Securing Internet Routing: RPSL

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
Target Audience

- Knowledge of Internet Routing (specially BGP)
- Fair idea on Routing Policy
- Familiar with any IRR Database
- No need to know Cryptography
- Basic knowledge of PKI (Public Key Infrastructure)
Agenda

• BGP 101
• Routing Policy
• RPSL
  – Configuration & Hands on Lab
• RPKI
  – Configuration & Hands on Lab
AS Path

Send a packet to 2001:DB8::1

I have 2001:DB8::/32

65553

65549
AS Path

2001:DB8::/32  65551  65550  65549  i

Send a packet to 2001:DB8::1

I have 2001:DB8::/32

65553  65551  65550  65549
AS Path

<table>
<thead>
<tr>
<th>AS Path</th>
<th>AS1</th>
<th>AS2</th>
<th>AS3</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001:DB8::/32</td>
<td>65551</td>
<td>65550</td>
<td>65549</td>
<td>i</td>
</tr>
<tr>
<td>2001:DB8::/32</td>
<td>65552</td>
<td>65536</td>
<td>i</td>
<td></td>
</tr>
</tbody>
</table>

Send a packet to 2001:DB8::1

I have 2001:DB8::/32
BGP Best Path Calculation

- Drop if own AS in AS-Path
- Prefer path with highest Weight
- Highest Local Preference
- Shortest AS-Path
- Lowest MED
- Path with shortest next hop metric (minimum IGP cost)
- Oldest received path
- Path from lowest neighbour address
Constructing the Forwarding Table

BGP in process

BGP table

Routing table

Forwarding table

Accepted

Discarded

In

BGP peer

Out

Everything

Best path
Control Plane and Forwarding Plane
Routing Incidents Types

- Incidents
  - Misconfiguration
  - Malicious
  - Targeted Traffic Misdirection

- For theory of positivity lets call all these as Mis-Origination

- Traffic Hijacking or Prefix Hijacking assumes Negative intent
Historical Incident

- April 1997: The "AS 7007 incident" UU/Sprint for 2 days
- February 24, 2008: Pakistan's attempt to block YouTube access within their country takes down YouTube entirely.[6]
- November 11, 2008: The Brazilian ISP CTBC - Companhia de Telecomunicações do Brasil Central leaked their internal table into the global BGP table.
- April 8, 2010: China Telecom originated 37,000 prefixes not belonging to them in 15 minutes, causing massive outage of services globally.

source: http://en.wikipedia.org/wiki/IP_hijacking
## Securing Internet Routing

To Secure Internet Routing; we need to check:

<table>
<thead>
<tr>
<th>A network should only originate his own prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. How do we verify?</td>
</tr>
<tr>
<td>2. How do we avoid false advertisement?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A transit network should filter customer prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Check customer prefix and ASN delegation</td>
</tr>
<tr>
<td>2. Transitive trust</td>
</tr>
</tbody>
</table>
Secure Internet Routing

Secure Inter-Domain Routing (SIDR) Working Group’s model

Routing Policy System (RPS) Working Group’s model
Routing Policy

• Public description of the relationship between external BGP peers
• Can also describe internal BGP peer relationship
• Usually registered at an IRR (Internet Routing Registry) such as RADB or APNIC
Routing Policy

• Who are my BGP peers

• What routes are
  – Originated by a peer
  – Imported from each peer
  – Exported to each peer
  – Preferred when multiple routes exist

• What to do if no route exists
Why Define a Routing Policy

- Documentation
- Provides routing security
  - Can peer originate the route?
  - Can peer act as transit for the route?
- Allows automatic generation of router configurations
- Provides a debugging aid
  - Compare policy versus reality
What is RPSL

• Routing Policy Specification Language
• RPSL is object oriented
  – These objects are registered in the Internet Routing Registry (IRR)
  – route, autonomous system, router, contact and set objects
• RIPE-81 was the first language deployed in the Internet for specifying routing policies
  – It was later replaced by RIPE-181
  – RPSL is a replacement for the RIPE-181 or RFC-1786
  – RPSL addresses RIPE-181's limitations
What is RPSL

• Describes things interesting to routing policy
  – Prefixes
  – AS Numbers
  – Relationships between BGP peers
  – Management responsibility

• For more about RPSL
  – RFC-1786: RIPE-181
  – RFC-2650: Using RPSL in Practice
  – RFC-2726: PGP Authentication for RIPE Database Updates
  – RFC-2725: Routing Policy System Security
  – RFC-2769: Routing Policy System Replication
  – RFC-4012: Routing Policy System Replication next generation
RPSL Objects

- RPSL objects are similar to RIPE-181 objects
- Objects
  - set of attributes
- Attributes
  - mandatory or optional
  - values: single, list, multiple
- Class “key”
  - set of attributes
  - usually one attribute has the same name as the object’s class
  - uniquely identify each object
- Class “key” = primary key
  - must be specified first
RPSL Attributes

• Case insensitive

• Value of an attribute has a type
  – <object-name>
  – <as-number>
  – <ipv4-address>
  – <ipv6-address>
  – <address-prefix>
  – etc

• Complete list of attributes and types in RFC 2622
  – https://www.rfc-editor.org/rfc/rfc2622.txt
RPSL Objects Example

attribute Name       Attribute Value
role:               APNIC Training
address:            6 Cordelia Street
address:            South Brisbane
address:            QLD 4101
country:            AU
phone:              +61 7 3858 3100
fax-no:             +61 7 3858 3199
e-mail:             training@apnic.net
admin-c:            NR97-AP
tech-c:             NR97-AP
nic-hdl:            AT480-AP
mnt-by:             MAINT-AU-APNICTRAINING
changed:            hm-changed@apnic.net 20080424
source:             APNIC
Integration of whois & IRR

• Integrated APNIC whois database & Internet Routing Registry

IP, ASNs, reverse domains, contacts, maintainers etc

APNIC whois

IRR

routes, routing policy, filters, peers etc

Internet Resources & Routing Information
# APNIC Database Objects and Routing Registry

