APNIC Training

Internet Routing Registry

23 February 2009
Manila, Philippines

In conjunction with

Introduction

- Presenter
  - Amante Alvaran, *Senior Network Engineer*
    amante@apnic.net
- Elearning Facilitator
  - John Tan, *Training Officer*
    jtan@apnic.net

Objectives

- To provide an introduction to the APNIC Routing Registry
  - Explain basic concepts of the global RR
  - Outline the benefits of the APNIC Routing Registry
- NOT to:
  - Teach basic routing
  - Explain Internet resource policy and procedures
  - Provide advise on network configuration

Assumptions

- The audience
  - Knowledgeable about BGP routing
  - Curious about Internet Routing Registry usage (IRR)
  - But not yet familiar with Routing Policy Specification Language (RPSL) and IRR

Internet Routing Registry

Overview

Overview

- APNIC database recap
- What is IRR?
- Why use an IRR?
- APNIC database and the IRR
- Using the Routing Registry
  - Overview of IRRToolSet
- Benefit of using IRR
- Using RPSL in practice

APNIC database recap

APNIC database

- Public network management database
  - APNIC whois database contains:
    - Internet resource information and contact details
  - APNIC Routing Registry (RR) contains:
    - routing information
- APNIC RR is part of IRR
  - Distributed databases that mirror each other

Database object

- An object is a set of attributes and values
- Each attribute of an object...
  - Has a value
  - Has a specific syntax
  - Is mandatory or optional
  - Is single- or multi-valued
- Some attributes ...
  - Are primary (unique) keys
  - Are lookup keys for queries
  - Are inverse keys for queries
  - Object “templates” illustrate this structure

Person object example

– Person objects contain contact information

```
person:     Ky Xander
address:    ExampleNet Service Provider
address:    2 Pandora St Boxville
address:    Wallis and Futuna Islands
country:    WF
phone:      +680-368-0844
fax-no:     +680-367-1797
e-mail:     kxander@example.com
nic-hdl:    KX17-AP
mnt-by:     MAINT-ENET-WF
changed:    kxander@example.com 20020731
source:     APNIC
```

Querying whois db

- Unix
  - Whois –h whois.apnic.net <lookup key>
    - E.g. whois –h whois.apnic.net whois AS2000
- Whois web interface
  - http://www.apnic.net/apnic-bin/whois.pl
- Keys for querying
  - Primary key, other lookup keys
    - E.g. whois EX91-AP
  - Inverse key "-i {attribute} {value}"
    - E.g. whois -i mnt-by MAINT-EXAMPLE-AP
- APNIC whois db query options:
  - http://www.apnic.net/db/search/all-options.html

Advanced database queries

– Flags used for inetnum queries

None	find exact match
- l	find one level less specific matches
- L	find all less specific matches
- m	find first level more specific matches
- M	find all More specific matches
- x	find exact match (if no match, nothing)
- d	enables use of flags for reverse domains
- r	turn off recursive lookups

Database update process

- Update transactions
  - Create a new object
  - Change an object
  - Delete an object

- Updates are submitted by email
  - E-mail to: <auto-dbm@apnic.net>
- Email message contains template representing new or updated object

Database protection

- Authorisation
  - "mnt-by" references a mntner object
    - Can be found in all database objects
    - "mnt-by" should be used with every object!
- Authentication
  - Updates to an object must pass authentication rule specified by its maintainer object

Authentication methods

- 'auth' attribute
  - Crypt-PW
    - Crypt (Unix) password encryption
    - Use web page to create your maintainer
  - PGP – GNUPG
    - Strong authentication
    - Requires PGP keys
  - MD5
    - Available

Hierarchical authorisation

- 'mnt-by' attribute
  - Can be used to protect any object
  - Changes to protected object must satisfy authentication rules of 'mntner' object
- 'mnt-lower' attribute
  - Also references mntner object
  - Hierarchical authorisation for inetnum & domain objects
  - Creation of child objects must satisfy this mntner
  - Protects against unauthorised updates to an allocated range - highly recommended!

Prerequisite for updating objects

- Create person objects for contacts
  - To provide contact info in other objects
- Create a mntner object
  - To provide protection of objects
- Protect your person object

What is an IRR?

What is a Routing Registry?

- A repository (database) of Internet routing policy information
  - Autonomous Systems exchanges routing information via BGP
  - Exterior routing decisions are based on policy based rules
  - However BGP does not provides a mechanism to publish/communicate the policies themselves
  - RR provides this functionality
- Routing policy information is expressed in a series of objects

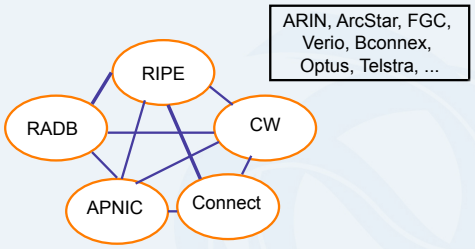
Routing registry objects

- Route, aut-num, inet-rtr, peering-set, AS-set, rtr-set, filter-set
  - Each object has its own purpose
  - Together express routing policies
- More details covered later

What is a Routing Registry?

