues digital certificates
  - To LIRs
  - To PI end users

- Upon request

- Certificate lists all resources held by the member
Which Resources Are Certified?

- Everything for which we are 100% sure who the holder is
  - Provider Aggregatable (PA) addresses
  - Provider Independent (PI) addresses
    - marked as LIR “Infrastructure”
    - for which we have a contract (Policy 2007-01)
  - Legacy Resources
RPKI Chain of Trust

- RIPE NCC holds self-signed root certificate for all resources they have in the registry
  - Signed by the root’s private key

- The root certificate is used to sign all certificates for members listing their resources
  - Signed by the root’s private key
RPKI Chain of Trust

**RIPE NCC’s Root Certificate**
- All RIPE NCC’s resources
- Root public key
- Signature

Root’s (RIPE NCC) private key

**LIR’s Certificate**
- All member’s resources
- LIR’s public key
- Signature

LIR’s private key
ROA (Route Origin Authorisation)

- LIRs can use their certificate to create a ROA for each of their resources (IP address ranges)
  - Signed by the LIR’s private key

- ROA states
  - Address range
  - Which AS this is announced from (freely chosen)
  - Maximum length (freely chosen)

- You can have multiple ROAs for an IP range

- ROAs can overlap
ROA Chain of Trust

RIPE NCC’s Root Certificate
- All RIPE NCC’s resources
- Root public key
- Signature

LIR’s Certificate
- All member’s resources
- LIR’s public key
- Signature

Root’s (RIPE NCC) private key

ROA
- IP Range
- AS Number: AS123
- Max Length: /24
- Signature

Routing Security
Example: ROA

ROA

193.0.24.0/21
AS2121
Max Length: _

193.0.24.0/21

193.0.24.0/22
193.0.30.0/23
Example: ROA

ROA

193.0.24.0/21
AS2121
Max Length: /23

193.0.24.0/21
193.0.24.0/22
193.0.28.0/22
193.0.28.0/23
193.0.26.0/23
193.0.28.0/23
193.0.30.0/23
Example: ROA

ROA

193.0.24.0/21
AS2121
Max Length: _

193.0.24.0/22

193.0.24.0/23
AS2121
Max Length: /24

193.0.28.0/22

193.0.30.0/23
AS2121
Max Length: _

/23
/23
/23
/23
/23
/23
/24
/24
/24
/24
/24
/24
/24
/24
/24
/24
Public Repository

- RIPE NCC maintains a Certificate Repository containing
  - All the certificates
  - All the public keys
  - All the ROAs
Enabling Access in the LIRPortal

Edit Contact

First name
Andrzej

Last name
Wolski

Email
awolski@ripe.net

Status
Active

Comments
trainer

What this user is entitled to do:

- Manage contacts and access all RIPE NCC services
- Access all RIPE NCC services
- Make payments and manage billing information

[Save changes]
Setting up Certificate Authority

RIPE NCC Certification Service Terms and Conditions

Introduction

Article 1 – Definitions
In the Terms and Conditions, the following terms shall be understood to have the meanings assigned to them below:
RIPE NCC – Réseaux IP Européens Network Coordination Centre, a membership association under Dutch law, operating from its registered office in Amsterdam, the Netherlands.
Certificate Holder – A natural person or a legal entity that has entered into an agreement regarding the registration of their resources either with a sponsoring LIR or with the

By clicking on 'I accept' below you confirm that that you have read, understood and agree to the RIPE NCC Certification Service Terms and Conditions.

I accept. Create my Certificate Authority

https://localcert.ripe.net
Managing ROAs

<table>
<thead>
<tr>
<th>Origin AS</th>
<th>Prefix</th>
<th>Current Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS12654</td>
<td>2001:7fb:fe01::/48</td>
<td>UNKNOWN</td>
</tr>
<tr>
<td>AS12654</td>
<td>2001:7fb:fe0c::/48</td>
<td>UNKNOWN</td>
</tr>
<tr>
<td>AS12654</td>
<td>2001:7fb:fe0f::/48</td>
<td>UNKNOWN</td>
</tr>
<tr>
<td>AS12654</td>
<td>2001:7fb:ff00::/48</td>
<td>UNKNOWN</td>
</tr>
<tr>
<td>AS12654</td>
<td>2001:7fb:ff01::/48</td>
<td>UNKNOWN</td>
</tr>
<tr>
<td>AS12654</td>
<td>2001:7fb:ff02::/48</td>
<td>UNKNOWN</td>
</tr>
<tr>
<td>AS12654</td>
<td>2001:7fb:ff03::/48</td>
<td>UNKNOWN</td>
</tr>
</tbody>
</table>
RPKI
Relying Party’s side
Section 10
Validator