<table>
<thead>
<tr>
<th>OBJECT</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>person</td>
<td>Technical or administrative contacts responsible for an object</td>
</tr>
<tr>
<td>role</td>
<td>Technical or administrative contacts represented by a role, performed by one or more people</td>
</tr>
<tr>
<td>inetnum</td>
<td>Allocation or assignment of IPv4 address space</td>
</tr>
<tr>
<td>inet6num</td>
<td>Allocation or assignment of IPv6 address space</td>
</tr>
<tr>
<td>aut-num</td>
<td>Registered holder of an AS number and corresponding routing policy</td>
</tr>
<tr>
<td>domain</td>
<td>in-addr.arpa (IPv4) or ip6.arpa (IPv6) reverse DNS delegations</td>
</tr>
<tr>
<td>route / route6</td>
<td>Single IPv4/IPv6 route injected into the Internet routing mesh</td>
</tr>
<tr>
<td>mntner</td>
<td>Authorized agent to make changes to an object</td>
</tr>
<tr>
<td>irt</td>
<td>Dedicated abuse handling team</td>
</tr>
</tbody>
</table>
### person / role Object

- The Person object register contact information

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>person:</td>
<td>mandatory</td>
<td>[single]</td>
</tr>
<tr>
<td>address:</td>
<td>mandatory</td>
<td>[multiple]</td>
</tr>
<tr>
<td>country:</td>
<td>mandatory</td>
<td>[single]</td>
</tr>
<tr>
<td>phone:</td>
<td>mandatory</td>
<td>[multiple]</td>
</tr>
<tr>
<td>fax-no:</td>
<td>optional</td>
<td>[multiple]</td>
</tr>
<tr>
<td>e-mail:</td>
<td>mandatory</td>
<td>[multiple]</td>
</tr>
<tr>
<td>nic-hdl:</td>
<td>mandatory</td>
<td>[single]</td>
</tr>
<tr>
<td>remarks:</td>
<td>optional</td>
<td>[multiple]</td>
</tr>
<tr>
<td>notify:</td>
<td>optional</td>
<td>[multiple]</td>
</tr>
<tr>
<td>abuse-mailbox:</td>
<td>optional</td>
<td>[multiple]</td>
</tr>
<tr>
<td>mnt-by:</td>
<td>mandatory</td>
<td>[multiple]</td>
</tr>
<tr>
<td>changed:</td>
<td>mandatory</td>
<td>[multiple]</td>
</tr>
<tr>
<td>source:</td>
<td>mandatory</td>
<td>[single]</td>
</tr>
</tbody>
</table>
person / role Object

person: Fakrul Alam
address: 6 Cordelia Street
address: South Brisbane
address: QLD 4101
country: AU
phone: +61738583100
e-mail: fakrul@apnic.net
nic-hdl: FA129-AP
mnt-by: MAINT-AU-APNICTRAINING
changed: fakrul@apnic.net 20151217
source: APNIC
**intenum / inetnum6 Object**

- Contains details of an allocation or assignment of IPv4/IPv6 address space

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Required</th>
<th>Key Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>inet6num</td>
<td>mandatory</td>
<td>single</td>
<td>primary/lookup key</td>
</tr>
<tr>
<td>netname</td>
<td>mandatory</td>
<td>single</td>
<td>lookup key</td>
</tr>
<tr>
<td>descr</td>
<td>mandatory</td>
<td>multiple</td>
<td></td>
</tr>
<tr>
<td>country</td>
<td>mandatory</td>
<td>multiple</td>
<td></td>
</tr>
<tr>
<td>geoloc</td>
<td>optional</td>
<td>single</td>
<td></td>
</tr>
<tr>
<td>language</td>
<td>optional</td>
<td>multiple</td>
<td></td>
</tr>
<tr>
<td>admin-c</td>
<td>mandatory</td>
<td>multiple</td>
<td>inverse key</td>
</tr>
<tr>
<td>tech-c</td>
<td>mandatory</td>
<td>multiple</td>
<td>inverse key</td>
</tr>
<tr>
<td>status</td>
<td>mandatory</td>
<td>single</td>
<td></td>
</tr>
<tr>
<td>remarks</td>
<td>optional</td>
<td>multiple</td>
<td></td>
</tr>
<tr>
<td>notify</td>
<td>optional</td>
<td>multiple</td>
<td>inverse key</td>
</tr>
<tr>
<td>mnt-by</td>
<td>mandatory</td>
<td>multiple</td>
<td>inverse key</td>
</tr>
<tr>
<td>mnt-lower</td>
<td>optional</td>
<td>multiple</td>
<td>inverse key</td>
</tr>
<tr>
<td>mnt-routes</td>
<td>optional</td>
<td>multiple</td>
<td>inverse key</td>
</tr>
<tr>
<td>mnt-irt</td>
<td>mandatory</td>
<td>single</td>
<td>inverse key</td>
</tr>
<tr>
<td>changed</td>
<td>mandatory</td>
<td>multiple</td>
<td></td>
</tr>
<tr>
<td>source</td>
<td>mandatory</td>
<td>single</td>
<td></td>
</tr>
</tbody>
</table>
intenum / inetnum6 Object

inet6num: 2406:6400::/32
netname: APNIC-TRAININGIPv6-Lab-AP
descr: APNIC TRAINING Lab
country: AU
admin-c: AT480-AP
tech-c: AT480-AP
mnt-by: APNIC-HM
mnt-lower: MAINT-AU-APNICTRAINING
mnt-routes: MAINT-AU-APNICTRAINING
status: ALLOCATED PORTABLE
remarks: To report network abuse, please contact the IRT
remarks: For troubleshooting, please contact tech-c and admin-c
remarks: For assistance, please contact the APNIC Helpdesk
source: APNIC
mnt-irt: IRT-APNICTRAINING-AU
changed: hm-changed@apnic.net 20100216
changed: hm-changed@apnic.net 20100818
mntner Object