- Global Internet Routing Registry database
  - http://www.irr.net/
    - Uses RPSL
- 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

What is a Routing Registry?



IRR = APNIC RR + RIPE DB + RADB + C&W + ARIN + …

Representation of routing policy

NET1 NET2

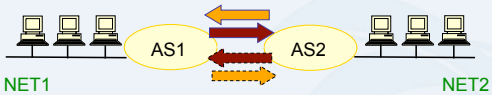
In order for traffic to flow from NET2 to NET1 between AS1 and AS2:

AS1 has to announce NET1 to AS2 via BGP

And AS2 has to accept this information and use it

Resulting in packet flow from NET2 to NET1

Representation of routing policy (cont.)



NET1 NET2

In order for traffic to flow towards from NET1 to NET2:

AS2 must announce NET2 to AS1

And AS1 has to accept this information and use it

Resulting in packet flow from NET 1 to NET2

What is routing policy?

- Description of the routing relationship between autonomous systems
  - Who are my BGP peers?
    - Customer, peers, upstream
  - What routes are:
    - Originated by each neighbour?
    - Imported from each neighbour?
    - Exported to each neighbour?
    - Preferred when multiple routes exist?
  - What to do if no route exists?
  - What routes to aggregate?

Why use an IRR?

Information to share

- Routes and AS objects give an abstract specification of the policy of an AS
  - Provides device independent view of routing policy
  - Neighbouring ASes can lookup, verify and understand the other party's policy
  - Provides a clear picture where this AS fits into the Internet

Information to share (cont.)

- Information – if every AS registers its policy and routes….
  - a global view of routing policy could be mapped
    - This global picture has the ability to improve the integrity of global Internet routing
  - Provides LIR/ISP with a mechanism to find all possible paths between any two points in the Internet
- Provides a high level of abstraction

Network planning

- Network planning
  - Simulation
    - Changes in polices can be simulated first by changing the registry but not the routers
      - To understand effects of policy changes to the existing networks
      - To make better network planning
      - To make it easier to adjust policies to maximise the performance of the network
  - Route filtering
    - Peering networks
    - A provider and its customer

Router configuration and network troubleshooting

- Router configuration
  - By using IRRToolSet
    - https://www.isc.org/software/irrtoolset-485
    - Extract information from IRR to create a router readable configuration file
    - Vendor independent
    - Protect against inaccurate routing info distribution
    - Verification of Internet routing
- Network troubleshooting
  - Easier to locate routing problems outside your network

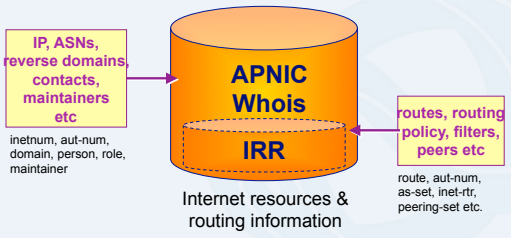
APNIC database and the IRR

APNIC Database & the IRR

- APNIC whois Database
  - Two databases in one
- Public Network Management Database
  - "whois" info about networks & contact persons
    - IP addresses, AS numbers etc
- Routing Registry
  - contains routing information
    - routing policy, routes, filters, peers etc.
  - APNIC RR is part of the global IRR

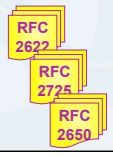
Integration of Whois and IRR

- Integrated APNIC Whois Database & Internet Routing Registry



RPSL

- Routing Policy Specification Language
  - Object oriented language
    - Based on RIPE-181
  - Structured whois objects
- Higher level of abstraction than access lists
- Describes things interesting to routing policy:
  - Routes, AS Numbers …
  - Relationships between BGP peers
  - Management responsibility
- Relevant RFCs
  - Routing Policy Specification Language
  - Routing Policy System Security
  - Using RPSL in Practice

