Introduction to RPSL

TorIX Meeting, September 2004

Joe Abley, jabley@isc.org

Agenda

- Some handwaving about why any of this is actually useful
- Architectural overview
- Incredibly brief history lesson
- Brief introduction to the language
- Examples using RtConfig
- Operational Considerations

Handwaving

Route Filtering is Good

- When people leak a full (or even slightly full) table to you, it hurts
  - ouch
- The problem isn’t really address hijacking or route theft (those are different problems)
  - the problem is that routers are often configured by crazed caffeine junkies at 4am

Route Filtering is Hard

- Filter lists can be long
  - too long for poor little cisco routers with their 1980s-era CPUs and tiny flash
    - hi Vince!
- Knowing what to put in filters can be difficult
- Keeping filters up-to-date can be difficult

Being Filtered Badly is Bad

- Announcing a new net for a customer (or a newly-allocated net from an RIR) relies on your peers and transit providers accepting it
- Getting transit providers' filters updated can take weeks
- Getting peers' filters updated sometimes never happens (or if it does, it's hard to tell that it has)

Sending Filter Updates to Mailing Lists is Annoying

- More precisely, reading those updates is annoying. Sending them is just futile, most of the time.
- Reading the follow-up messages from individual people saying "I've updated your filters" is even more annoying
- I am unusually grumpy, though, so this may not be an important motivation for you

Maximum-Prefix

- If all you want to do is protect yourself against tsunami-style leaks from peers, then setting a maximum-prefix limit is probably good enough
  - don’t forget martian filters, too
  - especially, deny 0.0.0.0/0

AS-PATH filters

- Applying AS-PATH filters is really no more convenient than prefix filters
  - the update frequency is a little lower, perhaps
- An AS-PATH filter will still allow you to receive a full leaked table if it has been cleaned through redistribution through an IGP (double ouch)

Panacea

- What we really need is a unified method for publishing the routes we want to announce:
  - never having to spam your peers to tell them to update their filters
  - being able to apply strict filters to all your peers to protect yourself from their after-hours routing explosions

Routing Registry Architectural Overview

Routing Policy

- We’re talking about BGP
- A description of the technical handling of BGP updates you receive *from* other people, and the BGP updates you send *to* others
  - not a philosophy on peering

RPSL

- Route Policy Specification Language
  - RFC 2622
- A language for describing routing policy
  - not a tool
  - not a database, not a whois server

Routing Registry

- A repository of route policy, expressed in RPSL
  - the interface for updating objects is quite often e-mail
  - the interface for retrieving objects is quite often whois
- There are lots of Routing Registries

Routing Arbiter Database (RADB)

- One particular Routing Registry operated by Merit Networks
- Was once free to use
  - is not free any more
  - but that's ok; there are free alternatives

Internet Routing Registry (IRR)

- Phrase invented by Merit Networks to describe a collection of Routing Registries
- Individual Routing Registries quite often mirror objects from other Routing Registries
  - (some don't, though)
- "IRR" tends to mean "the set of Routing Registries that Merit chooses to mirror"

Incredibly Brief History Lesson

History of RPSL

- RIPE-81, February 1993
- RIPE-181 (RIPE-81++), October 1994
- RFC 2622, “RPSL”, January 1998
- draft-blunk-rpslng-08, “RPSLng”, July 2004

Brief Introduction to the Language

Objects

- RPSL describes routing policy using collections of objects which have something to do with routing
  - aut-num (for things relating to ASes)
  - route (for routes)
  - and lots more

Security

- Each object has an associated maintainer object
- Each maintainer object has one or more authentication methods
- Authentication is required to update objects (and, in some cases, to add them)

No Security

- In most Routing Registries there is no assurance that route/aut-num data represents routes that are *allowed* to be announced
    - the RIPE registry is more advanced in this regard, for European routes
- In general, Routing Registry data is good for avoiding unintentional leaks, not intentional ones

A Few Examples

- Some brief examples here:
  - aut-num object
  - as-set object
  - route object
- See RPSL specification (also, query some registries for random ASes' policies) for more

The aut-num Object

- Contains a description of the import and export policies of an AS
- The expressions that can be used to describe route filtering is extensive
  - way too much to describe in detail here
  - we will just wave our hands a little bit

AS3557

```
aut-num:    AS3557
as-name:    ISC-CALIFORNIA
descr:      Internet Systems Consortium, Inc.
admin-c:    PV15-ARIN
tech-c:     SHS-ARIN
export:     to AS3557:AS-FLN  announce AS3557 AND {192.5.5.0/24}
export:     to AS-ANY  announce AS3557:AS-ISC
remarks:    contacts per RFC2142:
remarks:    Abuse / UCE reports  abuse@isc.org
remarks:    Security issues      noc@isc.org
notify:     noc^chat@isc.org
mnt-by:     MAINT-ISC
changed:    jabley@isc.org 20040101
source:     VERIO
```