- Maintainer objects used for authentication
  - Multiple auth / mnt-by / mntner-s are OR-ed

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mntner:</td>
<td>mandatory</td>
<td>single</td>
<td>primary/lookup key</td>
</tr>
<tr>
<td>descr:</td>
<td>mandatory</td>
<td>single</td>
<td></td>
</tr>
<tr>
<td>country:</td>
<td>optional</td>
<td>single</td>
<td></td>
</tr>
<tr>
<td>admin-c:</td>
<td>mandatory</td>
<td>multiple</td>
<td>inverse key</td>
</tr>
<tr>
<td>tech-c:</td>
<td>optional</td>
<td>multiple</td>
<td>inverse key</td>
</tr>
<tr>
<td>upd-to:</td>
<td>mandatory</td>
<td>multiple</td>
<td>inverse key</td>
</tr>
<tr>
<td>mnt-nfy:</td>
<td>optional</td>
<td>multiple</td>
<td>inverse key</td>
</tr>
<tr>
<td>auth:</td>
<td>mandatory</td>
<td>multiple</td>
<td>inverse key</td>
</tr>
<tr>
<td>remarks:</td>
<td>optional</td>
<td>multiple</td>
<td></td>
</tr>
<tr>
<td>notify:</td>
<td>optional</td>
<td>multiple</td>
<td>inverse key</td>
</tr>
<tr>
<td>abuse-mailbox:</td>
<td>optional</td>
<td>multiple</td>
<td>inverse key</td>
</tr>
<tr>
<td>mnt-by:</td>
<td>mandatory</td>
<td>single</td>
<td>inverse key</td>
</tr>
<tr>
<td>referral-by:</td>
<td>mandatory</td>
<td>single</td>
<td>inverse key</td>
</tr>
<tr>
<td>changed:</td>
<td>mandatory</td>
<td>multiple</td>
<td></td>
</tr>
<tr>
<td>source:</td>
<td>mandatory</td>
<td>single</td>
<td></td>
</tr>
</tbody>
</table>
mntner Object Example

mntner: MAINT-AU-APNICTRAINING
descr: APNIC Training
country: AU
admin-c: NR97-AP
technical-c: NR97-AP
auth: # Filtered
mnt-by: MAINT-AU-APNICTRAINING
upd-to: nurul@apnic.net
referral-by: APNIC-HM
changed: hm-changed@apnic.net 20131129
source: APNIC
Hierarchical Authorization

- `mnt-by` attribute
  - Refers to mntner object
  - Can be used to protect any object
  - Changes to protected object must satisfy authentication rules of `mntner` object

- `mnt-lower` attribute
  - Also refers to mntner object
  - Hierarchical authorization for inetnum, inetnum6 & domain objects
  - The creation of child objects must satisfy this mntner
  - Protects against unauthorized updates to an allocated range - highly recommended!

- `mnt-routers` attribute
  - Can be used to control the creation of 'route' objects associated with the address range specified by the inetnum and inet6num objects
Allocated to APNIC:
mnt-by can only be changed by IANA

Allocated to Member:
mnt-by can only be changed by APNIC

Sub-allocated to Customer:
mnt-by can only be changed by Member
Authorisation Mechanism

fakrul@www:~$ whois -h whois.apnic.net 2406:6400::/32

% Information related to '2406:6400::/32'

inet6num: 2406:6400::/32
netname: APNIC-TRAININGIPv6-Lab-AP
descr: APNIC TRAINING Lab
descr: LEVEL 1, 33 PARK RD
country: AU
admin-c: AT480-AP
tech-c: AT480-AP
mnt-by: APNIC-HM
mnt-lower: MAINT-AU-APNICTRAINING
mnt-routes: MAINT-AU-APNICTRAINING
status: ALLOCATED PORTABLE

1. This object can only be modified by APNIC-HM

2. Creation of more specific objects within this range has to pass the authentication of MAINT-AU-APNICTRAINING

3. Creation of route objects matching/within this range has to pass the authentication of MAINT-AU-APNICTRAINING
route/route6 Object

- Use CIDR length format
- Specifies origin AS for a route.
- Use both route and origin fields as the primary key

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>route</td>
<td>mandatory single</td>
</tr>
<tr>
<td>descr</td>
<td>mandatory multiple</td>
</tr>
<tr>
<td>country</td>
<td>optional single</td>
</tr>
<tr>
<td>origin</td>
<td>mandatory single</td>
</tr>
<tr>
<td>holes</td>
<td>optional multiple</td>
</tr>
<tr>
<td>member-of</td>
<td>optional multiple</td>
</tr>
<tr>
<td>inject</td>
<td>optional multiple</td>
</tr>
<tr>
<td>aggr-mtd</td>
<td>optional single</td>
</tr>
<tr>
<td>aggr-bndry</td>
<td>optional single</td>
</tr>
<tr>
<td>export-comps</td>
<td>optional single</td>
</tr>
<tr>
<td>components</td>
<td>optional single</td>
</tr>
<tr>
<td>remarks</td>
<td>optional multiple</td>
</tr>
<tr>
<td>notify</td>
<td>optional multiple</td>
</tr>
<tr>
<td>mnt-lower</td>
<td>optional multiple</td>
</tr>
<tr>
<td>mnt-routes</td>
<td>optional multiple</td>
</tr>
<tr>
<td>mnt-by</td>
<td>optional multiple</td>
</tr>
<tr>
<td>changed</td>
<td>mandatory multiple</td>
</tr>
<tr>
<td>source</td>
<td>mandatory single</td>
</tr>
</tbody>
</table>
route/route6 Example

```
route6: 2406:6400::/32
descr: APNIC Training Lab parent block
country: AU
origin: AS17821
notify: training@apnic.net
mnt-by: MAINT-AU-APNICTRAINING
changed: hm-changed@apnic.net 20100818
source: APNIC
```
aut-num Object

