RPSL - Practical Tool for ISPs?

14th APNIC Open Policy Meeting
Kitakyushu, Japan

Andy Linton <asjl@lionra.net.nz>

Agenda

- Routing Policy
  - What is Routing Policy?
  - Why define one?
- RPSL
  - What is RPSL?
  - Benefits of using RPSL
  - How to use RPSL
- Questions anytime!

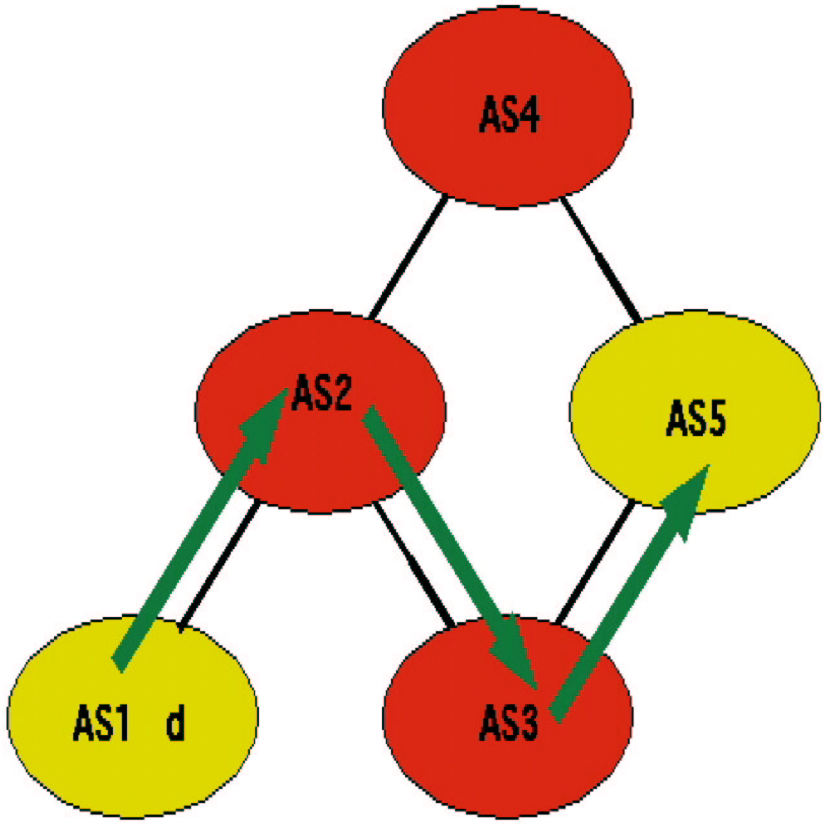
What is Routing Policy

- Public description of the relationship between external BGP peers
- Can also describe internal BGP peer relationship

Routing Policy

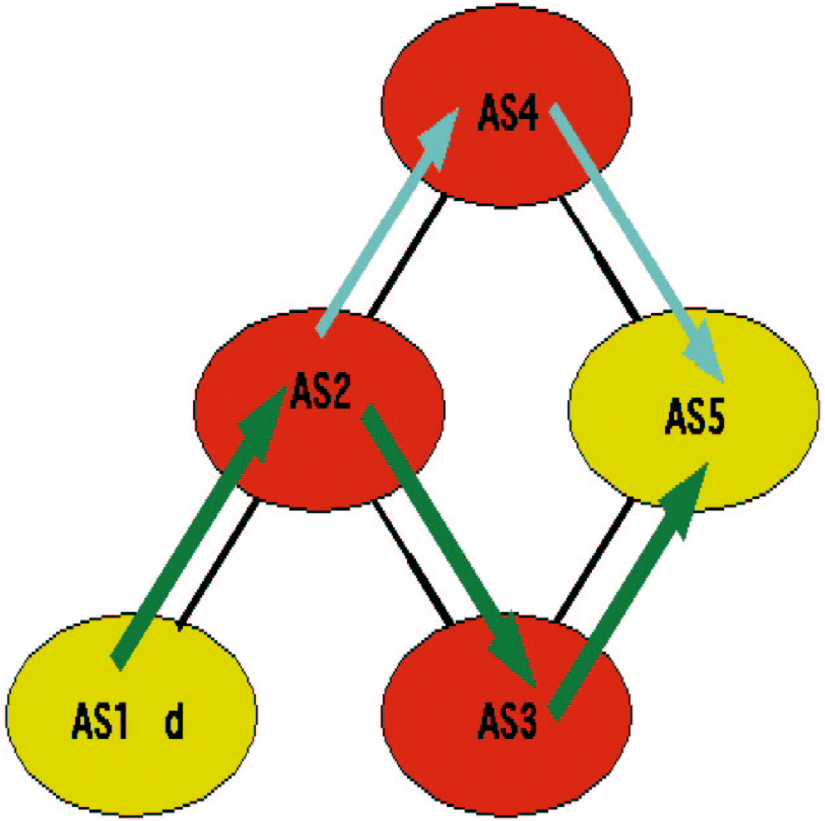
- Who are my BGP peers
- What routes are
  - Originated by a peer
  - Imported from each peer
  - Exported to each peer
  - Preferred when multiple routes exist
- What to do if no route exists