The as-set Object

- The as-set object describes a set of AS numbers
  - can be used interchangeably with AS numbers in aut-num policy expressions
  - can include other as-sets
  - can be named hierarchically to avoid namespace collisions (e.g. AS3557:AS-FLN)

AS3557:AS-FLN

```
as-set:     AS3557:AS-FLN
descr:      F-Root Local Node ASes
members:    AS23710, AS30125, AS30122, AS23709, AS27322
members:    AS27318, AS27319, AS25572, AS23707, AS27320
members:    AS27313, AS27321, AS30124, AS30123
admin-c:    PV15
tech-c:     SHS
notify:     noc^chat@isc.org
mnt-by:     MAINT-ISC
changed:    jabley@isc.org 20031114
source:     VERIO
```

AS3557:AS-ISC

```
as-set:     AS3557:AS-ISC
descr:      ISC
members:    AS3557, AS112, AS3402, AS1280, AS9327
members:    AS30071, AS-BUNGI
admin-c:    PV15
tech-c:     SHS
notify:     noc^chat@isc.org
mnt-by:     MAINT-ISC
changed:    jabley@isc.org 20040430
source:     VERIO
```

The route Object

- Route objects are used to associate routes with origin ASes
    - AS numbers in import/export policies are shorthand for “all routes with this origin AS”
- You can also group collections of routes (including covering supernets with allowable prefix ranges) in route-set objects

192.5.5.0/24

```
route:      192.5.5.0/24
descr:      Internet Software Consortium
origin:     AS3557
remarks:    Covering route for F.ROOT-SERVERS.NET (192.5.5.241).
remarks:    Always originated from AS 3557, but part of a
remarks:    anycast deployment, and hence enjoys transit from
remarks:    many places. See http://f.root-servers.org/
notify:     noc^chat@isc.org
mnt-by:     MAINT-ISC
changed:    jabley@isc.org 20030925
source:     OTTIX
```

“What’s your AS macro?”

- AS macro is the old (RIPE-181) name for “as-set”
- What people are really asking is “what expression should I put on the import line in my aut-num object?”
    - the convention in some circles is to standardise all their import expressions to something like `import: from ASxxxx AS-something`

Examples using RtConfig

RtConfig

- RtConfig is part of the IRRToolset
  - was once called the RAToolset
  - originally developed at ISI, then at RIPE, and (shortly) at ISC

RtConfig

- RtConfig is a macro expansion tool that replaces tokens in its input stream with router config bits derived from RPSL
- Easiest to appreciate what it is good for by looking at examples, rather than endless tedious slides
  - the manual page is good

Where to Get It

- Currently hosted at RIPE
  - see link in references section at end of this slide set
- It’s in FreeBSD’s ports collection
  - net-mgmt/irrtoolset

Example

- Suppose you were peering with AS 3557 at the PAIX in Palo Alto, and you wanted to apply a prefix filter to the session

AS3557 reminder

```
aut-num:    AS3557
as-name:    ISC-CALIFORNIA
descr:      Internet Systems Consortium, Inc.
admin-c:    PV15-ARIN
tech-c:     SHS-ARIN
export:     to AS3557:AS-FLN  announce AS3557 AND {192.5.5.0/24}
export:     to AS-ANY  announce AS3557:AS-ISC
remarks:    contacts per RFC2142:
remarks:    Abuse / UCE reports  abuse@isc.org
remarks:    Security issues      noc@isc.org
notify:     noc^chat@isc.org
mnt-by:     MAINT-ISC
changed:    jabley@isc.org 20040101
source:     VERIO
```

Building a Filter...

```
[jabley@felix]% RtConfig
RtConfig> @RtConfig access_list filter AS3557:AS-ISC
!
no access-list 100
access-list 100 permit ip 128.177.0.0   0.0.0.0   255.255.0.0   0.0.0.0
access-list 100 permit ip 128.177.247.0   0.0.0.0   255.255.255.0   0.0.0.0
access-list 100 permit ip 149.20.0.0   0.0.0.0   255.255.0.0   0.0.0.0
access-list 100 permit ip 168.61.0.0   0.0.0.0   255.255.0.0   0.0.0.0
access-list 100 permit ip 192.5.4.0   0.0.1.0   255.255.254.0   0.0.1.0
access-list 100 permit ip 192.83.249.0   0.0.0.0   255.255.255.0   0.0.0.0
[.... lines omitted to fit in slide... ]
access-list 100 permit ip 209.133.38.0   0.0.0.0   255.255.255.0   0.0.0.0
access-list 100 permit ip 209.133.117.0   0.0.0.0   255.255.255.0   0.0.0.0
access-list 100 permit ip 209.249.2.0   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
RtConfig>
```

