Configuring routers with RPSL

APAN/TransPAC/NLANR/Internet2 Techs Workshop
Honolulu, January 2001
Mark Prior
Network Architect - Backbone Engineering
Who am I?

Network Architect for Tier 1 ISP in Australia
Use RPSL to manage our routing policy and configure routers
Member of the RPS working group at IETF
Agenda

Overview Routing Policy
Creating policy in RPSL
Using RtConfig to generate policy
Questions anytime!
What is 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 RIPE
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
What is RPSL?

• Object oriented language
• Structured whois objects
• Refinement of RIPE 181 (and it’s predecessors) based on operational experience
• Describes things interesting to routing policy
  – Prefixes
  – AS Numbers
  – Relationships between BGP peers
  – Management responsibility

RFC 2622 - “Routing Policy Specification Language (RPSL)”
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
Maintainer Objects

- Maintainer objects used for authentication
- Multiple authentication methods
  - NONE, MAIL-FROM, CRYPT-PW, PGPKEY

mntner: [mandatory] [single] [primary/look-up key]
descr: [mandatory] [multiple]
admin-c: [mandatory] [multiple] [inverse key]
tech-c: [optional] [multiple] [inverse key]
upd-to: [mandatory] [multiple] [inverse key]
mnt-nyf: [optional] [multiple] [inverse key]
auth: [mandatory] [multiple]
remarks: [optional] [multiple]
notify: [optional] [multiple] [inverse key]
mnt-by: [mandatory] [multiple] [inverse key]
changed: [mandatory] [multiple]
source: [mandatory] [single]
Maintainer Object Example

mntner: MAINT-AS2764
descr: Maintainer for AS 2764
admin-c: MP151
upd-to: routing@connect.com.au
mnt-nfy: routing@connect.com.au
auth: PGPKEY-81E92D91
auth: PGPKEY-562C2749
auth: PGPKEY-8C1EEB21
mnt-by: MAINT-AS2764
changed: mrp@connect.com.au 20000725
source: RADB
**key-cert Object Example**

```
key-cert:      PGPKEY-562C2749
method:        PGP
owner:         Connect Registry System <dbmon@connect.com.au>
owner:         Connect Registry System <routing@connect.com.au>
fingerpr:      A9 B7 B5 08 E5 37 07 B5 60 84 7B D3 E3 69 AA 2B
certif:

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.3ia

mQCNAzUDNN0AAAEALGWO23hXxzuvjrn1MvCHRWEtE51QeHxQ54EeqQYwQPEAMA
8kXyGe3Bz/2H71kgcrcBJByWhqrlpaxAJKzJyqPbrZDIXlyq63T35deCm2mSVN
G2hRe61j2cQSO4TN/3p5QujzXSBS6T8Rb66Yp/5amjEJvXNHcFFaxWLCdJAAUR
rDBDxb25uZWN0IFJ1Z21zdHJ5IFN5c3R1bSA8cm91dG1uZ0Bj25uZWN0ImNvbS5h
dT6jAUDPRA1AzZM2Tb8YHpxZEBAf9BADmIs6Nw+jnby4u+RVUurjQw9L615v
Ig9p6ohSikLMn7QffycjaclYzoN1uab0sc1ytd4vgzDHFRm30vt+3XPRRQPOqek
jv4C7iVJaQavoJxgXdoZBaCgUfTJLzBz3M20QlwAdx48oWOggmeoEMUTmfaw61
0Dj6b6k+i45hF2Yka1QMFEDUDNN4QhrWsvIwnSQEBRVID/jRQYNDD69Zj7ab3hR
R4ifpdRmja3j0mol9ho6JFrv/Um6f35Jjpu5LHW2LVP2ie1Cd91HVCopv/L0z344
PH8nT4jJdmcVcj4dHMIpqQDm3pt2t8h291Y27In1FmmHZvSolug6Qyw5b25mWDv
+cr5f0noJILxutua8ctrzPFc5Db25uZWN0IFJ1Z21zdHJ5IFN5c3R1bSA82GJt
b25AY9ubmVjdc5j2b0uXYU+iQCVAwUQNQMLBCCFxWLCdJAQF8KAP/XhrCbmMmX
4y21EK3r6k6kypa+j1F+NUEv7hdMmm609Su+yyvc3MwzqgwqopmLlz0U0huf71Ad
4NP4U5VifUz6C770v1S4NbnNqzUCSHTSmj0q31J2bQmLLUXRK0GoroRmjbbqKf
CMKvN1v1ac7vg6QNPvFqVR2OBMkbaei4=
=Keyg

-----END PGP PUBLIC KEY BLOCK-----

mnt-by:        MAINT-AS2764
changed:       mrp@connect.com.au 20000709
source:        RADB
```
Route Object