Routing Policy Example



- AS1 originates prefix "d"
- AS1 exports "d" to AS2, AS2 imports
- AS2 exports "d" to AS3, AS3 imports
- AS3 exports "d" to AS5, AS5 imports

Routing Policy Example (cont)



- AS5 also imports "d" from AS4
- Which route does it prefer?
  - Does it matter?
  - Consider case where
    - AS3 = Commercial Internet
    - AS4 = Internet2

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?

- Object oriented language
- Development of RIPE 181
- Structured whois objects
- Describes things interesting to routing policy:
  - Routes
  - AS Numbers
  - Relationships between BGP peers
  - Management responsibility

Maintainer Objects

- Maintainer objects used for authentication
- Authentication methods include:
  - NONE (not secure)
  - MAIL-FROM (not secure) (phased out by RIPE)
  - CRYPT-PW (better)
  - MD5-PW (better again)
  - PGPKEY (best)
- For more information see:

  http://www.ripe.net/ripencc/faq/database/qa5.html

Maintainer Object Example

mntner:	MAINT-AL325-AP
descr:	Lionra
descr:	NZ
admin-c:	AL325-WIX
tech-c:	AL325-WIX
upd-to:	asjl@lionra.net.nz
mnt-nfy:	asjl@lionra.net.nz
auth:	CRYPT-PW ap5rhmu6PnzI.
mnt-by:	MAINT-WIX-NZ
changed:	asjl@lionra.net.nz 20010831
source:	WIX

PGPKEY Example

key-cert: PGPKEY-691EB6E5
method: PGP
owner: Andy Linton <asjl@lionra.net.nz>
fingerpr: 673F 85DE 101A FFCC 691E B6E5
certif:
-----BEGIN PGP PUBLIC KEY BLOCK-----

gRAgAMBQI6rTj0BRsMAAAAAAoJEJDaZk.....

-----END PGP PUBLIC KEY BLOCK-----
mnt-by: MAINT-AL325-AP
changed: asjl@lionra.net.nz 20020610
source: WIX

Maintainer Object (updated)

mntner:	MAINT-AL325-AP
descr:	Lionra
descr:	NZ
admin-c:	AL325-WIX
tech-c:	AL325-WIX
upd-to:	asjl@lionra.net.nz
mnt-nfy:	asjl@lionra.net.nz
auth:	PGPKEY-691EB6E5
mnt-by:	MAINT-WIX-NZ
changed:	asjl@lionra.net.nz 20020803
source:	WIX

Person and Role Objects

- Person and role objects are for contact info
  - A person object contains information about a technical or administrative contact for an object
  - The role class is similar to the person class. However, instead of describing a human being, it describes a role performed by one or more human beings.

Person Object Example

person:	Andy Linton
address:	149 Cecil Road
address:	Wilton
address:	Wellington
country:	NZ
phone:	+64 4 970 1764
e-mail:	asjl@lionra.net.nz
nic-hdl:	AL325-AP
mnt-by:	MAINT-AL325-AP
changed:	asjl@lionra.net.nz 20011025
source:	WIX

Autonomous System Object

- Routing Policy Description object
- Most important components are
  - import
  - export
- These define the incoming and outgoing routing announcement relationships

Autonomous System Object Example

```
aut-num:        AS65518
as-name:        WIX-AS65518
import:         from AS4763
                action pref=90;
                accept AS4763
import:         from AS65335
                action pref=100;
                accept AS65335
export:         to AS4763
                        announce NOT ANY
export:         to AS65335
                        announce AS65518
```

Route object

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

Route Object Example

```
route:      130.195.0.0/16
descr:      Route for AS681
origin:     AS681
mnt-by:     MAINT-WIX-NZ
changed:    asjl@lionra.net.nz 20020628
source:     WIX
```

AS Set

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

AS Set Object Example

as-set: AS9439:AS-PRIVATE
descr: Private ASes for WIX
members: AS64512, AS64517, AS64525, AS64530,
AS64537, AS64543, AS64548, AS64553, AS64513,
AS64518, AS64526, AS64532, AS64538, AS64544,
AS64549, AS65025, AS64514, AS64520, AS64527,
AS64534, AS64539, AS64545, AS64550, AS65498,
AS64515, AS64521, AS64528, AS64535, AS64540,
AS64546, AS64551, AS65518, AS64516, AS64523,
AS64529, AS64536, AS64541, AS64547, AS64552,
AS65518, AS65531
mnt-by: MAINT-WIX-NZ
changed: asjl@lionra.net.nz 20020628
source: WIX