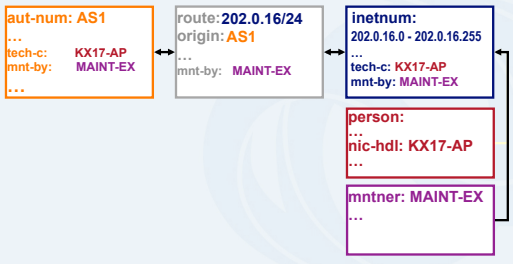


IRR objects

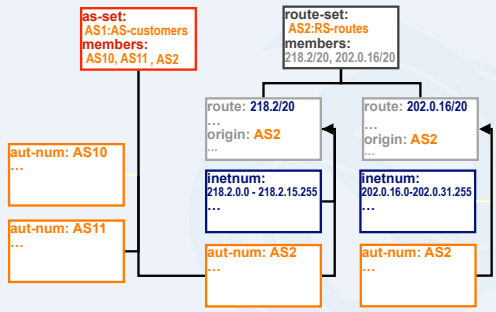
- route
  - Specifies interAS routes
- aut-num
  - Represents an AS. Used to describe external routing policy
- inet-rtr
  - Represents a router
- peering-set
  - Defines a set of peerings
- route-set
  - Defines a set of routes
- as-set
  - Defines a set of **aut-num** objects
- rtr-set
  - Defines a set of routers
- filter-set
  - Defines a set of routes that are matched by its filter

www.apnic.net/db/ref/db-objects.html

Inter-related IRR objects



Inter-related IRR objects



Hierarchical authorisation

- **mnt-routes**
  - authenticates ***creation*** of route objects
    - creation of route objects must pass authentication of mntner referenced in the mnt-routes attribute
  - Format:
    - `mnt-routes: <mntner>`

In:

`inetnum` `aut-num` **and** `route` **objects**

Authorisation mechanism

```
inetnum:     202.137.181.0 - 202.137.196.255
netname:     SPARKYNET-WF
descr:       SparkyNet Service Provider
…
mnt-by:      APNIC-HM
mnt-lower:   MAINT-SPARKYNET1-WF
mnt-routes:  MAINT-SPARKYNET2-WF
```

This object can only be modified by APNIC

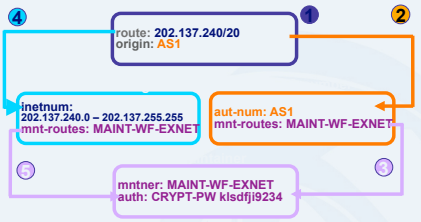
Creation of more specific objects (assignments) within this range has to pass the authentication of MAINT-SPARKYNET

Creation of route objects matching/within this range has to pass the authentication of MAINT-SPARKYNET-WF

Creating route objects

- Multiple authentication checks:
  - Originating ASN
    - mntner in the mnt-routes is checked
    - If no mnt-routes, mnt-lower is checked
    - If no mnt-lower, mnt-by is checked
  - AND the address space
    - Exact match & less specific route
      - mnt-routes etc
    - Exact match & less specific inetnum
      - mnt-routes etc
  - AND the route object mntner itself
    - The mntner in the mnt-by attribute

Creating route objects



1. Create route object and submit to APNIC RR database
2. DB checks aut-num obj corresponding to the ASN in route obj
3. Route obj creation must pass auth of mntner specified in aut-num *mnt-routes* attribute.
4. DB checks inetnum obj matching/encompassing IP range in route obj
5. Route obj creation must pass auth of mntner specified in inetnum *mnt-routes* attribute.

Using the Routing Registry

Overview of the IRRToolSet

IRRToolSet

- Set of tools developed for using the Internet Routing Registry (IRR)
- Work with Internet routing policies
  - These policies are stored in IRR in the Routing Policy Specification Language (RPSL)
- The goal of the IRRToolSet is to make routing information more convenient and useful for network engineers
  - Tools for automated router configuration,
  - Routing policy analysis
  - On-going maintenance etc.

IRRToolSet

- History
  - Originated at the USC Information Sciences Institute during 1997-2001 as the Routing Arbiter ToolSet (RAToolSet) project
  - Later migrated to RIPE NCC in order to continue its development and support (RAToolSet was later changed to IRRToolSet)
  - RIPE NCC later transferred maintenance of the tool set to ISC, who began accepting code from the community and providing code maintenance

IRRToolSet

- Now maintained by ISC:
  - http://irrtoolset.isc.org
  - Download: ftp://ftp.isc.org/isc/IRRToolSet/
    - Installation needs: lex, yacc and C++ compiler

Use of RPSL - RtConfig

- RtConfig v4
  - part of IRRToolSet
- Reads policy from IRR (aut-num, route & -set objects) and generates router configuration
  - vendor specific:
    - Cisco, Bay's BCC, Juniper's Junos and Gated/RSd
  - Creates route-map and AS path filters
  - Can also create ingress / egress filters
    - (documentation says Cisco only)

Why use IRR and RtConfig?