- Defines routing policy for an AS
- Uses import/mp-import: and export/mp-export: attributes to specify policy
- These define the incoming and outgoing routing announcement relationships
- Can reference other registry objects such as
  - as-sets / route-sets / filter-sets
aut-num Object

aut-num: [mandatory] [single] [primary/lookup key]
as-name: [mandatory] [single] [ ]
descr: [mandatory] [multiple] [ ]
country: [mandatory] [single] [ ]
member-of: [optional] [multiple] [inverse key]
import: [optional] [multiple] [ ]
export: [optional] [multiple] [ ]
default: [optional] [multiple] [ ]
remarks: [optional] [multiple] [ ]
admin-c: [mandatory] [multiple] [inverse key]
tech-c: [mandatory] [multiple] [inverse key]
notify: [optional] [multiple] [inverse key]
mnt-lower: [optional] [multiple] [inverse key]
mnt-routes: [optional] [multiple] [inverse key]
mnt-by: [mandatory] [multiple] [inverse key]
mnt-irt: [mandatory] [multiple] [inverse key]
changed: [mandatory] [multiple] [ ]
source: [mandatory] [single] [ ]
aut-num Object Example

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>aut-num:</td>
<td>AS17821</td>
</tr>
<tr>
<td>as-name:</td>
<td>APNIC-TRAINING-Lab-AS-AP</td>
</tr>
<tr>
<td>descr:</td>
<td>Two-byte AS number for APNIC Training</td>
</tr>
<tr>
<td>import:</td>
<td>from as4608 accept ANY</td>
</tr>
<tr>
<td>export:</td>
<td>to AS4608 announce AS17821</td>
</tr>
<tr>
<td>admin-c:</td>
<td>AT480-AP</td>
</tr>
<tr>
<td>tech-c:</td>
<td>AT480-AP</td>
</tr>
<tr>
<td>mnt-by:</td>
<td>MAINT-AU-APNICTRAINING</td>
</tr>
<tr>
<td>mnt-routes:</td>
<td>MAINT-AU-APNICTRAINING</td>
</tr>
<tr>
<td>mnt-irt:</td>
<td>IRT-APNICTRAINING-AU</td>
</tr>
<tr>
<td>changed:</td>
<td><a href="mailto:hm-changed@apnic.net">hm-changed@apnic.net</a> 20110701</td>
</tr>
<tr>
<td>source:</td>
<td>APNIC</td>
</tr>
</tbody>
</table>
as-set Object

• Collect together Autonomous Systems with shared properties
• Can be used in policy in place of AS
• RPSL has hierarchical names, can reference other as-set’s
  • Non-Hierarchical: AS-
  • Hierarchical: <origin-as-number>: AS-CUSTOMERS
  <origin-as-number>: AS-PEERS
## as-set Object

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>as-set:</td>
<td>mandatory</td>
<td>[primary/lookup key]</td>
</tr>
<tr>
<td>descr:</td>
<td>mandatory</td>
<td>[multiple] [ ]</td>
</tr>
<tr>
<td>country:</td>
<td>optional</td>
<td>[single] [ ]</td>
</tr>
<tr>
<td>members:</td>
<td>optional</td>
<td>[multiple] [ ]</td>
</tr>
<tr>
<td>mbrs-by-ref:</td>
<td>optional</td>
<td>[multiple] [inverse key]</td>
</tr>
<tr>
<td>remarks:</td>
<td>optional</td>
<td>[multiple] [ ]</td>
</tr>
<tr>
<td>tech-c:</td>
<td>mandatory</td>
<td>[multiple] [inverse key]</td>
</tr>
<tr>
<td>admin-c:</td>
<td>mandatory</td>
<td>[multiple] [inverse key]</td>
</tr>
<tr>
<td>notify:</td>
<td>optional</td>
<td>[multiple] [inverse key]</td>
</tr>
<tr>
<td>mnt-by:</td>
<td>mandatory</td>
<td>[multiple] [inverse key]</td>
</tr>
<tr>
<td>mnt-lower:</td>
<td>optional</td>
<td>[multiple] [inverse key]</td>
</tr>
<tr>
<td>changed:</td>
<td>mandatory</td>
<td>[multiple] [ ]</td>
</tr>
<tr>
<td>source:</td>
<td>mandatory</td>
<td>[single] [ ]</td>
</tr>
</tbody>
</table>
as-set Object Example

as-set: AS-APNICTRAINING
descr: AS-SET for APNIC Training
tech-c: AT480-AP
admin-c: AT480-AP
mnt-by: MAINT-AU-APNICTRAINING
changed: fakrul@apnic.net 20151215
members: AS17821
source: APNIC
route-set Object

• Defines a set of routes prefixes
• Name must begin with prefix “RS-” or in the format
  • ASNUM:RS-<ORGANIZATION>
• Can reference other route-sets, AS's or as-set's
  • In this case, the route-set will include all route object prefixes which have an origin which matches the AS numbers
## route-set Object