Route Set

- Collects routes together with similar properties

Route Set Object Example

route-set:	AS9439:RS-ROUTES:AS64515
descr:	Route set for AS64515
members:	203.97.237.120/29, 192.88.190.0/24 192.190.108.0/24
tech-c:	AL325-WIX
mnt-by:	MAINT-WIX-NZ
changed:	asjl@lionra.net.nz 20020628
source:	WIX

-

Autonomous System Object Example

```
aut-num:        AS9439
as-name:        WIX-AS9439
descr:          WIX Master AS
import:         from AS9439:AS-PRIVATE
                accept PeerAS^+
import:         from AS9439:AS-PUBLIC
                accept PeerAS
export:         to AS9439:AS-PRIVATE
                announce AS9439:AS-PRIVATE
                OR AS9439:AS-PUBLIC
export:         to AS9439:AS-PUBLIC
                announce ANY
```

Why use (RPSL) Policy?

- Consistent configuration between BGP peers (peers & customers)
- Expertise encoded in the tools that generate the policy rather than engineer configuring peering session
- Automatic, manageable solution for filter generation

Use of RPSL

- Use RtConfig v4 (part of IRRToolSet from RIPE) 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 :-)

RtConfig

- Version 4.0 supports RPSL (Latest version is 4.7.2 as at 4 August 2002)
- Generates Cisco, Bay's BCC, Juniper's Junos and Gated/RSd configurations
- Creates route and AS path filters.
- Can also create ingress/egress filters (Cisco only)

RtConfig options

-help
-version
-s <source-list>
-f <file name>
-config <config-format>
-supress-martian
-T [whois_query | whois_response | input | all]

RtConfig tips

Use command:

$ RtConfig -cisco_use_prefix_lists

Initialise Cisco list parameters

@RtConfig set cisco_map_first_no = 10
@RtConfig set cisco_map_increment_by = 10
@RtConfig set cisco_prefix_acl_no = 130
@RtConfig set cisco_aspath_acl_no = 130
@RtConfig set cisco_pktfilter_acl_no = 130
@RtConfig set cisco_community_acl_no = 30
@RtConfig set cisco_max_preference = 100

RtConfig Environment variables

IRR_HOST=xxx.net.nz
IRR_PORT=43
IRR_SOURCES=WIX

Overridden by command line options

RtConfig: Creating access lists

AS681 has one route object in the database:

route:	130.195.0.0/16
descr:	Route for AS681
origin:	AS681
mnt-by:	MAINT-WIX-NZ
changed:	asjl@lionra.net.nz 20020628
source:	WIX

RtConfig: Access List filter 1

```
$ RtConfig -h localhost -p 43 -s WIX
RtConfig> @RtConfig access_list filter AS681

no access-list 100
access-list 100 permit ip 130.195.0.0   0.0.0.0
   255.255.0.0   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
```

RtConfig: Access List filter 2

$ RtConfig -h localhost -p 43 -s WIX -supress_martian
RtConfig> @RtConfig access_list filter AS681

```
no access-list 100
access-list 100 deny   ip host 0.0.0.0 any
access-list 100 deny   ip 127.0.0.0 0.255.255.255 255.0.0.0 0.255.255.255
access-list 100 deny   ip 10.0.0.0 0.255.255.255 255.0.0.0 0.255.255.255
access-list 100 deny   ip 172.16.0.0 0.15.255.255 255.240.0.0 0.15.255.255
access-list 100 deny   ip 192.168.0.0 0.0.255.255 255.255.0.0 0.0.255.255
access-list 100 deny   ip 192.0.2.0 0.0.0.255 255.255.255.0 0.0.0.255
access-list 100 deny   ip 128.0.0.0 0.0.255.255 255.255.0.0 0.0.255.255
access-list 100 deny   ip 191.255.0.0 0.0.255.255 255.255.0.0 0.0.255.255
access-list 100 deny   ip 192.0.0.0 0.0.0.255 255.255.255.0 0.0.0.255
access-list 100 deny   ip 223.255.255.0 0.0.0.255 255.255.255.0 0.0.0.255
access-list 100 deny   ip 224.0.0.0 31.255.255.255 224.0.0.0 31.255.255.255
access-list 100 deny   ip 169.254.0.0 0.0.255.255 255.255.0.0 0.0.255.255
access-list 100 permit ip 130.195.0.0   0.0.0.0   255.255.0.0   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
```