- Benefits of RtConfig
  - Avoid filter errors (typos)
  - Expertise encoded in the tools that generate the policy rather than engineer configuring peering session
  - Filters consistent with documented policy
    - (need to get policy correct though)

RtConfig commands

- @RtConfig import <ASN-1> <rtr-1> <ASN-2> <rtr-2>
  - Generate import filters where <rtr-1> in <ASN-1> is importing routes from <rtr-2> in <ASN-2>

i.g. @RtConfig import AS1 10.20.0.3 AS2 10.3.15.2

- @RtConfig export <ASN-1> <rtr-1> <ASN-2> <rtr-2>
  - Generate export filters where <rtr-1> in <ASN-1> is exporting routes to <rtr-2> in <ASN-2>

i.g. @RtConfig export AS1 10.20.0.3 AS2 10.3.15.2

RtConfig command (cont.)

- @RtConfig static2bgp <ASN-1> <rtr-1>
  - Generate the import policies of <ASN-1> where "protocol STATIC" or "protocol STATIC into BGP4" is used

  i.g. @RtConfig static2bgp AS1 10.3.0.1

RtConfig Cisco specific commands

- @RtConfig set cisco_map_name = <map-name>
  - Use <map-name> as the name for the route maps generated.
  - %d in <map-name> ⇨ replaced by the peer's ASN.
  - Second %d ⇨ replaced by an integer incremented No. of new map file creation.
  - The default cisco_map_name is "MyMap_%d_%d".

  i.g. @RtConfig set cisco_map_name = "AS%d-IMPORT"

- @RtConfig set cisco_max_preference = <no>
  - <no> is an integer defaulting to 1000.
  - Instruct RtConfig to start using preferences from <no> and counting down from there.

RtConfig Junos specific commands

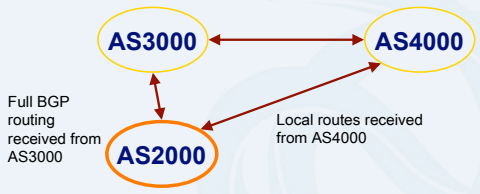
- @RtConfig set junos_policy_name = <policy-name>
  - Instruct RtConfig to use <policy-name> as the name for the policy statement generated.
  - %d in <policy-name> ⇨ replaced by the peer's ASN.
  - Second %d ⇨ replaced by an integer incremented No. of new policy file creation.
  - The default junos_policy_name is "policy_%d_%d".

RtConfig BCC specific commands

- @RtConfig set bcc_max_preference =<no>
  - <no> is an integer defaulting to 1000
  - Instruct RtConfig to start using preferences from <no> (most preferred) and counting down from there

Using RtConfig - Case scenario

Not fully multi-homing



Full BGP routing received from AS3000

Local routes received from AS4000

10.20.0.0/24 *(range received from upstream)*

10.187.65.0/24 *(portable address range)*

Using RtConfig – IRR objects

```
aut-num: AS2000
import:  from AS3000 accept ANY
export:  to AS3000 announce AS2000
import:  from AS4000 accept AS4000
export:  to AS4000 announce AS2000
[…]
```

full BPG routing
local routes

```
route: 10.20.0.0/24
origin: AS2000
[…]
```

```
route: 10.187.65.0/24
origin: AS2000
[…]
```

RtConfig commands

```
@RtConfig set cisco_map_name = "AS%d-IMPORT"
@RtConfig import AS2000 10.20.0.3 AS3000 10.3.15.2
!
@RtConfig set cisco_map_name = "AS%d-IMPORT"
@RtConfig import AS2000 10.20.0.4 AS4000 10.4.192.2
!
```

RtConfig output (import)

```
route-map AS3000-IMPORT permit 1
 match ip address prefix-list pl100
!
router bgp 2000
neighbor 10.3.15.2 route-map AS3000-IMPORT in
!
!
no ip prefix-list pl101
ip prefix-list pl101 permit 10.4.192.0/19
ip prefix-list pl101 deny 0.0.0.0/0 le 32
!
no route-map AS4000-IMPORT
!
route-map AS4000-IMPORT permit 1
 match ip address prefix-list pl101
!
router bgp 2000
neighbor 10.4.192.2 route-map AS4000-IMPORT in
```

The rest of the IRRToolSet

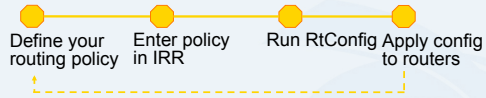
- peval
  - (Lightweight) policy evaluation tool
- prtraceroute
  - Prints the route packets take - including policy information (as registered in RR)
- aoe (aut-num object editor)
  - Displays the aut-num object for the specified AS
- roe
  - Creates the “route” object (based on BGP dump and routes in aut-num objects)

The rest of the IRRToolSet

- prpath
  - enumerates possible paths between two ASes
- CIDRAdvisor
  - suggests safe aggregates per AS
- *rpslcheck*
  - syntax checks objects for IRR