<table>
<thead>
<tr>
<th>Field</th>
<th>Mandatory</th>
<th>Single</th>
<th>Primary/Lookup Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>route-set:</td>
<td>[         ]</td>
<td>[single]</td>
<td>[primary/lookup key]</td>
</tr>
<tr>
<td>descr:</td>
<td>[         ]</td>
<td>[multiple]</td>
<td>[ ]</td>
</tr>
<tr>
<td>members:</td>
<td>[         ]</td>
<td>[multiple]</td>
<td>[ ]</td>
</tr>
<tr>
<td>mp-members:</td>
<td>[         ]</td>
<td>[multiple]</td>
<td>[ ]</td>
</tr>
<tr>
<td>mbrs-by-ref:</td>
<td>[         ]</td>
<td>[multiple]</td>
<td>[inverse key]</td>
</tr>
<tr>
<td>remarks:</td>
<td>[         ]</td>
<td>[multiple]</td>
<td>[ ]</td>
</tr>
<tr>
<td>tech-c:</td>
<td>[         ]</td>
<td>[multiple]</td>
<td>[inverse key]</td>
</tr>
<tr>
<td>admin-c:</td>
<td>[         ]</td>
<td>[multiple]</td>
<td>[inverse key]</td>
</tr>
<tr>
<td>notify:</td>
<td>[         ]</td>
<td>[multiple]</td>
<td>[inverse key]</td>
</tr>
<tr>
<td>mnt-by:</td>
<td>[         ]</td>
<td>[multiple]</td>
<td>[inverse key]</td>
</tr>
<tr>
<td>mnt-lower:</td>
<td>[         ]</td>
<td>[multiple]</td>
<td>[inverse key]</td>
</tr>
<tr>
<td>changed:</td>
<td>[         ]</td>
<td>[multiple]</td>
<td>[ ]</td>
</tr>
<tr>
<td>source:</td>
<td>[         ]</td>
<td>[single]</td>
<td>[ ]</td>
</tr>
</tbody>
</table>

Source: https://www.rfc-editor.org/rfc/rfc2622.txt
route-set Object Example

route-set: RS-APNICTRAINING
descr: Routes announced by APNIC Training
tech-c: AT480-AP
admin-c: AT480-AP
mnt-by: MAINT-AU-APNICTRAINING
changed: fakrul@apnic.net 20151215
mp-members: 2406:6400::/32, AS17821
source: APNIC
filter-set Object

- Defines a set of routes that are matched by a filter expression
- Similar in concept to route-set's
- Name must begin with prefix “fltr-”
filter-set Object Example

filter-set: fltr-martian-v6
descr: Current IPv6 MARTIANS
tech-c: FA129-AP
admin-c: FA129-AP
mnt-by: MAINT-AU-APNICTRAINING
changed: fakru1@apnic.net 20151221
mp-filter: {
0000::/8^+, # loopback, unspecified, v4-mapped
0064:ff9b::/96^+, # IPv4-IPv6 Translat. [RFC6052]
0100::/8^+, # reserved for Discard-Only Address Block [RFC6666]
0200::/7^+, # Reserved by IETF [RFC4048]
0400::/6^+, # Reserved by IETF [RFC4291]
0800::/5^+, # Reserved by IETF [RFC4291]
c000::/3^+, # Reserved by IETF [RFC4291]
e000::/4^+, # Reserved by IETF [RFC4291]
f000::/5^+, # Reserved by IETF [RFC4291]
f800::/6^+, # Reserved by IETF [RFC4291]
f000::/7^+, # Reserved by IETF [RFC4291]
f800::/10^+, # Link Local Unicast [RFC4291]
fe80::/10^+, # Reserved by IETF [RFC3879]
ff00::/8^+  # Multicast [RFC4291]
}
remarks: fltr-martian-v6 from RIPE-NCC
remarks: this object is manually maintained.
source: APNIC
Relation between objects

1. person:
   nic-hdl:
   KX17-AP
   Contact info

2. mntner:
   Data Protection

3. Allocation
   (Created by APNIC)

4. inetnum:
   KX17-AP
   mnt-by:
   ...

5. inetnum:
   KX17-AP
   mnt-by:
   ...

6. Customer Assignments
   (Created by Member)

inetnum:

KX17-AP
mnt-by:
...
Inter-related IRR Objects

- `aut-num: AS1`
  - ...
  - `tech-c: KX17-AP`
  - `mnt-by: MAINT-EX`
  - ...

- `route: 202.0.16/24`
  - `origin: AS1`
  - ...
  - `mnt-by: MAINT-EX`

- `inetnum: 202.0.16.0 - 202.0.16.255`
  - ...
  - `tech-c: KX17-AP`
  - `mnt-by: MAINT-EX`

- `person: ...
  - nic-hdl: KX17-AP
  - ...

- `mntner: MAINT-EX ...
  - ...

**Diagram:**
- Arrows indicate relationships between the IRR objects.
- Each box represents an IRR object with its associated attributes.
Inter-related IRR Objects

- **as-set:** AS1:AS-customers
  - **members:** AS10, AS11, AS2

- **route-set:** AS2:RS-routes
  - **members:** 218.2/20, 202.0.16/20

- **route:** 218.2/20
  - **origin:** AS2

- **inetnum:** 218.2.0.0 - 218.2.15.255
  - ...

- **route:** 202.0.16/20
  - **origin:** AS2

- **inetnum:** 202.0.16.0-202.0.31.255
  - ...

- **aut-num:** AS10
  - ...

- **aut-num:** AS11
  - ...

- **aut-num:** AS2
  - ...

- **aut-num:** AS2
  - ...
RPSL Objects & Routing Policy
The Internet Routing Registry (IRR)

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

- Stability and consistency of routing – network operators share information

- Both public and private databases

- These databases are independent – but some exchange data
  - only register your data in one database

- List of Routing Registry
  - http://www.irr.net/docs/list.html
The Internet Routing Registry (IRR)

- IRRs are used in at least three distinct ways
  - To publish your own routing intentions
  - To construct and maintain routing filters and router configurations
  - Diagnostic and information service for more general network management
IRR Objects Query

• whois query from cli

```bash
whois -h whois.apnic.net 2406:6400::/32
```

• You can search from APNIC website also
IRR Objects Query Flags

- IRR supports a number of flag option
  - ! RADB Query Flags
  - - RIPE/BIRD Query Flags
- `-i` flags for inverse query
  ```
  whois -h whois.apnic.net -i mnt-by MAINT-AU-APNICTRAINING
  
  [All the objects with a matching `mnt-by` attribute]
  whois -h whois.apnic.net -i origin as17821
  
  [`route` and `route6` objects with a matching `origin` attribute]
  
  - `q` flag for Informational queries
    ```
    whois -h whois.apnic.net -q sources
    
    [list of sources]
  ```
IRR Objects Query Flags