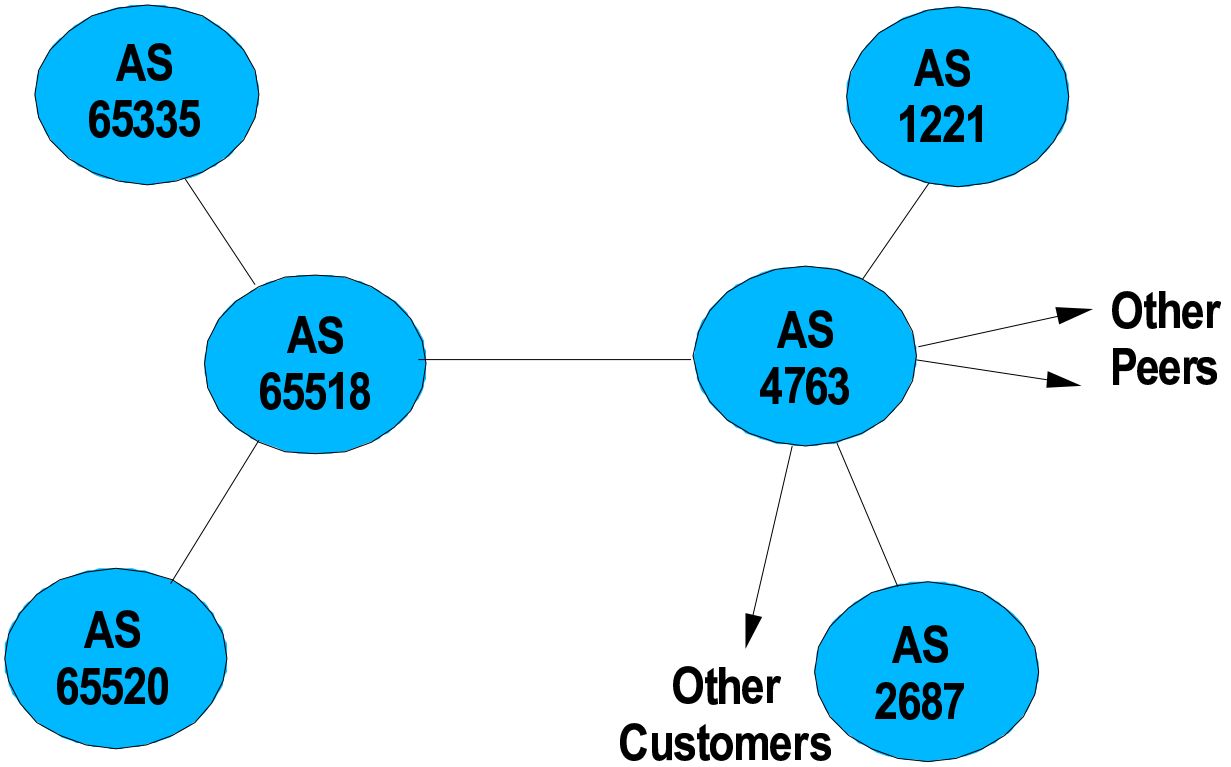
RtConfig: Access List filter 3

- Also works for Juniper routers

```
$ RtConfig -h localhost -p 43 -s WIX -config junos
RtConfig> @RtConfig access_list filter AS681

 policy-statement prefix-list-100 {
    term prefixes {
       from {
          route-filter 130.195.0.0/16 exact accept;
       }
    }
    term catch-rest {
        then reject;
    }
 }
```

Example Network



AS65518 Configuration

- AS65518 has a number of routing requirements
  - Static routes injected at customer edge
  - BGP peering with customers
  - BGP peering with peers
  - BGP peering with upstream providers
- Use RPSL and RtConfig in each case

Static route importation into BGP

- Use policy to filter static routes into BGP
  - Allows for martian filtering
  - AS path stuffing
  - Tagging routes with special communities
  - Other filtering, such as filter host routes

Static route importation (contd)

```
aut-num:    AS65518
import:     protocol STATIC into BGP4
            from AS65518
            action aspath.prepend (AS1);
            community.append (65518:1);
            accept AS65518:RS-STATIC
            AND NOT {129.168.10.0/24}
```

AS65518:RS-STATIC

route-set:	AS65518:RS-STATIC
descr:	Static routes for injection
member:	129.168.10.0/24,128.240.0.0/16
admin-c:	AL325-WIX
tech-c:	AL325-WIX
notify:	rpsl@lionra.net.nz
mnt-by:	MAINT-WIX-NZ
changed:	asjl@lionra.net.nz 20020205
source:	WIX

RtConfig commands for static import

```
RtConfig> @RtConfig set cisco_map_name = "AS%d-STATIC-
    EXPORT"
RtConfig> @RtConfig static2bgp AS65518 0.0.0.0

no ip prefix-list pl100
ip prefix-list pl100 permit 128.240.0.0/16
ip prefix-list pl100 deny 0.0.0.0/0 le 32
!
no route-map AS65518-STATIC-EXPORT
!
route-map AS65518-STATIC-EXPORT permit 1
 match ip address prefix-list pl100
 set as-path prepend 1
 set community  65518:1 additive
!
router bgp 65518
 redistribute static route-map AS65518-STATIC-EXPORT
```

Simple "Documentation" Policy

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

BGP peering with customers

- Static routes are OK for some customers
- Other customers have a BGP peering session