Benefit of using IRR

Using the Routing Registry



Costs

- Requires some initial planning
- Takes some time to define & register policy
- Need to maintain data in RR

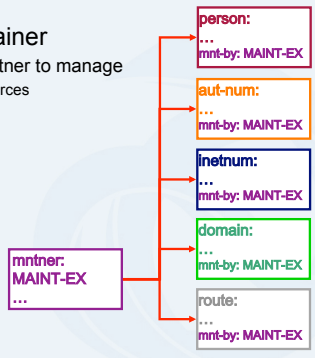
Benefits

- You have a clear idea of your routing policy
- Consistent config over the whole network
- Less manual maintenance in the long run

Benefits of APNIC RR

- Single maintainer
  - Use same mntner to manage
    - internet resources
    - reverse DNS
    - routing policy
    - contact info
    - etc

(Single person object can also be used)



Benefits of APNIC RR

– APNIC able to assert resources for a registered route within APNIC ranges.

```
inetnum:    221.0.0.0 - 221.3.127.255
netname:    CNCGROUP-SD
descr:      CNCGROUP Shandong province network
country:    CN
admin-c:    CH455-AP
tech-c:     XZ14-AP
mnt-by:     APNIC-HM
mnt-lower:  MAINT-CNCGROUP-SD
changed:    hm-chnaged@apnic.net 20021224
status:     ALLOCATED PORTABLE
source:     APNIC

mntner:     APNIC-HM
descr:      APNIC Hostmaster - Maintainer
...
```

Allocation objects maintained by APNIC

APNIC RR service scope

- Routing Queries
  - Regular whois clients
  - APNIC whois web interface
  - Special purpose programs such as IRRToolSet
- Routing Registration and Maintenance
  - Similar to registration of Internet resources

APNIC RR service scope

- Support
  - APNIC Helpdesk support

<helpdesk@apnic.net>

- Training
  - IRR Training
- Mirroring
  - APNIC mirrors IRRs within Asia Pacific and major IRRs outside of the region.

Summary

- APNIC RR integrated in APNIC Whois DB
  - whois.apnic.net
  - <auto-dbm@apnic.net>
- IRR benefits
  - Facilitates network troubleshooting
  - Generation of router configuration
  - Provides global view of routing
- APNIC RR benefits
  - Single maintainer (& person obj) for all objects
  - APNIC asserts resources for a registered route
  - Part of the APNIC member service!

Questions ?

RPSL

Objects, syntax and semantics

Overview

- Review of some of RR objects
- Useful queries
- Address prefix range operator
- AS-path regular expression
- Action specification
- Seven rp-attributes
- Syntax of policy actions and filters

RPSL

- Purpose of RPSL
  - Allows you to specify your routing configuration in the public IRR
    - Allows you to check “Consistency” of policies and announcements
  - Gives the opportunity to consider the policies and configuration of others
  - There are required syntax and semantics which need to be understood before using RPSL

RR objects review

- Aut-num object

Attribute	Value	Type
aut-num	<as-number>	mandatory, single-valued, class key
as-name	<object-name>	mandatory, single-valued
member-of	List of <as-set-name>	optional, multi-value
import	see next slide	optional, multi-value
export	see next slide	optional, multi-value

Aut-num object import attribute

- Each import policy expression is specified using an import attribute
- Syntax

```
import: from <peering-1> [action <action-1>]
        . . .
        from <peering-N> [action <action-N>]
        accept <filter>
```

The action specification is optional.

- Semantics
  - the set of routes that are matched by <filter> are imported from all the peers in <peerings>
  - importing routes at <peering-M>, <action-M> is executed

Aut-num object export attribute

- Each export policy expression is specified using an export attribute
- Syntax

```
export: to <peering-1> [action <action-1>]
        . . .
        to <peering-N> [action <action-N>]
        announce <filter>
```

The action specification is optional

- Semantics
  - the set of routes that are matched by <filter> are exported to all the peers specified in <peerings>
  - exporting routes at <peering-M>, <action-M> is executed

RR objects review

- route object

Attribute	Value	Type
route	Prefix of the InterAS route	mandatory, single-valued, class key
origin	<AS-number> originates the route	mandatory, single-valued
member-of	List of <route-set-name>	optional, multi-value
mnt-routes	see slide# 40	optional, multi-value

RR object review

- As-set object

Attribute	Value	Type
as-set	<object-name>	mandatory, single-valued, class key
members	List of <as-numbers> or <as-set-names>	optional, multi-value
Mbrs-by-ref	List of <mntner-names>	optional, multi-value

- As-set attribute starts with “as-”