- The validator of the client can access RIPE NCC’s Repository with all the certificates, public keys, ROAs

- It downloads everything and then performs validation, checking whether the certificates and ROAs are valid. Then it constructs a list of valid ROAs, which is its “validated cache”
ROA Chain of Trust

RIPE NCC’s Root Certificate

- All RIPE NCC’s resources
- Root public key
- Signature

LIR’s Certificate

- All member’s resources
- LIR’s public key
- Signature

Root’s (RIPE NCC) private key

ROA

- IP Range
- AS Number: AS123
- Max Length: /24
- Signature
Validated Cache

RIPE NCC’s Repository

Certificates

ROAs

Certificate

Validator

Validated cache

ROA

Validated ROAs only

at the Relying Party’s site
Invalid ROAs

- Invalid ROAs are simply not included in the list of validated ROAs when the validator of the client computes them

- Reasons for a ROA to be invalid
  - The signing certificate or key pair has expired or has been revoked
  - It does not validate back to a configured trust anchor
  - The LIR’s resource has been returned to the RIPE NCC
Modifying the Validated Cache

- The RIPE NCC Validator allows you to manually override the validation process.

- Adding an ignore filter will ignore all ROAs for a given prefix.
  - The end result is the validation state will be “unknown”.

- Creating a whitelist entry for a prefix and ASN will locally create a valid ROA.
  - The end result is the validation state becomes “valid”.
The Relying Party’s router can connect and download the cache from the validator.

- Router can then compare any BGP announcements to the list of valid ROAs in the validated cache.
BGP Verification

Client (ISP, Relying Party)

Validator

Validated cache

ROA
191.71.8.0/24
AS93

Validate ROAs only

191.71.8.0/24
origin: AS93

compare

AS14
Results of BGP Verification

• valid
  - There is a ROA in the validated cache that matches the BGP announcement of the peer, size matches too

• unknown
  - There is no ROA for that prefix in the cache

• invalid
  - There is a ROA for the prefix, but for a different AS
  - The size doesn’t match
ROA vs Announcement

• Invalid ROA
  - The ROA in the repository cannot be validated by the client (ISP) so it is not included in the validated cache

• Invalid BGP announcement
  - There is a ROA in validated cache for that prefix but for a different AS.
  - Or the max length doesn’t match.

• If no ROA in the cache then announcement is “unknown”
You are in control

• 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)
Download the Validator

- http://www.ripe.net/certification

- No Installation required
  - Unzip the package
  - Run the program: rpki-validator.sh start

- Interface available on localhost port 8080
The Web Interface

Quick Overview of BGP Origin Validation

Trust Anchors → ROAs → Ignore Filters → Whitelist → Router

Trust Anchors are the entry points used for validation in any Public Key Infrastructure (PKI) system. This validator is intended for the validation of Resource PKI (RPKI) systems. It is pre-configured with Trust Anchors for all the RIRs who are running such systems now.

If you would like to add or change the Trust Anchors that are used by this validator, please see the README.txt file for details.
## Trust Anchors

![RPKI Validator - Configured Trust Anchors](image)

<table>
<thead>
<tr>
<th>Trust anchor</th>
<th>Processed Items</th>
<th>Expires in</th>
<th>Last updated</th>
<th>Next update in</th>
</tr>
</thead>
<tbody>
<tr>
<td>APNIC RPKI Root</td>
<td>1356</td>
<td>4 years and 2 months</td>
<td>7 minutes ago</td>
<td>3 hours</td>
</tr>
<tr>
<td>ARIN Test Lab</td>
<td>88</td>
<td>1 year and 2 months</td>
<td>8 minutes ago</td>
<td>3 hours</td>
</tr>
<tr>
<td>AfrINIC RPKI Root</td>
<td>80</td>
<td>4 years and 7 months</td>
<td>8 minutes ago</td>
<td>3 hours</td>
</tr>
<tr>
<td>LACNIC RPKI Root</td>
<td>216</td>
<td>10 months and 3 weeks</td>
<td>8 minutes ago</td>
<td>3 hours</td>
</tr>
<tr>
<td>RIPE NCC RPKI Root</td>
<td>3576</td>
<td>4 years and 9 months</td>
<td>7 minutes ago</td>
<td>3 hours</td>
</tr>
</tbody>
</table>
Validated Cache

Validated ROAs from APNIC RPKI Root, ARIN Test Lab, AfriNIC RPKI Root, LACNIC RPKI Root, RIPE NCC RPKI Root.