- Consider ISP who has two customers with AS64530, AS64536

BGP peering with customers (2)

```
aut-num: AS65518
import:    from AS64530
           action community.append (65518:4);
           accept AS64530
import:    from AS64536
           action community.append (65518:4);
           accept AS64536
export:    to AS64530
           announce ANY
export:    to AS64536
           announce ANY
```

AS65518:AS-CUSTOMERS

as-set:	AS65518:AS-CUSTOMERS
descr:	Lionra
members:	AS64530, AS64536
admin-c:	AL325-WIX
tech-c:	AL325-WIX
notify:	rpsl@lionra.net.nz
mnt-by:	MAINT-WIX-NZ
changed:	asjl@lionra.net.nz 20020129
source:	WIX

BGP peering with customers (2)

Now we can refine this to:

```
aut-num: AS65518
import:     from AS65518:AS-CUSTOMERS
            action community.append (65518:4);
            accept PeerAS
export:     to AS65518:AS-CUSTOMERS
            announce ANY
```

-

BGP4 Customer RtConfig

```
! config for CustomerA (AS64530) connection
!
@RtConfig set cisco_map_name = "AS%d-IMPORT"
@RtConfig import AS65518 203.79.121.54 AS64530
  192.168.5.13
!
@RtConfig set cisco_map_name = "AS%d-EXPORT"
@RtConfig export AS65518 203.79.121.54 AS64530
  192.168.5.13
```

Cisco Configuration

```
no access-list 100
access-list 100 permit ip 138.235.224.0   0.0.0.0   255.255.255.0
    0.0.0.0
access-list 100 permit ip 210.48.103.96   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 AS64530-IMPORT
!
route-map AS64530-IMPORT permit 1
 match ip address 100
 set community  65518:4 additive
!
router bgp 65518
neighbor 192.168.5.13 route-map AS64530-IMPORT in
```

Cisco Configuration (cont)

```
no route-map AS64530-EXPORT
!
route-map AS64530-EXPORT permit 1
!
router bgp 65518
neighbor 192.168.5.13 route-map AS64530-EXPORT out
```

Juniper Configuration

```
policy-options {
  community community-1 members [65518:4];
  policy-statement prefix-list-100 {
     term prefixes {
        from {
           route-filter 138.235.224.0/24 exact accept;
           route-filter 210.48.103.96/28 exact accept;
        }
     }
     term catch-rest {
         then reject;
     }
  }
```

Juniper Configuration (contd)

```
 policy-statement policy_65518_1 {
    term policy_65518_1-term-1 {
        from {
            policy prefix-list-100;
        }
        then {
            community add community-1;
            accept;
        }
    }

    term policy_65518_1-catch-rest {
        then reject;
    }
  }
}
```

Juniper Configuration (contd)

```
protocols {
  bgp {
    group peer-192.168.5.13 {
      type external;
      peer-as 64530;
      neighbor 192.168.5.13 {
        import policy_65518_1 ;
      }
    }
  }
}
```

Juniper Configuration (contd)

```
policy-options {
  policy-statement policy_65518_2 {
    term policy_65518_2-term-1 {
      from {
      }
      then {
        accept;
      }
    }

    term policy_65518_2-catch-rest {
      then reject;
    }
  }
}
```

Juniper Configuration (end)

```
protocols {
  bgp {
     group peer-192.168.5.13 {
        type external;
        peer-as 64530;
        neighbor 192.168.5.13 {
           export policy_65518_2 ;
        }
     }
  }
}
```

BGP peering with peers (1)

- Likely to be more complex than customer configurations
- This example uses AS Path Filters

```
aut-num:    AS65518
import:     from AS4763
            accept NOT { 0.0.0.0/0 }
            AND <^AS4763+ AS4763:AS-CUSTOMERS*$>
export:     to AS4763
            announce ANY
```

AS4763:AS-CUSTOMERS

as-set:	AS4763:AS-CUSTOMERS
descr:	TSCPNZ
members:	AS2687, AS4637, AS4740, AS9439, AS4770, AS6831, AS7657, AS7714, AS9345, AS9495, AS9559, AS9727, AS9731, AS9736, AS9887, AS9889, AS10131, AS17646, AS17705, AS17914
admin-c:	AL325-AP
tech-c:	AL325-AP
notify:	rpsl@lionra.net.nz
mnt-by:	MAINT-AL325-AP
changed:	asjl@lionra.net.nz 20020129
source:	APTEST

BGP4 Peering for AS65518

```
! config for PeerA (AS4763) connection
!
@RtConfig set cisco_map_name = "AS%d-IMPORT"
@RtConfig import AS65518 203.79.121.54 AS4763
    203.98.21.27
!
@RtConfig set cisco_map_name = "AS%d-EXPORT"
@RtConfig export AS65518 203.79.121.54 AS4763
    203.98.21.27
```