- Use CIDR length format
- Specifies origin AS for a route
- Can indicate membership of a route set

```
route: [mandatory] [single] [primary/look-up key]
descr: [mandatory] [multiple]
origin: [mandatory] [single] [primary/inverse key]
withdrawn: [optional] [single]
member-of: [optional] [single] [inverse key]
inject: [optional] [multiple]
components: [optional] [single]
aggr-bndry: [optional] [single] [inverse key]
aggr-mtd: [optional] [single]
export-comps: [optional] [single]
holes: [optional] [single]
remarks: [optional] [multiple]
cross-nfy: [optional] [multiple] [inverse key]
cross-mnt: [optional] [multiple] [inverse key]
notify: [optional] [multiple] [inverse key]
mnt-by: [mandatory] [multiple] [inverse key]
changed: [mandatory] [multiple]
source: [mandatory] [single]
```
Route Object Examples

route: 203.63.0.0/16
descr: connect.com.au pty ltd
origin: AS2764
notify: routing@connect.com.au
mnt-by: MAINT-AS2764
changed: mrp@connect.com.au 19971027
source: RADB

route: 203.102.39.0/24
descr: Web One (13480)
origin: AS2764
member-of: AS2764:RS-NEWSKIES
notify: routing@connect.com.au
mnt-by: CONNECT-AU
changed: mrp@connect.com.au 20001211
source: CCAIR
AS Set

- Collect together Autonomous Systems with shared properties
- Can be used in policy in place of AS
- RPSL has hierarchical names

\[
\begin{align*}
\text{as-set:} & \quad \text{[mandatory]} \quad \text{[single]} \quad \text{[primary/look-up key]} \\
\text{descr:} & \quad \text{[mandatory]} \quad \text{[multiple]} \\
\text{members:} & \quad \text{[optional]} \quad \text{[single]} \\
\text{mbrs-by-ref:} & \quad \text{[optional]} \quad \text{[single]} \\
\text{remarks:} & \quad \text{[optional]} \quad \text{[multiple]} \\
\text{tech-c:} & \quad \text{[mandatory]} \quad \text{[multiple]} \quad \text{[inverse key]} \\
\text{admin-c:} & \quad \text{[mandatory]} \quad \text{[multiple]} \quad \text{[inverse key]} \\
\text{notify:} & \quad \text{[optional]} \quad \text{[multiple]} \quad \text{[inverse key]} \\
\text{mnt-by:} & \quad \text{[mandatory]} \quad \text{[multiple]} \quad \text{[inverse key]} \\
\text{changed:} & \quad \text{[mandatory]} \quad \text{[multiple]} \\
\text{source:} & \quad \text{[mandatory]} \quad \text{[single]}
\end{align*}
\]
as-set:       AS2764:AS-CUSTOMERS:AS3409
descr:        connect.com.au AS set
members:      AS7632, AS9324
remarks:      Autonomous systems that transit through AS3409
admin-c:      CC89
tech-c:       MP151
mnt-by:       MAINT-AS2764
changed:      mrp@connect.com.au 20001214
source:       RADB
Route Set