RR object review

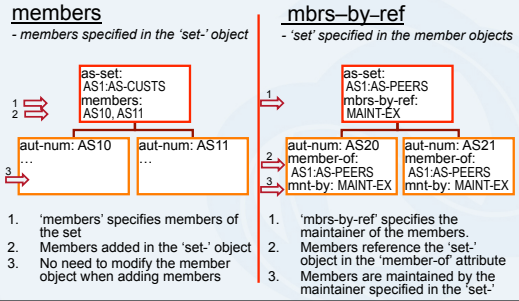
- Route-set object

Attribute	Value	Type
route-set	<object-name>	mandatory, single-valued, class key
members	List of <address-prefix-range> or <route-set-name><range-operator>	optional, multi-value
Mbrs-by-ref	List of <mntner-names>	optional, multi-value

- Route-set attribute starts with “rs-”

‘Set-’ objects and their members

- Two ways of referencing members



Useful IRR queries

- *What routes are originating from my AS?*
  - whois -i origin <ASN>
    - route objects with matching origin
- *What routers does my AS operate?*
  - whois -i local-as <ASN>
    - inet-rtr objects with a matching local-as
- *What objects are protecting "route space" with my maintainer?*
  - whois -i mnt-routes <mntner>
    - aut-num, inetnum & route objects with matching mnt-routes

(always specify host. e.g. 'whois –h whois.apnic.net')

Useful IRR queries (cont'd)

- *What '-set objects' are the objects protected by this maintainer a member of?*
  - whois –i mbrs-by-ref <mntner>
    - set objects (as-set, route-set and rtr-set) with matching mbrs-by-ref
- *What other objects are members of this '-set object'?*
  - whois -i member-of <set name>
    - Objects with a matching member-of
      - provided the membership claim is validated by the mbrs-by-ref of the set.

Address prefix range operator

Operator	Meanings
^-	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
^+	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

Address prefix operator (cont.)

Operator	Meanings
^n	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
^n-m	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

AS-path regular expressions

- Regular expressions
  - A context-independent syntax that can represent a wide variety of character sets and character set orderings
  - These character sets are interpreted according to the current The Open Group Base Specifications (IEEE)
- Can be used as a policy filter by enclosing the expression in “<“ and “>”.

AS-path regular expression

Operator	Meanings
<AS3>	Route whose AS-path contains AS3
<^AS1>	Routes whose AS-path starts with AS1
<AS2$>	Routes whose AS-path end with AS2
<^AS1 AS2 AS3$>	Routes whose AS-path is exactly “1 2 3”
<^AS1 . * AS2$>	AS-path starts with AS1 and ends in AS2 with any number ASN in between
<^AS3+$>	AS-path starts with AS3 and ends in AS3 and AS3 is the first member of the path and AS3 occurs one or more times in the path and no other AS can be present in the path after AS3

AS-path regular expression (cont.)

Operator	Meanings
<AS3|AS4>	Routes whose AS-path is with AS3 or AS4
<AS3 AS4>	Routes whose AS-path with AS3 followed by AS4

Action specification

- Policy action in RPSL
  - Set or modify route attributes
    - assigning a preference to a route
    - adding a BGP community to the BGP community path attribute
    - setting the Multi-Exist-Discriminator (MED) attribute
    - instructing routers to perform special operations
      - Route flap damping

Action specification (cont.)

- Routing policy attributes (rp-attributes)
  - Specified in the RPSL dictionary
  - Each action in RPSL is terminated by ";"
  - Possible to form composite policy actions
  - Actions are executed left to right

Sample:

```
aut-num:  AS1
import:   from AS2
          action pref = 10; med = 0;
          community.append (10250, 3561:10);
          accept { 128.9.0.0/16 }
```

Seven rp-attributes

pref	To assign local preference to the routes accepted
med	To assign a value to the Multi-Exit-Discriminator BGP attribute
dpa	To assign a value to the DPA BGP attribute
aspath	To prepend a value to the AS_PATH BGP attribute
community	To assign a value to or to check the value of the community BGP attribute
next-hop	To assign next hop routers to static routes
cost	To assign a cost to static routes

Questions ?

Using RPSL in practice

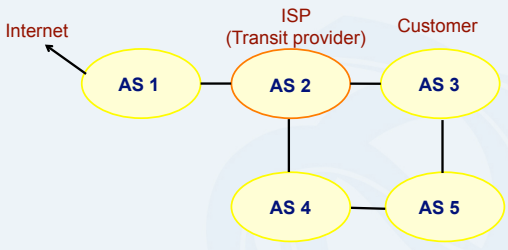
Overview

- Review examples of routing policies expression
  - Peering policies
  - Filtering policies
  - Backup connection
  - Multihoming policies

RPSL - review

- Purpose of RPSL
  - Allows specification of your routing configuration in the public IRR
    - Allows you to check “Consistency” of policies and announcements
  - Gives opportunities to consider the policies and configuration of others

