Implementation of RPKI and IRR filtering on the AMS-IX platform

Stavros Konstantaras
NOC Engineer
RIPE EDUCA 2018
Agenda

• AMS-IX Route Servers
  – Architecture
  – Features
• Filtering
  – IRRdb
  – RPKI
  – BGP Communities
• Real-life examples/problems
Route Servers in IXPs

- Reduces the number of BGP connections per member/customer
- Manage only your most important peers, let the route server do the rest
- Send and receive routes from day one
- Use it as a backup

Figure retrieved from IMC238 (Richter et al): “Peering at Peerings: On the Role of IXP Route Servers”, 2014
AMS-IX Route Servers

• 4 BIRD instances in high spec servers

• 764 IPv4 Peers & 620 IPv6 peers
  – Prefixes received: 267635 IPv4 || 41037 IPv6
  – Prefixes Sent: 190915 IPv4 || 28175 IPv6
  – Average Prefixes per peer: 375 IPv4 || 72 IPv6

• Neutral prefix handling
  – Local_pref = 100
AMS-IX RS architecture
AMS-IX RS features

• Receive Prefixes / Propagate best paths
• Ensure peering rules are satisfied
• Perform IRR and RPKI based filtering
  – The 4 filtering modes
• Perform community-based filtering
• Expose info to lg and notification system*

*WIP
Tools used in implementation

• External tools
  – whois (to read member policy)
  – bgpq3 (for resolving AS-SETs)
  – RIPE validator (to validate announcements)
• Lots of internal tools
  – rs_configurator.pl
  – rs_prefixes_api
  – …
Filtering
Prefix filtering in AMS-IX

- Basic (ingress)
  - Bogons & Martians
  - Default route
  - RFC 1918 ranges
- Extended (egress)
  - The 4 peering modes
Where is applied
Peering rules (ingress)

- Not accepted prefixes:
  - Bogons & Martians
  - AMS-IX prefixes
  - Prefixes with AS path length > 64
  - The first AS in AS path is **not** the customer one
  - BGP next hop not belonging to the router advertising the prefix
The 4 filtering modes (egress)

• "Filtering based on both IRRdb and RPKI data" (default)
• “Filtering based on IRRdb data”
• “Filtering based on RPKI data”
• “Just tagging”
IRRdb Filtering 1/2

• RS config is generated automatically based on IRRdb parser scripts
  – Info gathered from all major IRR DBs
  – We detect policy changes every hour
• Import-via/export-via are supported
Outgoing filtering based on IRR policies
– You define your policy -> you instruct the RS
Keep IRR objects up-to-date
RKPİ Filtering

• BGP announcements are validated with RIPE’s RKPİ validator
• The prefixes that are being blocked are the ones with ROA status “INVALID”
Just tagging

• No filtering is applied to announced prefixes
  – But we still mark the received prefixes with the corresponding community tags:
    • ROA status: **VALID** (6777:65012)
    • ROA status: **INVALID** (6777:65022)
    • ROA status: **UNKNOWN** (6777:65023)

  o Present in AS’s announced AS/AS-SET (6777:65011)
  o Not present in AS’s announced AS/AS-SET (6777:65021)
BGP communities

• Manipulate prefix announcement via BGP community attributes:
  – Do not announce a prefix to a certain peer (0:peer-as)
  – Announce a prefix to a certain peer (6777:peer-as)
  – Do not announce a prefix to any peer (0:6777)
  – Announce a prefix to all peers (6777:6777)
Dynamic per-AS Prefix Limits

• Intended to prevent route leaks
• Dynamic limit is a necessity due to Tier 1 networks
  – Use IRRdb prefixes to calculate initial limit
  – For customers sending few prefixes limit = 100
  – Maximum = 20,000
Policy explorer

- Available at my.ams-ix.net (soon for users)

![Route Server filtering and policy explorer](image-url)

<table>
<thead>
<tr>
<th>Announced (outgoing) Prefix</th>
<th>ROA valid?</th>
<th>IRRdb object present?</th>
</tr>
</thead>
<tbody>
<tr>
<td>129.125.0.0/16</td>
<td>unknown</td>
<td>✓</td>
</tr>
<tr>
<td>130.112.0.0/16</td>
<td>unknown</td>
<td>✓</td>
</tr>
<tr>
<td>130.115.0.0/16</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>130.161.0.0/16</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>130.37.0.0/16</td>
<td>unknown</td>
<td>✓</td>
</tr>
<tr>
<td>130.80.0.0/16</td>
<td>unknown</td>
<td>✓</td>
</tr>
<tr>
<td>131.155.0.0/16</td>
<td>unknown</td>
<td>✓</td>
</tr>
<tr>
<td>131.174.0.0/16</td>
<td>unknown</td>
<td>✓</td>
</tr>
<tr>
<td>131.176.1.0/24</td>
<td>unknown</td>
<td>✓</td>
</tr>
<tr>
<td>131.176.103.0/24</td>
<td>unknown</td>
<td>✓</td>
</tr>
<tr>
<td>131.176.105.0/24</td>
<td>unknown</td>
<td>✓</td>
</tr>
<tr>
<td>131.176.106.0/23</td>
<td>unknown</td>
<td>✓</td>
</tr>
<tr>
<td>131.176.108.0/24</td>
<td>unknown</td>
<td>✓</td>
</tr>
<tr>
<td>131.176.123.0/24</td>
<td>unknown</td>
<td>✓</td>
</tr>
<tr>
<td>131.176.124.0/24</td>
<td>unknown</td>
<td>✓</td>
</tr>
<tr>
<td>131.176.126.0/24</td>
<td>unknown</td>
<td>✓</td>
</tr>
</tbody>
</table>
Other functionalities

• Traffic engineering
AS-Path prepending

• By tagging a specific prefix with one of the following communities:
  – 6777:65501 to prepend AS x1 towards all other peers
  – 6777:65502 to prepend AS x2 towards all other peers
  – 6777:65503 to prepend AS x3 towards all other peers
Real life example/problems

- Member A (old config)
- Member B (prefix hijack)
Member A example

• A big outage due to a BGP announcement to AMS-IX peering LAN (March 2011):
  – Containing the AMS-IX prefix (195.69.144.0/22)
  – ASN was not “6777”
  – The subnet mask was more specific
Member B example

- Classic prefix hijacking
  - Advertising 80.249.208.0/22 instead of /21
    - Announced by: ASXXXX
    - Upstream AS: ASYYYY
    - ASPATH: YYYY XXXX
  - RPKI detected it successfully
    - “RPKI Status: ROA validation failed: Invalid Origin ASN, expected 1200”
Questions?

stavros.konstantaras@ams-ix.net
aris.lambrianidis@ams-ix.net