- Collects routes together with similar properties

```
route-set:    [mandatory]  [single]    [primary/look-up key]
descr:        [mandatory]  [multiple]
members:      [optional]   [single]
mbrs-by-ref:  [optional]   [single]
remarks:      [optional]   [multiple]
technical-c:  [mandatory]  [multiple]  [inverse key]
administrative-c:  [mandatory]  [multiple]  [inverse key]
notify:       [optional]   [multiple]  [inverse key]
monitor-by:   [mandatory]  [multiple]  [inverse key]
changed:      [mandatory]  [multiple]
source:       [mandatory]  [single]
```
route-set: AS2764:RS-PROVIDER
descr: Connect's provider blocks
member 202.21.8.0/21, 203.8.176.0/21, 203.63.0.0/16, 210.8.0.0/14
admin-c CC89
technical-contact MP151
notify: routing@connect.com.au
mnt-by: MAINT-AS2764
changed: mrp@connect.com.au 20010118
source: RADB

route-set: AS2764:RS-NEWSKIES
descr: Routes announced across NewSkies satellite link
members-by-ref: CONNECT-AU
technical-contact MP151
admin-contact CC89
notify: routing@connect.com.au
mnt-by: CONNECT-AU
changed: mrp@connect.com.au 20010112
source: CCAIR
Autonomous System Object

- Routing Policy Description object
- Most important components are
  - import
  - export
- These define the incoming and outgoing routing announcement relationships
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aut-num</td>
<td>mandatory</td>
<td>single</td>
<td>primary/look-up key</td>
</tr>
<tr>
<td>as-name</td>
<td>mandatory</td>
<td>single</td>
<td></td>
</tr>
<tr>
<td>descr</td>
<td>mandatory</td>
<td>multiple</td>
<td></td>
</tr>
<tr>
<td>member-of</td>
<td>optional</td>
<td>single</td>
<td>inverse key</td>
</tr>
<tr>
<td>import</td>
<td>optional</td>
<td>multiple</td>
<td>inverse key</td>
</tr>
<tr>
<td>export</td>
<td>optional</td>
<td>multiple</td>
<td>inverse key</td>
</tr>
<tr>
<td>default</td>
<td>optional</td>
<td>multiple</td>
<td>inverse key</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>remarks</td>
<td>optional</td>
<td>multiple</td>
<td></td>
</tr>
<tr>
<td>cross-nfy</td>
<td>optional</td>
<td>multiple</td>
<td>inverse key</td>
</tr>
<tr>
<td>cross-mnt</td>
<td>optional</td>
<td>multiple</td>
<td>inverse key</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>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>
Simple “Documentation” Policy

• The simplest policy is strict customer/provider relationship
  – Customer accepts everything the provider sends
  – Customer sends its routes to provider

aut-num: AS2
as-name: EXAMPLE-NET
descr: RPSL Example
import: from AS1 accept ANY
export: to AS1 announce AS2
admin-c: ADMINISTRATION
tech-c: OPERATIONS
mnt-by: MAINT-AS2
changed: noc@example.net 20010101
source: TEST
Use of RPSL

• Use RtConfig v4 (part of RAToolSet from ISI) 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 but Connect finds that RtConfig does about 90% of BGP peering configuration
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
Connect’s 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)
RtConfig

- Version 4.0 supports RPSL
- Generates cisco configurations
- Contributed support for Bay’s BCC, Juniper’s Junos and Gated/RSd
- Creates route and AS path filters.
- Can also create ingress/egress filters (cisco only)
Martians

• RtConfig has built in list of martians that can be added automatically to filters by use of command line option
  – Based on Bill Manning’s Internet Draft
    • draft-manning-dsua-03.txt (now expired)

• Some people on RIPE WG mailing list were suggesting building a martian route set and using it explicitly in policy
“import” statements

• Use ASx to create prefix list
  – Include “route-set” and/or “as-set”
  – List of specific prefixes
• Use <ASx> to create AS path list
• Can combine these components in “interesting” ways
Simple “import” Policy

import: from ASx accept ASx and <^ASx+$>

- From ASx accept a prefix iff there exists a route object that exactly matches the prefix and is originated by ASx and the AS path is solely composed of ASx
Combining rules using sets

import: from ASx accept ASx and <^ASx+>$
import: from ASy accept ASy and <^ASy+>$
import: from ASz accept ASz and <^ASz+>$

import: from AS-SET
    accept PeerAS and <^PeerAS+>$

as-set:          AS-SET
descr:            Example Set
members:          ASx, ASy, ASz
technical-c:      MP151
administrative-c: MP151
mnt-by:           MAINT-AS2764
changed:          mqp@connect.com.au 20010101
source:           TEST
RFC 1998 - Use of BGP communities

import: from AS-SET
    action pref=30;
    accept community.contains(3561:70)
        and PeerAS and <^PeerAS+$$>
import: from AS-SET
    action pref=20;
    accept community.contains(3561:80)
        and PeerAS and <^PeerAS+$$>
import: from AS-SET
    action pref=10;
    accept community.contains(3561:90)
        and PeerAS and <^PeerAS+$$>
import: from AS-SET
    action pref=0;
    accept PeerAS and <^PeerAS+$$>
RtConfig command line options