Common peering policies



- Peering policies of an AS
  - Registered in an aut-num object

Common peering policies

- Policy for AS3 in the AS2 aut-num object

```
aut-num:    AS2
as-name:    SAMPLE-NET
dsescr:     Sample AS
import:     from AS1 accept ANY
import:     from AS3 accept <^AS3+$>
export:     to AS3 announce ANY
export:     to AS1 announce AS2 AS3
admin-c:    CW89-AP
tech-c:     CW89-AP
mtn-by:     MAINT-SAMPLE-AP
changed:    sample@sample.net
```

ISP customer – transit provider policies

- Policy for AS3 and AS4 in the AS2 aut-num object

```
aut-num:    AS2
import:     from AS1 accept ANY
import:     from AS3 accept <^AS3+$>
import:     from AS4 accept <^AS4+$>
export:     to AS3 announce ANY
export:     to AS4 announce ANY
export:     to AS1 announce AS2 AS3 AS4
```

AS-set object

- Describe the customers of AS2

```
as-set:     AS2:AS-CUSTOMERS
members:    AS3 AS4
changed:    sample@sample.net
source:     APNIC
```

Aut-num object referring as-set object

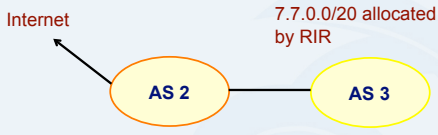
```
aut-num:    AS2
import:     from AS1 accept ANY
import:     from AS2:AS-CUSTOMERS accept
            <^AS2:AS-CUSTOMERS+$>
export:     to AS2:AS-CUSTOMERS announce ANY
export:     to AS1 announce AS2 AS2:AS-
            CUSTOMERS

aut-num:    AS1
import:     from AS2 accept <^AS2+AS2:AS-
            CUSTOMERS+$>
export:     ……
```

Express filtering policy

- To limit the routes one accepts from a peer
  - To prevent the improper use of unassigned address space
  - To prevent malicious use of another organisation's address space

Filtering policy



AS3 wants to announce part or all of 7.7.0.0/20 the global Internet.

AS2 wants to be certain that it only accepts announcements from AS3 for address space that has been properly allocated to AS3.

Aut-num object with filtering policy

```
aut-num:    AS2
import:     from AS3 accept { 7.7.0.0/20^20-24 }
…….
```

For an ISP with a growing or changing customer base, this mechanism will not scale well.

Route-set object can be used.

Route-set

```
route-set:  AS2:RS-ROUTES:AS3
members:    7.7.0.0/20^20-24
changed:    sample@sample.net
source:     APNIC
```

Specifies the set of routes that will be accepted from a given customer

Set names are constructed hierarchically:

AS2 : RS-ROUTES : AS3

AS2 — indicates whose sets these are

AS3 — indicates peer AS

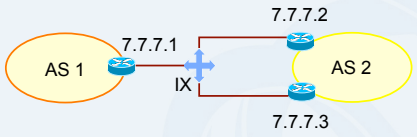
Filter configuration using route-set – AS2

```
import:     from AS1 accept ANY
import:     from AS3 accept AS2:RS-ROUTES:AS3
import:     from AS4 accept AS2:RS-ROUTES:AS4
export:     to AS2:AS-CUSTOMERS announce ANY
export:     to AS1 announce AS2 AS2:AS-CUSTOMERS
```

RPSL allows the peer's AS number to be replaced by the keyword PeerAS

```
import:  from AS2:AS-CUSTOMERS accept
         AS2:RS-ROUTES:PeerAS
```

Including interfaces in peering definitions: AS1



How to define AS1's routing policy by specifying its boundary router?

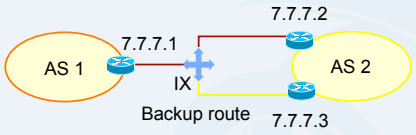
Including interfaces in peering definitions: AS1 (cont.)

```
aut-num:    AS1
import:     from AS2 at 7.7.7.1 accept <^AS2+$>
```

AS1 may want to choose to accept:
- only those announcements from router 7.7.7.2
- discard those announcements from router 7.7.7.3

```
aut-num: AS1
import:  from AS2 7.7.7.2 at 7.7.7.1 accept <^AS2+$>
```

Describing simple backup connections: AS1



How to define AS1's routing policy of its backup route?