- `K` flags for primary keys of an object are returned
  
  ```
  whois -h whois.apnic.net -K 2406:6400::/32
  ```

- IRRd (IRR Daemon) supports service side set expansions (as-set and route-set)
  
  ```
  whois -h whois.radb.net '!iAS-APNICTRAINING'
  ```

  [returns members of AS-APNICTRAINING as-set object]

- For details please check
  - https://www.apnic.net/apnic-info/whois_search/using-whois/searching/query-options
  - http://www.radb.net/support/query2.php
RPSL Implementation: How to Begin

• Need to identify which IRR to use
  – May want to run your own for control

• Need to decide what degree of filtering is desired
  – Prefix filters
  – AS path filters
  – Both

• Register a maintainer object at chosen IRR
  – Usually a “manual” process and could be multi-stage if PGP key authentication required
RPSL Implementation : Checklist

1. Define your routing policy
2. Creating the objects in IRR
3. Use automated tools to generate the configuration
## Objects Involved

<table>
<thead>
<tr>
<th>Objects</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>route or route6 object</td>
<td>Connects a prefix to an origin AS</td>
</tr>
<tr>
<td>aut-num object</td>
<td>Registration record of an AS Number Contains the routing policy</td>
</tr>
<tr>
<td>sets</td>
<td>Objects can be grouped in sets, i.e. as-set, route-set</td>
</tr>
<tr>
<td>keywords</td>
<td>“ANY” matches every route</td>
</tr>
</tbody>
</table>
Import and Export Attributes

• You can document your routing policy in your aut-num object in the APNIC Database:
  – Import lines describe what routes you accept from a neighbor and what you do with them
  – Export lines describe which routes you announce to your neighbor

```
aut-num: AS17821
as-name: APNIC-TRAINING-Lab-AS-AP
descr: Two-byte AS number for APNIC Training Lab
country: AU
import: from AS45192 action pref=200; accept ANY
import: from AS4608 action pref=100; accept ANY
export: to AS45192 announce AS17821
export: to AS4608 announce AS17821
default: to AS45192 action pref=50; networks ANY
admin-c: AT480-AP
tech-c: AT480-AP
mnt-by: MAINT-AU-APNICTRAINING
mnt-routes: MAINT-AU-APNICTRAINING
changed: hm-changed@apnic.net 20080424
changed: hm-changed@apnic.net 20100818
changed: hm-changed@apnic.net 20100819
mnt-irt: IRT-APNICTRAINING-AU
changed: hm-changed@apnic.net 20110701
source: APNIC
```
Route Announcements vs Traffic Direction

AS17821 accepting all prefixes from AS4608 so that outbound traffic goes towards AS4608. It also makes localpref to 100.

AS17821 announcing prefixes (originating in AS17821) to AS4608, so that the incoming traffic for AS17821 can flow away from the AS4608.

aut-num: AS17821
import: from AS4608 action pref=100; accept ANY
export: to AS4608 announce AS17821
Routing Policy Scenarios

Internet

AS4608

Transit Provider

AS17821

You

AS131107

Downstream Customer

AS65543

Peer

aut-num: AS17821
import: from AS4608 accept ANY
export: to AS4608 announce AS17821 AS131107

import: from AS131107 accept AS131107
export: to AS131107 announce ANY

import: from AS65543 accept AS65543
export: to AS65543 announce AS17821 AS131107
Building an aut-num Object

- 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 AS131107 accept AS131107
mp-export: afi ipv6.unicast to AS131107 announce ANY
```

- More information in RFC 4012 RPSLng
## Filter List: Regular Expression

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS17821</td>
<td>AS 17821</td>
</tr>
<tr>
<td>AS17821*</td>
<td>0 or more occurrences of AS17821</td>
</tr>
<tr>
<td>AS17821+</td>
<td>1 or more occurrences of AS17821</td>
</tr>
<tr>
<td>AS17821?</td>
<td>0 or 1 occurrence of AS17821</td>
</tr>
<tr>
<td>&amp;</td>
<td>Beginning of Path</td>
</tr>
<tr>
<td>$</td>
<td>End of Path</td>
</tr>
<tr>
<td>\</td>
<td>Escape a regular expression character</td>
</tr>
<tr>
<td>_</td>
<td>Beginning, end, white-space, brace</td>
</tr>
<tr>
<td>AS17821</td>
<td>AS45192</td>
</tr>
<tr>
<td>AS17821AS45192</td>
<td>AS17821 followed by AS45192</td>
</tr>
<tr>
<td>()</td>
<td>Brackets to contain expression</td>
</tr>
<tr>
<td>[]</td>
<td>Brackets to contain numbers</td>
</tr>
</tbody>
</table>

Enclose the expression in “<“ and “>”
# Address Prefix Range Operator

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meanings</th>
</tr>
</thead>
<tbody>
<tr>
<td>^-</td>
<td>Exclusive more specifics of the address prefix: E.g. 128.9.0.0/16^- contains all more specifics of 128.9.0.0/16 excluding 128.9.0.0/16</td>
</tr>
<tr>
<td>^+</td>
<td>Inclusive more specific of the address prefix: E.g. 5.0.0.0/8^+ contains all more specifics of 5.0.0.0/8 including 5.0.0.0/8</td>
</tr>
<tr>
<td>^n</td>
<td>n = integer, stands for all the length “n” specifics of the address prefix: E.g. 30.0.0.0/8^16 contains all the more specifics of 30.0.0.0/8 which are length of 16 such as 30.9.0.0/16</td>
</tr>
<tr>
<td>^n-m</td>
<td>m = integer, stands for all the length “n” to length “m” specifics of the address prefix: E.g. 30.0.0.0/8^24-32 contains all the more specifics of 30.0.0.0/8 which are length of 24 to 32 such as 30.9.9.96/28</td>
</tr>
</tbody>
</table>
RPSL: localpref / prepend

• Controlling the traffic flow:
  – for outbound traffic set the value of local-pref
    • “action pref=NN” in the “import” lines of aut-num object
    • the lower the “pref”, the more preferred the route
  – for inbound traffic, modify as-path length
    • “action aspath.prepend(ASN)” in the “export” lines
    • Longer the as-path, less preferred the route

Note: the direction of traffic is reverse from accepting / announcing routes
RPSL: localpref/prepend Example

Local preference:

```
mp-import: afi ipv6.unicast from AS65001
2406:6400:10::2 at 2406:6400:10::1 action
community.append(17821:65001); pref=200; accept
<^AS65001+"> AND RS-APNICTRAINING:AS65001
```

Default value is 1000. Setting pref value to 200 mean downgrade the pref value by 200. Local pref will be 800.

Prepend:

```
mp-export: afi ipv6.unicast to AS65001 2406:6400:10::2
at 2406:6400:10::1 action aspath.prepend (AS17821,AS17821);
announce ANY AND NOT FLTR-MARTIAN-V6
```
RPSL: Multiple Links / MED

• By setting the value of MED on export lines, the preferred entry point into your AS can be controlled.

• The neighbour must agree to honour your MED values:
  – Instead of MED, it is possible to use as-path prepend on less preferred link.
RPSL: MED Example

export: to AS17821 10.0.0.4 at 10.0.0.1 action med=1000;
announce AS65001
export: to AS17821 10.0.0.5 at 10.0.0.2 action med=2000;
announce AS65001
RPSL: BGP Communities

• Elegant solution for implementing policies

• Optional tags
  – Can go through many peers

• Can be used for advanced filtering

• Enables customers to control their own routing policy
  – Publish your communities, and what you do with them
  – Filter incoming announcements accordingly
mp-import:       afi ipv6.unicast from AS65001
2406:6400:10::2 at 2406:6400:10::1 action
community.append(17821:65001); pref=200; accept
<^AS65001+'>$> AND RS-APNICTRAINING:AS65001
RPSL Tools

• IRRToolkit (written in C++)
  – https://github.com/irrtoolset/irrtoolset/

• Rpsltool (perl, using Template::Toolkit)
  – http://www.linux.it/~md/software

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

• BGPQ3 (C)

• Filtergen (Level 3)
  – Online tool using whois protocol
  – whois -h filtergen.level3.net RIPE::ASxxxx
# RPSL Tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| IRRToolSet    | • Full RPSL support  
• RPSLng support  
• 32-bit ASN support  
• Full BGP config generation | • No AS-Set query support  
• Manual peering configuration on the fly  
• Difficult to understand |
| IRR Power Tools | • Route aggregation  
• AS-SET queries | • No RPSLng support  
• No 32-bit ASN support |
| BGPq3         | • RPSL support  
• RPSLng support  
• 32-bit ASN  
• AS-SET queries  
• Easy to use | • Only partial BGP configuration. Can’t extract policy from IRR |
| RPSLtool      | • 32-bit ASN  
• AS-SET queries | • No RPSLng support |
| Net::IRR      | • RPSL and RPSLng support | • Outdated  
• Doesn’t support community attribute from RPSL data  
• No AS-SET queries |
| Netconfigs    | • Provides peering analysis  
• Can generate full configuration based on peering relationship | • Doesn’t support RPSLng  
• No command line query  
• Vendor dependent (CISCO) |

Source: Research project on “Automated configuration of BGP on edge routers” by University of Amsteradm; August 14, 2015
Use of RPSL

• Use RtConfig to generate filters based on information stored in our routing registry
  – Avoid filter errors (typos)
  – Filters consistent with documented policy (need to get policy correct though)
  – Engineers don’t need to understand filter rules (it just works :-)

• Some providers have own tools.
Using RPSL to Configure Routers

• Need to define “policy” for filtering
  – Inbound from customers & peers
  – Outbound to customers & peers

• Need to be aware of shortcomings in router configuration and/or configuration generator
  – Command line length (on cisco this is 512 bytes)
  – Complexity of rules
Filtering Philosophy

• Inbound
  – Filter customer by prefix and AS path
  – Filter peer by AS path only but don’t accept host routes
  – Filter providers for prefixes longer than a /24
  – Don’t accept martians from anyone

• Outbound
  – Filter by BGP community, which indicates the class of the prefix (customer, peer, etc)
Martians

• RtConfig has built in list of martians that can be added automatically to filters by use of command line option
• -supress_martian is Deprecated
• Properly maintained martian and bogon lists are visible in both the RIPE and Merit whois servers
• You can use following filter-set from APNIC whois
  – fltr-martian-v4 / fltr-martian-v6
IRRToolSet : Installation

• Dependency (Debian / Ubuntu)

```sh
# apt-get install build-essential libtool subversion bison flex libreadline-dev autoconf automake
```

• Installation

```sh
# wget ftp://ftp.isc.org/isc/IRRToolSet/IRRToolSet-5.0.1/irrtoolset-5.0.1.tar.gz
# tar -zxvf irrtoolset-5.0.1.tar.gz
# cd irrtoolset-5.0.1
# ./configure
# make
# make install
```

For details : https://github.com/irrtoolset/irrtoolset/blob/master/README.md
RtConfig Command Line Options

• Defaults to using RADB
  – -h whois.ra.net / whois.radb.net
  – -p 43
  – Default protocol irrd

• For other RIR use protocol bird
  – -protocol bird/ripe

• Defaults to “cisco” style output
  – -config cisco / -config junos

• -s <list of IRR sources>
  – -s APNIC,RADB,RIPE
RtConfig Syntax

• import / export pair for each link; syntax

```bash
@RtConfig [import/export] <yourASN> <yourRouterIP> <neighbourASN> <neighbourRouterIP>
```

• Takes other command also

```bash
@RtConfig configureRouter <inet-rtr-name>
@RtConfig static2bgp <ASN-1> <rtr-1>
@RtConfig access_list filter <filter>
```

• And many more. But best thing to look