Using Prefix Lists...

```
[jabley@felix]% RtConfig -cisco_use_prefix_lists
RtConfig> @RtConfig access_list filter AS3557:AS-ISC
!
no ip prefix-list pl100
ip prefix-list pl100 permit 128.177.0.0/16
ip prefix-list pl100 permit 128.177.247.0/24
ip prefix-list pl100 permit 149.20.0.0/16
ip prefix-list pl100 permit 168.61.0.0/16
ip prefix-list pl100 permit 192.5.4.0/23 le 24
ip prefix-list pl100 permit 192.83.249.0/24
[.... lines omitted to fit in slide... ]
ip prefix-list pl100 permit 209.133.38.0/24
ip prefix-list pl100 permit 209.133.117.0/24
ip prefix-list pl100 permit 209.249.2.0/24
ip prefix-list pl100 deny 0.0.0.0/0 le 32
RtConfig>
```

For Juniper Routers...

```
[jabley@felix]% RtConfig -config junos
RtConfig> @RtConfig access_list filter AS3557:AS-ISC
   policy-statement prefix-list-100 {
      term prefixes {
         from {
            route-filter 128.177.0.0/16 exact accept;
            route-filter 128.177.247.0/24 exact accept;
            route-filter 149.20.0.0/16 exact accept;
            route-filter 168.61.0.0/16 exact accept;
            route-filter 192.5.4.0/23 upto /24 accept;
            route-filter 192.83.249.0/24 exact accept;
[.... lines omitted to fit in slide... ]
            route-filter 209.133.38.0/24 exact accept;
            route-filter 209.133.117.0/24 exact accept;
            route-filter 209.249.2.0/24 exact accept;
         }
      }
      term catch-rest {
          then reject;
      }
   }
```

Lots of Options

- RtConfig has many, many options to help tailor it to individual router configuration styles
  - names of prefix lists, access lists, route maps
- Can represent quite complicated policies
  - route redistribution, for example

Script-Friendly

- Examples on previous slides showed RtConfig being run from the command line, but it can also be run from scripts
  - originally conceived as a tool which could produce entire router configs

Operational Considerations

Choice of Routing Registry

- You can run your own
  - you can mirror other peoples’
  - you can restrict access to your data
  - you can not rely on the network to configure the network
- You can use other peoples’ registries directly

Publishing Data

- Choice of registry for publishing your own data depends on what registries your neighbour ASes are happy to query
- Reasonable choices:
  - registries run by your provider
  - registries mirrored by Merit (RIPE, ALTDB, etc)

Retrieving Data

- If your peers are publishing their routing policy in a Routing Registry, you need to know which one to retrieve it from
    - you can run your own registry and mirror the external databases that you need
    - you can insist that people use one registry in particular (works well if you are Big and Important)

Provisioning Strategy

- The principal value of retrieving policy data from a Routing Registry is to be able to make it easier to configure routers
- You need to engage in serious beard-stroking before you decide that letting scripts update your live network is a good idea, however

Possible Approaches

- Use RPSL data to generate filter lists and apply them by hand
- Generate filter lists automatically and generate e-mail (or other beeping) when the config in the routers differs from the config generated from the RPSL
    - rancid, RtConfig

Publishing Private Data

- You can express details of your routing policy in RPSL that are not normally visible to the outside world
  - use of community string attributes, local preference, treatment of MED, etc
- You don't have to, however: you only need to publish the bits that are useful to peers

Further Reading

RPSL

- RFC 1786 (RIPE-181)
- RFC 2622 (RPSL)
- RFC 2650 (Using RPSL in Practice)
- http://www.irr.net/docs/rpsl.html

Routing Registries

- http://www.irr.net/
- http://www.ra.net/
- http://www.ripe.net/ripe/docs/databaseref-manual.html

IRRToolSet

- http://www.ripe.net/ripencc/pub-services/db/irrtoolset/index.html
- http://www.isc.org/something/soon

Routing Registry Software

- IRRd, by Merit
  - http://www.irrd.net/
- RIPE whois server
  - ftp://ftp.ripe.net/ripe/dbase/software/

Questions?

jabley@isc.org