Cisco Config - import

```
no ip prefix-list pl132
ip prefix-list pl132 deny 0.0.0.0/0
ip prefix-list pl132 permit 0.0.0.0/0 le 32
!
no ip as-path access-list  130
ip as-path access-list 130 permit
   ^(_4763)+(_(2687|4740|9439|4770|6831|7657|7714|9345|94
   95|9559|9727|9731|9736|9887|9889|10131|17646|17705|17
   914))*$
!
no route-map AS4763-IMPORT
route-map AS4763-IMPORT permit 10
 match as-path 130
 match ip address prefix-list pl132
exit
```

Cisco Config – export

```
router bgp 65518
neighbor 203.98.21.27 route-map AS4763-IMPORT in
exit
!
no route-map AS4763-EXPORT
!
route-map AS4763-EXPORT permit 10
exit
!
router bgp 65518
neighbor 203.98.21.27 route-map AS4763-EXPORT out
exit
```

BGP peering with peers (2)

- Peering policy between peers need not be exactly the same

```
aut-num:    AS4763
import:     from AS4763:AS-PEERS
        action community.append(4763:4);
         accept PeerAS PeerAS:AS-CUSTOMERS
export:     to AS4763:AS-PEER
        announce AS4763 AS4763:AS-CUSTOMERS
```

AS65518:AS-CUSTOMERS

as-set:	AS65518:AS-CUSTOMERS
descr:	Lionra
members:	AS65520
admin-c:	AL325-AP
tech-c:	AL325-AP
notify:	rpsl@lionra.net.nz
mnt-by:	MAINT-AL325-AP
changed:	asjl@lionra.net.nz 20020129
source:	APTEST

AS65520 has three prefixes:

192.168.5.0/24, 192.168.7.0/24, 192.168.10.0/24

BGP4 Peering for AS4763

```
! config for Lionra connection
!
@RtConfig set cisco_map_name = "AS%d-IMPORT"
@RtConfig import AS4763 203.98.21.27 AS65518
    203.79.121.54
!
@RtConfig set cisco_map_name = "AS%d-EXPORT"
@RtConfig export AS4763 203.98.21.27 AS65518
    203.79.121.54
```

Cisco Config – import

```
no ip prefix-list pl130

ip prefix-list pl130 permit 192.168.5.0/24
ip prefix-list pl130 permit 192.168.7.0/24
ip prefix-list pl130 permit 192.168.10.0/24
ip prefix-list pl130 deny 0.0.0.0/0 le 32
!
no route-map AS65518-IMPORT
!
route-map AS65518-IMPORT permit 10
 match ip address prefix-list pl130
 set community  4763:4 additive
!
router bgp 4763
neighbor 203.79.121.54 route-map AS65518-IMPORT in
```

Cisco Config – export

```
no ip prefix-list pl131
ip prefix-list pl131 permit 130.195.0.0/16

! 174 lines of prefix-list deleted here

ip prefix-list pl131 deny 0.0.0.0/0 le 32
!
no route-map AS65518-EXPORT
!
route-map AS65518-EXPORT permit 10
match ip address prefix-list pl131
!
router bgp 4763
neighbor 203.79.121.54 route-map AS65518-EXPORT out
```

BGP peering with providers

- Similar to BGP peering with customers except that we're now the customer!
  - Policy should be similar to policy defined before
    - We send our routes to provider
    - We accept everything the provider sends

```
aut-num:    AS65518
import:     from AS65335
            action pref=100;
            accept ANY
export:     to AS65335
          announce AS65518 AS65518:AS-CUSTOMERS
```

