

Using RPSL

to generate config templates

Lutz Donnerhacke

IKS Service GmbH

Routing Policy Specification Language (RPSL)

- RFC 2280, RFC 4012 (IPv6), RFC 7909 (rPKI)
- 1999, the year of IOS 11.x
- Document *real world* config in databases
- Problems
 - (Router) Software evolves quicker than standards
 - Standards evolve quicker than (processing) software
- Consequences
 - becomes artificial, incomplete
 - Documentation by *remarks*

Why?

- *Explain* your peering concepts to colleagues and partner
 - Avoid verbal discussion and human interaction
 - Clear and concise, minimizing errors
- Explain your peering *concepts* to colleagues and partner
 - Highlight common parts (peerings, up/downlinks, communities)
 - Define schemata easy to extend and memorize
- *Automate* processing
 - Update peering policies directly from RADB
 - Avoid unnecessary announcements by validate peer's policy, too

Basics

- from X action Y; accept Z / to X action Y; announce Z
 1. For each *route*, which *matches* Z (not peer specific!)
 2. Apply *route-map* action Y
 3. Add this route-map to peer X
- Only positive match, no notation for negation (no no-no)
 - from AS123 accept AS123 and $\langle ^AS123+\$ \rangle$
from AS-ANY accept ASPeer $^{-24}$
 - will accept routes from AS123 even if they do not match first rule
- Routes selected by AS... (^n-m)
- Paths selected by regex $\langle AS... \rangle$

RPSL Lego

- a EXCEPT b
 - Match b first
then only routes which not match b, are tried on a
 - Problem: *Not specific* to the peer, only to the route
- a REFINE b
 - Match a, take actions
match b, take actions
 - Useful for *generic actions*
- Matching is right associative, but actions work from left to right

Example: BLACKHOLE

```
protocol MPBGP into static
  afi ipv4.unicast {
    to AS199284 action next-hop = 127.6.6.6;
    announce communtiy(65535:666);
  } REFINE afi ipv6.unicast {
    to AS199284 action next-hop = fd9d:4778:4316::666;
    announce communtiy(65535:666);
  }
```

Example: community based prepending

```
... REFINE afi any {  
    to AS-ANY action aspath.prepend(AS199284)  
        announce community(64629:PeerAS)  
    to AS-ANY action aspath.prepend(AS199284, AS199284)  
        announce community(64630:PeerAS)  
} REFINE afi any {  
    to AS-ANY announce community(64628:PeerAS, 64628:0)  
    to AS-ANY announce not community(64628:PeerAS)  
}
```

Example: Input sanitization (1)

```
afi any { # prevent injection of internal communities
    from AS-ANY action community.delete(64628:10, ...); accept ANY;
} REFINE afi any { # GSHUT
    from AS-ANY action pref = 65535; accept community(65535:0);
    from AS-ANY action pref = 65435; accept ANY;
} REFINE afi any { # spoofing
    from AS-ANY accept NOT AS199284^+;
} REFINE afi ipv4 { # reserved
    from AS-ANY accept NOT fltr-martian;
}
```


Example: Input sanitization (2)

```
} REFINE afi ipv4 {  
    # BLACKHOLE only hosts, otherwise up to /24  
    from AS-ANY accept { 0.0.0.0/0^1-24 } AND NOT community(65535:666);  
    from AS-ANY accept { 0.0.0.0/0^32 } AND community(65535:666);  
}  
  
} REFINE afi ipv6 {  
    # BLACKHOLE only LANs or hosts, otherwise up to /64  
    from AS-ANY accept { 2000::/3^4-48 } AND NOT community(65535:666);  
    from AS-ANY accept { 2000::/3^64-128 } AND community(65535:666);  
}
```

Software

- Too old (even for me)
- Unable to cover IPv6 (try to enumerate all IPs)
 - Segfault after 2h while evaluating „2001::/3[^]-48“
- Incomplete support of RFC (RIPE)
 - RPSL-parser does not accept all valid aut-num objects
- Extension support missing
 - Generic method to introduce i.e. large communities
 - Only known and basic extensions are implemented

Questions?

How to import from OSPF into BGP?

Can I *aggregate on allocations* instead on route-objects to shorten ACLs?

What the heck are you doing?