- Defaults to using RADB
  - -h whois.ra.net
  - -p 43
  - -protocol irrd
- Defaults to “cisco” style output
  - -config cisco
- -suppress_martian
- -s <list of IRR sources>
  - -s CCAIR,RADB,CW
Simple example policy

aut-num: AS2170
as-name: ASN-EXAMPLE
descr: RPSL example policy
import: from AS2823
   action pref=0;
   accept AS2823
import: from AS2764
   action pref=5;
   accept ANY
import: protocol STATIC into BGP4
   from AS2170
   action community.append(2170:1);
   accept AS2170
export: to AS2823
   announce community.contains(2170:1)
export: to AS2764
   announce community.contains(2170:1)
default: to AS2764
admin-c: NOC
tech-c: NOC
remarks: simple policy with two "peers"
remarks: prefer AS2823 for it's own traffic
remarks: default to AS2764
notify: noc@inter.net
mnt-by: MAINT-AS2170
changed: noc@inter.net
source: TEST
Injecting static routes into BGP

• We use policy to filter static routes into BGP
  – Allows for martian filtering
  – Tagging routes with special communities
  – Filter host routes or other prefixes
RtConfig commands for static import

import: protocol STATIC into BGP4
from AS2170
action community.append(2170:1);
accept AS2170

@RtConfig set cisco_map_name = "STATIC-EXPORT"
@RtConfig static2bgp AS2170 0.0.0.0

• User defines name of route map
• RtConfig will create the required filters, etc
RtConfig commands for static import

```
RtConfig> @RtConfig set cisco_map_name = "STATIC-EXPORT"
RtConfig> @RtConfig static2bgp AS2170 0.0.0.0
    !
    no access-list 100
    access-list 100 permit ip 203.17.185.0 0.0.0.0 255.255.255.0 0.0.0.0
    access-list 100 permit ip 205.191.168.0 0.0.0.0 255.255.255.0 0.0.0.0
    access-list 100 permit ip 210.8.207.176 0.0.0.0 255.255.255.240 0.0.0.0
    access-list 100 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
    !
    no route-map STATIC-EXPORT
    !
    route-map STATIC-EXPORT permit 1
        match ip address 100
        set community 2170:1 additive
    !
    router bgp 2170
    redistribute static route-map STATIC-EXPORT
```
Using RtConfig on Simple Policy

RtConfig> @RtConfig import AS2170 0.0.0.0 AS2823 0.0.0.0
! no access-list 100
access-list 100 permit ip 203.10.111.0 0.0.0.0 255.255.255.0 0.0.0.0
access-list 100 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
! no route-map MyMap_2170_1
!
route-map MyMap_2170_1 permit 1
  match ip address 100
  set local-preference 1000
!
router bgp 2170
neighbor 0.0.0.0 route-map MyMap_2170_1 in
RtConfig> @RtConfig export AS2170 0.0.0.0 AS2823 0.0.0.0
! ip bgp-community new-format
! no ip community-list 1
ip community-list 1 permit 2170:1
!
no route-map MyMap_2170_2
!
route-map MyMap_2170_2 permit 1
  match community 1
!
router bgp 2170
neighbor 0.0.0.0 route-map MyMap_2170_2 out
Advanced static route importation

import:    protocol STATIC into BGP4 {
    from AS2764
    action community.append(2764:65408, 2764:65472);
    accept AS2764 OR ( AS2764:AS-CUSTOMERS AND NOT AS2764:RS-PROVIDER^0-30 );
} refine {
    from AS-ANY
    action community.append(2764:1);
    accept AS2764:RS-DOMESTIC;
    from AS-ANY
    action community.append(2764:8);
    accept AS2764:RS-SATELLITE;
    from AS-ANY
    action community.append(2764:10);
    accept AS2764:RS-HYBRID;
    from AS-ANY
    action community.append(2764:11);
    accept AS2764:RS-NEWSKIES;
    from AS-ANY
    action community.append(2764:13);
    accept AS2764:RS-TELSTRA;
    from AS-ANY
    accept ANY;
}