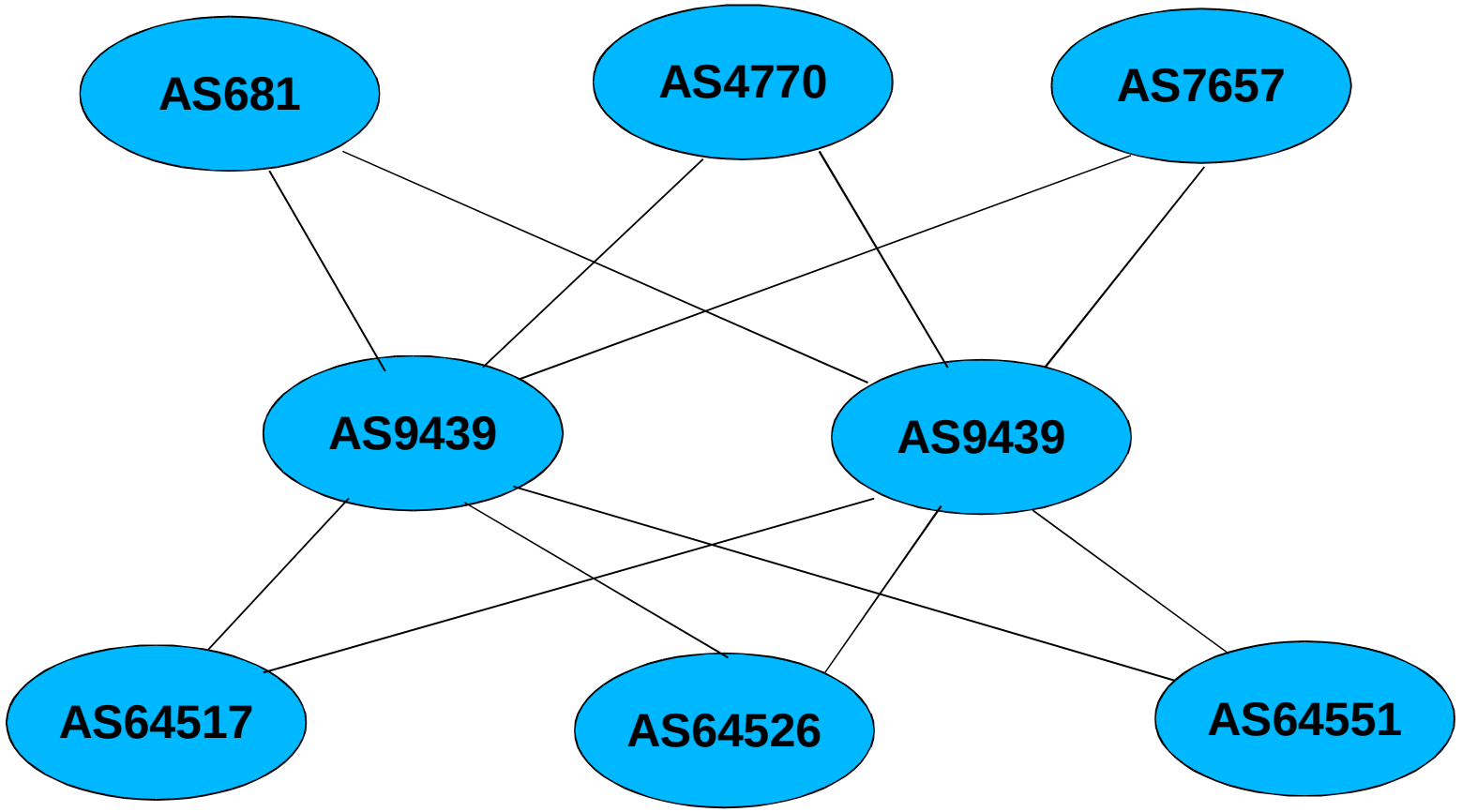
Problems?

- 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)
- Many are nervous about configuring their routers from external data. Compare this with virus updates!

Wellington Internet Exchange

- Distributed exchange running over Citylink
  - over 60 Km of fibre in city centre
  - approx 70 participants
  - extensive use of Linux routers with Zebra
  - many small players with no BGP clue
  - larger players wary because of lack of clue
- route reflectors need to implement policies to “make it safe”

WIX network (part of)



Preliminary work

- Because we have lots of Private AS numbers we have to run our own routing registry database
  - We chose ***irrd*** because our requirements are modest
- Create ***maintainer*** and ***person*** objects
- Set up PGP authentication
- Create ***aut-num*** objects for each AS
- Identify IP prefixes associated with each AS
  - Create ***route objects*** in database
- Create ***as-set*** objects where policy is common

AS9439 Configuration

- AS9439 has a relatively simple set of routing requirements
  - BGP peering with peers
  - Number of private AS = 42
  - Number of public AS = 20
  - AS9439 has no prefixes of its own!
- Use RPSL and RtConfig

AS9439 policies

```
aut-num:  AS9439
as-name: WIX-AS9439
descr:      WIX Master AS
import:      from AS9439:AS-PRIVATE
                accept PeerAS
import:      from AS9439:AS-PUBLIC
                accept PeerAS
export:      to AS9439:AS-PRIVATE
                announce AS9439:AS-PRIVATE
                              AS9439:AS-PUBLIC
export:      to AS9439:AS-PUBLIC
                announce AS9439:AS-PRIVATE
                              AS9439:AS-PUBLIC
```

AS9439:AS-PUBLIC

as-set: AS9439:AS-PUBLIC
descr: Public Ases for WIX
members: AS681, AS10022, AS17412, AS17792,
AS18119, AS4740, AS4768, AS4770, AS7657,
AS9325, AS9338, AS9436, AS9495, AS9503, AS9736,
AS9790, AS9872, AS9887
tech-c: AL325-WIX
notify: rpsl@lionra.net.nz
mnt-by: MAINT-WIX-NZ
changed: asjl@lionra.net.nz 20020612
source: WIX

