(RPS) Routing Policy System Working Group

Chairs:

Cengiz Alaettinoglu <cengiz@isi.edu>

Daniel Karrenberg <Daniel.Karrenberg@ripe.net>

Mailing List: `rps@isi.edu`

To subscribe: `rps-request@isi.edu`

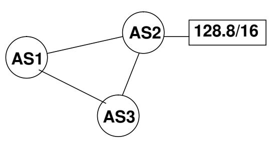
Ftp archive: `ftp://ftp.isi.edu/rps`

Autonomous System Path Expression Extension to Ripe-181

Cengiz Alaettinoglu

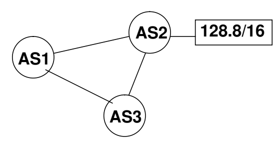
Jessica Yu

Autonomous System Objects and Import Policies




```
aut-num: AS1
as-name: foo
as-in: from AS2  1 accept {128.8.0.0/16}
as-in: from AS3  2 accept {128.8.0.0/16}
```


Autonomous System Objects and Export Policies




```
aut-num: AS2
as-name: bar
as-out: to AS1  announce {128.8.0.0/16}
as-out: to AS3  announce {128.8.0.0/16}
```


Route Objects Originating AS, Communities in Policy

```
route:     128.8.0.0/16
descr:     University of Maryland
origin:    AS86
comm-list: COMM_NSFNET

route: 192.101.80.0/24
descr: WVWC
origin: AS86

as-in:  from AS2 1 accept AS86
as-out: to   AS3 announce COMM_NSFNET
```


AS-Macro Objects

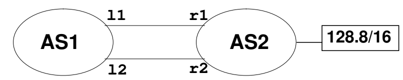
```
as-macro:  AS-ANSLA
descr:     List of ANS LA ASes
as-list:   AS1322 AS1671 AS2930

as-in:  from AS2 1 accept AS-ANSLA
```

Logical Combination

```
as-in: from AS2 1 accept
       AS-ANSLA AND NOT AS1322 OR {128.8.0.0/16}
```


Multiple Interconnections




```
aut-num: AS1
as-in: from AS2 1 accept {128.8.0.0/16}
interas-in: from AS2 l1 r1 (pref=1) accept {128.8.0.0/16}
interas-in: from AS2 l2 r2 (pref=2) accept {128.8.0.0/16}
interas-out: to AS2 ...
```


AS Path Expressions in Policy

Many providers have AS path based policies:

- **as-in: ... accept AS1 is not in the AS path**
- **as-out: ... announce AS1 is the first AS in the AS path**

Need AS Path Expressions!

AS Path Regular Expressions

`ASn`	"ASn"
`.`	any AS number
`[AS1 AS2 AS4-AS6]`	AS1, AS2, AS4, AS5, AS6
`[^AS1 AS2 AS4-AS6]`	AS3, AS7, AS8, ...
`^`	beginning of an AS path
`$`	end of an AS path
`exp* exp+ exp?`	0 or more, 1 or more, 0 or 1
`exp1 exp2`	`catenation`
`exp1 | exp2`	`alternative`
`()`	

AS Path Expression

< `AS-Path-Regular-Expression` >

matches routes whose AS paths are matched by the `AS-Path-Regular-Expression`

AS Path Expressions in Policy

```
aut-num: AS1
as-in: from AS2   1 accept NOT <AS1>
as-in: from AS2   1 accept <^AS1$>
as-out: to AS2 announce <^[AS1 AS3 AS4 AS5]>
```

Existing tools will break!

Extended Attributes

Original attributes:	Extended attributes:
as-in	extended-as-in
as-out	extended-as-out
interas-in	extended-interas-in
interas-out	extended-interas-out

Valid Policies: Original Attributes Only

```
as-in:  from AS2 1 accept AS86
as-out: to   AS3 announce COMM_NSFNET
```

old tools: will work

new tools: will work

Valid Policies: Extended Attributes Only

```
extended-as-in:  from AS2 1 accept <AS8$>
extended-as-out: to  AS3 announce <^[AS1-AS9]>
```

old tools: will assume default policy “NOT ANY”

new tools: will work

Valid Policies: Mixing Rule

on a per peer basis

- if policies are mixed
  - for *-in policies
    - for each extended-as-in specify as-in
    - for each extended-interas-in specify interas-in
  - for *-out policies
    - for each extended-as-out specify as-out
    - for each extended-interas-out specify interas-out

Tools and Mixed Policies

Old tools: use original attributes only

New tools: on import

- use extended attributes if exist
- use original attributes otherwise

on export

- use extended attributes if exist
- use original attributes otherwise

Eventually all tools will be upgraded!

Valid Examples

```
aut-num: AS1
as-in: from AS2 1 accept ANY
as-out: to AS2 announce NOT AS10
extended-as-in: from AS2 1 accept NOT <AS5>
extended-as-out: to AS2 announce NOT <AS1.* AS5.* AS10>
```

```
aut-num: AS1
as-in: from AS2 1 accept AS2
as-out: to AS2 announce AS1
extended-as-out: to AS2 announce <^[AS3 AS4 AS5]> OR AS1
```

Invalid Example

```
aut-num: AS1
as-in: from AS2 1 accept ANY
extended-as-in: from AS2 1 accept ANY
extended-as-out: to AS2 announce NOT <AS1 . * AS5.* AS10>
```

There should be an `as-out` attribute.

Conclusions

AS path expressions can now be specified

To avoid breaking tools:

extended-as-in

improved-extended-as-in

super-improved-extended-as-in

yet-another-super-improved-extended-as-in

Extensibility should be part of RPSL!