import:    protocol STATIC into BGP4
    from AS2764
    action community.append(2764:65472);
    accept AS2764:RS-PROVIDER^0-30 AND NOT AS2764:RS-SATELLITE^--

import:    protocol STATIC into BGP4
    from AS2764
    action community.append(2764:65472, no_export);
    accept AS2764:RS-PROVIDER^--
BGP Customer Import Policy

import: {
  from AS-ANY
  accept ANY AND NOT { 0.0.0.0/0 };
} refine {
  from AS-ANY
  action community.append(2764:65408); pref=25;
  accept community.contains(2764:3) AND NOT AS2764:RS-PROVIDER^--;
  from AS-ANY
  action community.append(2764:65408); pref=15;
  accept community.contains(2764:4) AND NOT AS2764:RS-PROVIDER^--;
  from AS-ANY
  action community.append(2764:65408); pref=5;
  accept community.contains(2764:5);
  from AS-ANY
  action community.append(2764:65408); pref=0;
  accept ANY;
} refine {
  from AS2764:AS-CUSTOMERS
  accept PeerAS AND <^PeerAS+$>;
  from AS2764:AS-TRANSIT
}
RtConfig Configuration Template

@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 = 100
!
router bgp 1
neighbor 10.0.0.1 remote-as 2
neighbor 10.0.0.1 description Internet2
@RtConfig set cisco_map_name = "AS2-EXPORT"
@RtConfig export AS1 0.0.0.0 AS2 0.0.0.0
@RtConfig set cisco_map_name = "AS2-IMPORT"
@RtConfig import AS1 0.0.0.0 AS2 0.0.0.0
neighbor 10.1.0.1 remote-as 3
neighbor 10.1.0.1 description Internet
@RtConfig set cisco_map_name = "AS3-EXPORT"
@RtConfig export AS1 0.0.0.0 AS2 0.0.0.0
@RtConfig set cisco_map_name = "AS3-IMPORT"
@RtConfig import AS1 0.0.0.0 AS2 0.0.0.0
!
end
cisco Configuration

! access-list 135 – customer routes
! no ip as-path access-list 130
ip as-path access-list 130 permit ^(_9313)+$!
! no route-map AS9313-IMPORT
! no ip community-list 32
ip community-list 32 permit 2764:3!
route-map AS9313-IMPORT permit 20
  match as-path 130
  match community 32
  match ip address 135
  set local-preference 75!
no ip community-list 33
ip community-list 33 permit 2764:4!
route-map AS9313-IMPORT permit 30
  match as-path 130
  match community 33
  match ip address 135
  set local-preference 85

no ip community-list 34
ip community-list 34 permit 2764:5!
route-map AS9313-IMPORT permit 40
  match as-path 130
  match community 34
  match ip address 135
  set local-preference 95!
route-map AS9313-IMPORT permit 50
  match as-path 130
  match ip address 135
  set local-preference 100!
routerr bgp 2764
neighbor 203.63.122.193 route-map AS9313-IMPORT in!
end
Problems?

• Policy can easily get very complex and result in even more complex router configuration
• Line limit on cisco AS path filters (need to be careful when using as-sets)
• ISI/Qwest whois server doesn’t cope with the RPSL v2 community format
References

• RPSL - RFC 2622

• Using RPSL in Practice - RFC 2650

• RAToolSet

• RPSL Training Page
  – http://www.isi.edu/ra/rps/training

• RADB
  – http://www.merit.edu/radb
Contact Details

person: Mark R. Prior
address: connect.com.au pty ltd
        C/- AAPT
        Level 1, 45 Pirie Street
        Adelaide 5000
        South Australia
phone: +61 8 8203 2088
fax-no: +61 8 8203 2087
e-mail: mrp@connect.com.au
nic-hdl: MP151
changed: mrp@connect.com.au 19980316
source: RADB