<table>
<thead>
<tr>
<th>ASN</th>
<th>Prefix</th>
<th>Maximum Length</th>
<th>Trust Anchor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10.0.1.0/24</td>
<td>24</td>
<td>ARIN Test Lab</td>
</tr>
<tr>
<td>1</td>
<td>192.168.1.0/24</td>
<td>24</td>
<td>ARIN Test Lab</td>
</tr>
<tr>
<td>1</td>
<td>61.45.250.0/23</td>
<td>23</td>
<td>APNIC RPKI Root</td>
</tr>
<tr>
<td>1</td>
<td>61.45.250.0/23</td>
<td>23</td>
<td>APNIC RPKI Root</td>
</tr>
<tr>
<td>21</td>
<td>10.4.0.0/16</td>
<td>16</td>
<td>ARIN Test Lab</td>
</tr>
<tr>
<td>22</td>
<td>10.255.1.0/24</td>
<td>24</td>
<td>ARIN Test Lab</td>
</tr>
<tr>
<td>42</td>
<td>2001:678:3::/48</td>
<td>48</td>
<td>RIPE NCC RPKI Root</td>
</tr>
<tr>
<td>42</td>
<td>194.0.17.0/24</td>
<td>24</td>
<td>RIPE NCC RPKI Root</td>
</tr>
<tr>
<td>174</td>
<td>89.207.56.0/21</td>
<td>21</td>
<td>RIPE NCC RPKI Root</td>
</tr>
</tbody>
</table>
Creating a Whitelist

Insert the prefix and click “Add”

This locally creates a valid (but fake) ROA
BGP Preview

- The validator downloads a copy of the RIS
  - Allows you to get a hint of what would happen
  - RIS view might be different from your routing table
BGP Preview Detail

The image shows a screenshot of the RPKI Validator, specifically the BGP Preview tab. The screen displays a list of ASNs and their corresponding prefixes. The details of a specific prefix (2001:468::/32) are highlighted, indicating it is marked as INVALID. The page contains a search bar and a table showing various entries, with a focus on the validity of the prefixes.
RPKI Quiz
Exercise 5
RPKI
Router Integration
Section 11
Exporting the Validated Cache

- **Router sessions**
  - Validator listens on 8282 for RPKI-RTR Protocol
  - Routers can connect and download the cache

- **Export function**
  - Allows you to download a CSV with the cache
  - Can be integrated with your internal workflow
  - Use for statistics or spotting anomalies
RPKI Support in Routers

- **RPKI** and **RPKI-RTR** are an IETF standards
  - All router vendors can implement it
- **Cisco** support:
  - XR 4.2.1 (CRS-x, ASR9000, c12K) / XR 5.1.1 (NCS6000, XRv)
  - XE 3.5 (C7200, c7600, ASR1K, CSR1Kv, ASR9k, ME3600…)
  - IOS15.2(1)S
- **Juniper** has support since version 12.2
- **Alcatel Lucent** has support since SR-OS 12.0 R4
- **Quagga** has support through BGP-SRX
- **BIRD** has support for ROA but does not do RPKI-RTR
Public Testbeds

• Cisco (hosted by the RIPE NCC)
  - Telnet to rpki-rtr.ripe.net
  - User: ripe, no password

• Juniper (hosted by Kaia Global Networks)
  - Telnet to 193.34.50.25 or 193.34.50.26
  - Username: rpki, password: testbed

http://www.ripe.net/certification
Community Activity

- Open source RPKI Tools
  - rpki.net
- SURFnet RPKI Dashboard
  - rpki.surfnet.nl
- BGPMon Route Monitoring
  - bgpmon.net/services/route-monitoring/
- RIPE NCC Github
  - github.com/RIPE-NCC
Questions
RIPE NCC Academy

Graduate to the next level!

http://academy.ripe.net
Feedback

http://www.ripe.net/training/rs/survey
Follow us!

@TrainingRIPECC
The End!