RPSL Policy Terms

Cengiz Alaettinoglu

Policy Term

event name	`import`
filter	`{128.8.0.0/16}`
event parameters	`AS2 * *`
action	`pref = 1`

Event Names

Can be protocol specific

For BGP/IDRP

`import`	preference policies
`export`	access restriction policies
`aggregate`	aggregation rules
`inject-static`	for injecting static routes
`...`	

Filters

logical combination (`AND, OR, NOT`) of

- address-prefix sets
- originating AS numbers
- Ripe-181 communities
- AS macros
- AS path expressions

More complex filters can be defined in the dictionary!

```
{128.8.0.0/16} OR AS-ANS AND NOT AS690
               OR <^AS1> OR color.equals(3)
```


Event Parameters

Depends on the event

For `import` and `export`:

`peer-as-no`	AS number or *
`local-interface-address`	IP address or *
`remote-interface-address`	IP address or *

Allowable Event Parameter Combinations for `import` and `export`

peer-as-no	local-interface	remote-interface
*	*	*
AS1	*	*
AS1	x.x.x.x	*
AS1	x.x.x.x	x.x.x.y

More specific match

Actions

set/modify/add/delete attributes and variables

```
action pref = 1;
action MED = 2 * IGP + 5;
action community = 10;
```

More complex actions can be defined in the dictionary!

Interaction Between Policy Terms

```
import filter AS2 AND HEPNET
       parameters AS1 * *
       action pref = 3;
import filter AS2 AND HEPNET
       parameters AS1 x.x.x.x *
       action pref = 2;
import filter AS2 AND HEPNET
       parameters AS1 x.x.x.x x.x.x.y
       action pref = -1;
```

Need a rule that tells which action to perform!

Overrides Rule

Perform the most specifically matching policy terms:

	filter	parameters	action
(1)	AS1 AS2 AS3 AS4	AS1 * *	pref = 0;
(2)	AS1	AS1 x.x.x.x x.x.x.y	pref = 1;
(3)	AS2	AS1 x.x.x.x x.x.x.y	pref = 2;
(4)	AS4	AS1 z.z.z.x z.z.z.y	pref = 4;

peering	AS1	AS2	AS3	AS4
x.x.x.x x.x.x.y	1	2		
z.z.z.x z.z.z.y				4
w.w.w.x w.w.w.y	0	0	0	0

Computation

Search over policy terms

for the most specifically matching policy terms

Linear in number of policy terms!

Complements Rule

Perform less specific action followed by more specific action:

	filter	parameters	action
(1)	AS1 AS2 AS3 AS4	AS1 * *	pref = 0;
(2)	AS1	AS1 x.x.x.x x.x.x.y	pref = 1;
(3)	AS2	AS1 x.x.x.x x.x.x.y	pref = 2;
(4)	AS4	AS1 z.z.z.x z.z.z.y	pref = 4;

peering	AS1	AS2	AS3	AS4
x.x.x.x x.x.x.y	1	2	0	0
z.z.z.x z.z.z.y	0	0	0	4
w.w.w.x w.w.w.y	0	0	0	0

Computation

```
less specifics:
   {(AS1 AS2 AS3 AS4, pref=0;)}
more specifics:
   {(AS1, pref=1;), (AS2, pref=2;)}
add difference:
   {(AS1, pref=1;), (AS2, pref=2;), (AS3 AS4,;)}
cartesian product:
   {(AS1 AND (AS1 AS2 AS3 AS4), pref=0; pref=1;),
    (AS2 AND (AS1 AS2 AS3 AS4), pref=0; pref=2;),
    ((AS3 AS4) AND (AS1 AS2 AS3 AS4), pref=0;;)}
equals
   {(AS1, pref=1;), (AS2, pref=2;), (AS3 AS4, pref=0;)}
```

Quadratic in number of policy terms!

Ripe-181 Rule

Perform more specific and the leftover policy terms

leftover policy term

less specifics minus terms w/ peer-as-no match

```
(1) AS1 AS2 AS3 AS4    AS1 * *                 pref = 0;
(2) AS1                AS1 x.x.x.x x.x.x.y     pref = 1;
(3) AS2                AS1 x.x.x.x x.x.x.y     pref = 2;
(4) AS4                AS1 z.z.z.x z.z.z.y     pref = 4;
```

peering	AS1	AS2	AS3	AS4
x.x.x.x x.x.x.y	1	2	0	
z.z.z.x z.z.z.y			0	4
w.w.w.x w.w.w.y			0	

Computation

```
less specifics:
   {(AS1 AS2 AS3 AS4, pref=0;)}
more specifics:
   {(AS1, pref=1;), (AS2, pref=2;)}
add leftover:
   {(AS1, pref=1;), (AS2, pref=2;), (AS3,pref=0;)}
cartesian product:
   {(AS1 AND (AS1 AS2 AS3 AS4), pref=(0,1);),
    (AS2 AND (AS1 AS2 AS3 AS4), pref=(0,2);),
    (AS3 AND (AS1 AS2 AS3 AS4), pref=(0,0);)}
equals
   {(AS1, pref=(0,1)),(AS2, pref=(0,2)),(AS3,pref=(0,0))}
```

Quadratic in number of policy terms!

Specification Order Rule

Perform the first specified policy term

filter parameters action

```
(1) AS1 AS2 AS3 AS4    AS1 * *                 pref = 0;
(2) AS1                AS1 x.x.x.x x.x.x.y     pref = 1;
(3) AS2                AS1 x.x.x.x x.x.x.y     pref = 2;
(4) AS4                AS1 z.z.z.x z.z.z.y     pref = 4;
```

peering	AS1	AS2	AS3	AS4
x.x.x.x x.x.x.y	0	0	0	0
z.z.z.x z.z.z.y	0	0	0	0
w.w.w.x w.w.w.y	0	0	0	0

Computation

Search over policy terms

for the first matching policy terms

Linear in number of policy terms!

Comparison

Rule	Expressibility	Simplicity	Computation
overrides	same	simple	linear
complements	same		quadratic
Ripe-181	same	complex	quadratic
spec. order	same	simple	linear

Proposed Interaction Rule

Modified Policy Term

event name
filter
event parameters action
event parameters action
event parameters action
. . .

Rule:

within a policy term: overrides rule

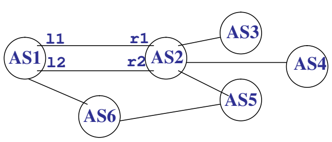
across policy terms: specification order rule

Example

```
import filter AS1
    parameters AS1 x.x.x.x x.x.x.y    action pref = 1;
import filter AS1 AS2
    parameters AS1 * *                action pref = 2
    parameters AS1 z.z.z.x z.z.z.y    action pref = 3;
import filter AS2 AS3 AS4
    parameters AS1 * *                action pref = 4
    parameters AS1 x.x.x.x x.x.x.y    action pref = 5;
```

peering	AS1	AS2	AS3	AS4
x.x.x.x x.x.x.y	1	2	5	5
z.z.z.x z.z.z.y	3	3	4	4
w.w.w.x w.w.w.y	2	2	4	4

Example




```
import filter AS2 AS4
   parameters AS2 * *     action pref = 1
   parameters AS6 * *     action pref = 2;
import filter AS3
   parameters AS2 l1 r1   action pref = 1
   parameters AS2 l2 r2   action pref = 2
   parameters AS6 *  *    action pref = 3;
```


Computation

search for most specific within a policy term

search for first match across policy terms

Linear in number of policy terms!

Conclusions

Proposed rule:

- **simple**
- **convenient**
- **expressive**
- **linear computation**
- **actions are not composed**