AS9439:AS-PRIVATE

as-set: AS9439:AS-PRIVATE
descr: Private ASes for WIX
members: AS64512, AS64517, AS64525, AS64530,
AS64537, AS64543, AS64548, AS64553, AS64513,
AS64518, AS64526, AS64532, AS64538, AS64544,
AS64549, AS65025, AS64514, AS64520, AS64527,
AS64534, AS64539, AS64545, AS64550, AS65498,
AS64515, AS64521, AS64528, AS64535, AS64540,
AS64546, AS64551, AS65518, AS64516, AS64523,
AS64529, AS64536, AS64541, AS64547, AS64552,
AS65531
tech-c: AL325-WIX
source: WIX

AS64512 policies

```
aut-num:  AS64512
as-name:  WIX-AS64512
descr:    Citylink
admin-c:  AL325-WIX
tech-c:   AL325-WIX
import:   from AS9439
          accept ANY
export:   to AS9439
          announce AS64512
notify:   rpsl@lionra.net.nz
mnt-by:   MAINT-WIX-NZ
changed:   asjl@lionra.net.nz 20020610
source:    WIX
```

AS64512 prefixes

- AS64512 has these prefixes:

  210.86.11.236/30 210.48.103.144/28 210.48.103.136/29
  210.48.103.0/28 203.97.231.224/28 203.96.131.96/29
  203.79.85.80/29 203.109.154.32/28 203.109.148.24/29

- Note small address blocks that wouldn't normally be seen at an Internet Exchange
- This is not unusual on the WIX!

Software Tools (1)

- Cisco output from RtConfig ***almost*** works with Zebra
  - Use ***cisco2zebra*** filter to massage the output
  - It's a hack. The solution is to fix RtConfig
- Use ***mk-zebra*** to generate input for RtConfig processing
  - Input to ***mk-zebra*** looks like:

```
202.7.0.1:64512:Citylink
202.7.0.5:64546:Puskas
202.7.0.12:64526:CitylinkVoIP
```

Makefile

```
#
# $Id: Makefile,v 1.8 2002/07/05 04:44:41 asjl Exp $
#
IRR_HOST=cheviot.lionra.net.nz
IRR_PORT=43
IRR_SOURCES=WIX

Zico.cfg: Zico.master mk-zebra Makefile
        /home/asjl/NZNOG/mk-zebra < Zico.master > Zico.rpsl
        RtConfig -h $(IRR_HOST) -p $(IRR_PORT) \
                -s $(IRR_SOURCES) \
                -cisco_use_prefix_lists < Zico.rpsl \
                | /home/asjl/NZNOG/cisco2zebra > Zico.cfg
```

Software Tools (2)

- Tools hide complexity:

```
$ wc -l Zico.master Zico.rpsl Zico.cfg
      62 Zico.master
     755 Zico.rpsl
    3442 Zico.cfg
```

- Can use ***mk-junos*** to build Juniper configs if Juniper donate a router!

```
$ wc -l Zico.cfg-j
  5410 Zico.cfg-j
```

Software Tools (3)

- BGP naïve customers get a sample BGP configuration
  - Generated using ***mk-clients*** tool

Adding a new peer

- Register an AS in the WIX database
- Add routes for the new AS
- Add the new AS to AS9439:AS-PUBLIC or AS9439:AS-PRIVATE
- Add a {IP-address,AS-num,Description}-tuple to master config file
- Use Make to rebuild config file(s)

What Next?

- Run your own routing registry?
  - Decide which software to run
    - IRRd or RIPE v3
- Or register your routes in a public registry such as APNIC?
- Or both?
  - You may not want to reveal all your internal secrets!

What Next? (cont)

- Look at your customers, peers, providers and decide how to represent policy in RPSL
- Implement router configuration using RPSL and associated tools!

References

- Using RPSL in Practice - RFC 2650
- RPSL - RFC 2622
  - http://www.rfc-editor.org/rfcsearch.html
- IRRToolSet
  - http://www.ripe.net/ripencc/pub-services/db/irrtoolset/
- RPSL Training Page
  - http://www.isi.edu/ra/rps/training/
- RIPE database manual
  - http://www.ripe.net/ripe/docs/databaseref-manual.html

References (cont)

- RADB
  - http://www.merit.edu/radb/
- RIPE database software
  - ftp://ftp.ripe.net/ripe/dbase/software
- IRRd software
  - http://www.irrd.net/
- Zebra
  - http://www.zebra.org

Acknowledgements

- Mark Prior <mrp@iagu.net>
- Ambrose Magee <ambrose.magee@ericsson.com>
- Sid Jones <sid@tsnz.net>
- Simon Blake <simon@katipo.co.nz>
- APNIC
  - for asking me to present the tutorial

Contact Details

```
$ whois -h rpsl.apnic.net AL325-AP
person:      Andy Linton
address:    149 Cecil Road
address:     Wadestown
address:     Wellington
country:     NZ
phone:       +64 4 970 1764
e-mail:      asjl@lionra.net.nz
nic-hdl:     AL325-AP
mnt-by:      MAINT-AU-AL325-AP
changed:     asjl@lionra.net.nz 20010831
source:      APNIC
```