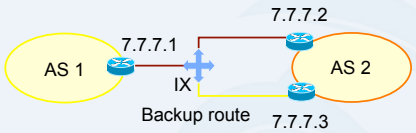
⇨ Use preference

Describing simple backup connections: AS1 (cont.)

```
aut-num:  AS1
import:   from AS2 7.7.7.2 at 7.7.7.1 action pref=10;
          from AS2 7.7.7.3 at 7.7.7.1 action pref=20;
          accept <^AS2+$>
```

Use of pref
- pref is opposite to local-pref
- Smaller values are preferred over larger values

Describing simple backup connections: AS2



How to define AS2's routing policy of AS1's backup route?

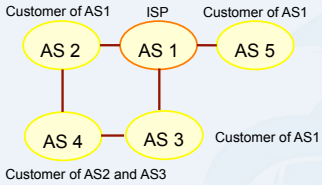
⇨ multi exit discriminator metric (med) can be used

Describing simple backup connections: AS2 (cont.)

```
aut-num:  AS2
export:   to AS1 7.7.7.1 at 7.7.7.2 action med=10;
          to AS1 7.7.7.1 at 7.7.7.3 action med=20;
          announce <^AS2+$>
```

Use of med
- Suitable for load balancing including backups

Multihome routing policy



AS1's base policy

- Only accepts routes from customers that are originated by the customer
- or by the customer's customers

Multihome routing policies (cont.)

```
aut-num:  AS1
import:    from AS2 accept (AS2 or AS4) AND
   <^AS2+AS4*$>
import:    from AS3 accept (AS3 or AS4) AND
   <^AS3+AS4*$>
import:    from AS5 accept AS5 AND <^AS5+$>
```

Review - BGP community attribute

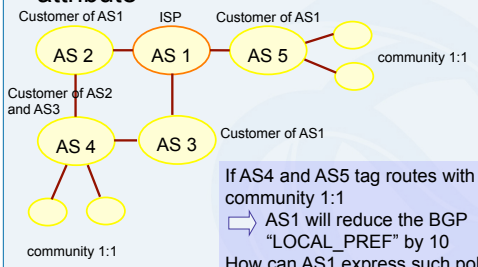
- Use of the BGP community attribute to provide support for:
  - Load balancing
  - Backup connections

- Basic premise of RFC1997:
  - To allow a customer to configure the BGP "LOCAL_PREF" on a provider's router

Review - BGP community attribute (cont.)

- Method to group destinations into communities and apply routing decisions
  - Community = a group of destinations (i.e. prefixes) that share some common attribute
  - Optional transitive attribute of variable length
- BGP community attribute
  - Format = aa:nn
    - aa = ASN
    - nn = 1 to 65,536

Multihome routing policies using the RPSL community attribute



If AS4 and AS5 tag routes with community 1:1
⇨ AS1 will reduce the BGP "LOCAL_PREF" by 10
How can AS1 express such policy in RPSL?

Multihome routing policies using the RPSL community attribute (cont.)

```
aut-num: AS1
import:   from AS2 action pref=10;
          accept (AS2 or AS4) AND <^AS2+AS4*$>  AND community(1:1)
import:   from AS2 action pref=0;
          accept (AS2 or AS4) AND <^AS2+AS4*$>

import:   from AS3 action pref=10;
          accept (AS3 or AS4) AND <^AS3+AS4*$> AND community(1:1)
import:   from AS3 action pref=0;
          accept (AS3 or AS4) AND <^AS3+AS4*$>

import:   from AS5 action pref=10;
          accept AS5 AND <^AS5+$> AND community(1:1)
import:   from AS5 accept pref=0;
          accept AS5 AND <^AS5+$>
```

Summary

What we discussed

- APNIC Whois database recap
- What is IRR and Why use it
- How to use the Routing Registry
- Benefit of using IRR
- Overview of IRRToolSet
- Using RPSL in practice

Usage: preliminary work for your AS

- Enter in the APNIC RR
  - Or in your own RR database
- Create person and mntner objects
- Describe policy in your aut-num object
- Identify IP prefixes associated with your AS
  - Create route objects in the database
  - Create route-set objects
- Crete various as-set objects, to group different categories of neighbours
- Create RtConfig template files
- Run RtConfig periodically to produce (parts of) router configuration file

Questions ?

Thank you!

References & Acknowledgements

- RPSL - RFC 2622
  - ftp.rfc-editor.org/in-notes/rfc2662.txt
- Using RPSL in Practice - RFC 2650
  - ftp.rfc-editor.org/in-notes/rfc2650.txt
- RIPE NCC IRR training material
  - http://www.ripe.net/training/rr/
- Internet Routing Registry Toolset Project
  - https://www.isc.org/software/IRRtoolset
- BGP community attribute
  - ftp://ftp.rfc-editor.org/in-notes/rfc1997.txt
- An Application of the BGP Community Attribute in Multi-home Routing
  - ftp://ftp.rfc-editor.org/in-notes/rfc1998.txt
- RADB
  - http://www.merit.edu/radb