```
man rtconfig
```
bash-3.2$ rtconfig -protocol bird -config cisco -h whois.radb.net

rtconfig> @RtConfig import AS17821 2406:6400:10::1 AS65001 2406:6400:10::2 !
no ipv6 access-list ipv6-500
ipv6 access-list ipv6-500 permit 2406:6400:8000::/48 any
ipv6 access-list ipv6-500 deny any any any!
no ip as-path access-list 500
ip as-path access-list 500 permit ^(_65001)+$

<output truncated>

router bgp 17821 !
  neighbor 2406:6400:10::2 remote-as 65001
  address-family ipv4
  no neighbor 2406:6400:10::2 activate
  address-family ipv6 unicast
  neighbor 2406:6400:10::2 activate
  neighbor 2406:6400:10::2 route-map AS65001-IN in
exit
IRRToolSet JunOS Example

bash-3.2$ rtconfig -protocol bird -config junos -h whois.radb.net

rtconfig> @RtConfig import AS17821 2406:6400:10::1 AS65001 2406:6400:10::2
policy-options {
    community community-1 members [17821:65001];
    as-path as-path-1 "( 65001)+";

<output truncated>

protocols {
    bgp {
        group peer-2406:6400:10::2 {
            type external;
            peer-as 65001;
            neighbor 2406:6400:10::2 {
                import policy_65001_1 ;
                family inet6 {
                    unicast;
                }
            }
        }
    }
}
RPSL in practice : LAB
RtConfig: The Big Picture

Step 1: Create objects and policies in IRR database

Step 2: Input request

Step 3: Connect IRR database and generate related configuration

Step 4: Push configuration to the router
Topology : Region 1

- RPSL Object
  - aut-num : AS17821
  - mnt-by: MAINT-AU-APNICTRAINING
  - route-set: RS-APNICTRAINING
  - fltr-set: FLTR-MARTIAN-V6
IRRToolSet : RPSL Object

# whois -h whois.apnic.net as17821

mp-import: afi ipv6.unicast from AS65001 2406:6400:10::2 at 2406:6400:10::1 action community.append(17821:65001); pref=200; accept <^AS65001+>$ AND RS-APNICTRAINING:AS65001

mp-export: afi ipv6.unicast to AS65001 2406:6400:10::2 at 2406:6400:10::1 announce ANY AND NOT FLTR-MARTIAN-V6
RtConfig Configuration Template (provision.cfg) – Provision Customer

@RtConfig set cisco_map_first_no = 10
@RtConfig set cisco_map_increment_by = 10
@RtConfig set cisco_prefix_acl_no = 100
@RtConfig set cisco_aspath_acl_no = 100
@RtConfig set cisco_pktfilter_acl_no = 100
@RtConfig set cisco_community_acl_no = 10
@RtConfig set cisco_max_preference = 500

! ip bgp-community new-format
ipv6 unicast-routing
!
! AS65001 CONFIGURATION
@RtConfig set cisco_access_list_no = 500
@RtConfig set cisco_map_name = "AS65001-IMPORT"
@RtConfig import AS17821 2406:6400:10::1 AS65001 2406:6400:10::2
@RtConfig set cisco_access_list_no = 501
@RtConfig set cisco_map_name = "AS65001-EXPORT"
@RtConfig export AS17821 2406:6400:10::1 AS65001 2406:6400:10::2
!
end
IRRToolSet : RtConfig Output File

• Now generate the router configuration file

```bash
rtconfig -protocol bird -cisco_use_prefix_lists -config cisco -h whois.radb.net < provision.cfg > /private/tftpboot/router_config.cfg
```

• You will get output of full configuration
• Configuration will be saved in /private/tftpboot
RtConfig Configuration Template (change.cfg) – Update Customer

• Filter customer based on
  – Prefix List
  – AS-PATH access list

• For that we use
  – AS-SET
Upload Configuration

• Various ways to upload configuration:
  – SNMP Write
  – NETCONF XML Based
  – Automated Script using expect
Upload Configuration : SNMP

• Enable SNMP:

  access-list 99 permit 10.10.0.0 0.0.255.255
  snmp-server community APNIC rw 99
  snmp-server ifindex persist

  – Recommended to use SNMPv3.

• Run TFTP server
Upload Configuration : SNMP

#Set copy method:
snmpset -v 2c -c {community-string} {device-ip-address}
1.3.6.1.4.1.9.9.96.1.1.1.2.116 i 1

#Set sourcefile to network file:
snmpset -v 2c -c {community-string} {device-ip-address}
1.3.6.1.4.1.9.9.96.1.1.1.3.116 i 1

#Set destination to running-config:
snmpset -v 2c -c {community-string} {device-ip-address}
1.3.6.1.4.1.9.9.96.1.1.1.4.116 i 4

#Set TFTP server ip:
snmpset -v 2c -c {community-string} {device-ip-address}
1.3.6.1.4.1.9.9.96.1.1.1.5.116 a {ip-address-tftp-server}

#Set destination filename:
snmpset -v 2c -c {community-string} {device-ip-address}
1.3.6.1.4.1.9.9.9a6.1.1.1.1.6.116 s router_config.cfg

#Start tftp upload via via OID ccCopyEntryRowStatus:
snmpset -v 2c -c {community-string} {device-ip-address}
1.3.6.1.4.1.9.9.96.1.1.1.14.116 i 1

Note: The integer highlighted in red is a random integer and you can choose any integer between 1 and 255. Keep in mind to use the same integer for the whole upload procedure! See the integer as a session.
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
RPSL in Summary

1. Define Routing Policy
2. Create IRR Object/Objects
3. Run RtConfig to generate config
4. Push config to router/routers
Challenges for the Routing Registries

• Lots of Routing Registries
• Accuracy and completeness
• Not every Routing Registry is linked directly to an Internet Registry
  – Offline verification of the resource holder is needed
• Different authorization methods
• Mirrors are not always up to date
Thanks