Network Working Group
Internet-Draft
Expires: June 18, 2003

J. Damas

F. Parent
Viagenie

A. Robachevski
RIPE NCC

December 18, 2002

RPSLng
draft-damas-rpslng-00.txt

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on June 18, 2003.

Copyright Notice

Abstract

This memo presents a new set of simple extensions to the RPSL language enabling the language to document routing policies for the IPv6 and multicast address families currently used in the Internet.

1. Introduction

RFC 2622 [1] defines the RPSL language for the IPv4 unicast routing protocols and a series of guidelines for extending the language itself.

This document proposes to extend RPSL according to the following goals and requirements:

- provide RPSL extensibility in the dimension of address families. Specifically, to allow users to document routing policy for ipv6 and multicast.

- the extensions must be backwards compatible and minimise risk of breaking existing tools. For instance, introducing a new class or attribute will less probably break the tools than would changing the format of an existing attribute. Section 10 of RFC2622 provides guidelines.

- clarity and non-ambiguity: RPSL information is used by software tools and by humans.

- minimise duplication of information, particularly when routing policies for different address families are the same.

- Internet Routing Registry (IRR) system requirements: It is impossible to consider RPSL extensions as a pure language modification. The capabilities and established operational practices the users are familiar with when interacting with the servers supporting IRR must also be taken into account.

An important point is to note the fact that there are two address families, corresponding to the two versions of the IP protocol currently in use in the Internet, but there are at least four distinct routing policies that need to be described (IPv4 {unicast|multicast}, IPv6 {unicast|multicast}).

2. Specifying routing policy for different address families

Routing policy is currently specified in the aut-num class using "import:"and "export:" attributes. Sometimes it is important to distinguish policy for different address families, as well as a unicast routing policy from a multicast one.

Use of existing import and export attributes is not a good option since itbreaks backward compatibility and could undermine clarity in the expressions.

Keeping this in mind, the "import:" and "export:" attributes implicitly specifyipv4 unicast policy and remain as defined previously in RPSL and new multi-protocol (mp) attributes are introduced. These will be described below.

2.1 The afi dictionary attribute

In this section we introduce a new dictionary attribute:

Address family, <afi>, is an RPSL list of address families for which the policy expression should be evaluated. <afi> is mandatory within the new mp attributes introduced in this document.

The possible values for <afi> are:

```
ipv4
ipv4.unicast (equivalent to ipv4)
ipv4.multicast
ipv6
ipv6.unicast (equivalent to ipv6)
ipv6.multicast
```

Appearance of these values in an attribute's value must be preceded by the keyword afi.

An <afi-list> is defined as a comma separated list of one or more afi values.

2.2 mp-import and mp-export

Three new policy attributes are introduced:

```
mp-import:
mp-export:
mp-default:
```

These attributes incorporate the afi (address-family) specification.

The definition of the "mp-import:" attribute is as follows:

```
mp-import ::=
             [protocol <protocol1>] [into <protocol2>] <importexpression>



<importexpression> ::=
       afi <afi-list> <import-term> accept <filter> |
       afi <afi-list> <import-term> accept <filter> except
                                              <importexpression> |
       afi <afi-list> <import-term> accept <filter> refine
                                              <importexpression>



<import-term> ::=  <import-factor> [
                   <import-factor>
                   ...
                   <import-factor>]


<import-factor> ::= from <peering> [action <action>];
```

The <peering> specification indicates the AS (and the router if present)

```
<peering> ::= <as-expression> [<router-expression-1>]
                           [at <router-expression-2>] |
              <peering-set-name>
```

with <router-expression-1> and <router-expression-2> being expressions over router IPv4 or IPv6 addresses (specifying their address family with the use of the appropriate "afi <afi>" term), inet-rtr names, and rtr-set names using operators AND, OR, and EXCEPT.

In the same manner the <filter> expression is the extension of the RPSL <filter> expression [section 5.4 of RFC2622], requiring the presence of an "afi <afi>" term before each address or address-prefix set.

The address family may be specified at any level of nesting of

<importexpression>, and is valid only within the <importexpression> that contains it.

Therefore in the example

```
aut-num:    AS65534
mp-import: afi ipv6.unicast,ipv4 from AS1 action pref = 1; accept as-foo
            except { afi ipv6.unicast,ipv4
            from AS2 action pref = 2; accept AS226
               except { afi ipv6.unicast
               from AS3 action pref = 3; accept {3FFE:FFFF::/35}
                      }
                   }
```

the last (rightmost) "except" is evaluated only for the ipv6 unicast address family, while other import-expressions are evaluated for both the ipv6 and ipv4 unicast address families.

The evaluation of an <importexpression> is done by evaluating all of its components. Evaluation of peering-sets and filter-sets is constrained by the address family. Such constraints may result in a {NOT ANY} <filter> or invalid <peering> depending on implicit or explicit definitions of the address family in the set. In the latter case an error is returned. {NOT ANY} filter may issue a warning.

Conflicts with explicit or implicit declarations are resolved at runtime, that is during evaluation of a policy expression. For example, when evaluating the following import policy:

```
aut-num: AS2
mp-import: afi ipv6 from AS1 accept {193.0.0.0/22}
```

the filter should be evaluated as {NOT ANY}.

```
aut-num: AS2
mp-import: afi ipv6.unicast {
   from AS-ANY action med = 0; accept {3FFE:FFFF::/35};
   } refine { afi ipv6.unicast
        from AS1 at 3FFE:FFFF::1 action pref = 1; accept AS-UPSTREAM;
        from prng6-ebgp-peers action pref = 2; accept AS1;
     }
```

In this example only ipv6 prefixes originated by AS1 will be collected, and while evaluating AS-UPSTREAM, an as-set, only ipv6 prefixes of the member ASes will be considered.

Export policy is specified in the mp-export attribute. The mp-export attribute is defined in a symmetric way to the mp-import attribute.

The "mp-default:" attribute is defined as

mp-default: <peering> [action <action>] [networks <filter>]

using the definitions above for <peering> and <filter>

2.3 Additional values for <protocol>

Two new additional values are possible for <protocol> specification:

BGP4+
MBGP

both support the same options available for the BGP4 value.

3. New classes and attributes to support the extensions

3.1 as-set Class

The as-set class defines a set of Autonomous Systems (AS), specified either directly by listing them in the members attribute, or indirectly by referring to another as-sets or using the mbrs-by-ref facility. More importantly, "In a context that expects a route set (e.g. members attribute of the route-set class), [...] an as-set AS-X defines the set of routes that are originated by the ASes in AS-X.", [section 5.3 of RFC2622].

The as-set class is therefore used to collect a set of route prefixes, which may be restricted to a specific address family.

The existing as-set class does not need any modifications. The evaluation of the class must be filtered to obtain prefixes belonging to a particular address family using the traditional filtering mechanism in use in IRR systems today.

3.2 route6 Class

An ipv6 inter-AS route has specific properties, such as prefix format, storage requirements that are different from the existing route class.

Additionally, IRR systems use filters to select which type of information is returned to the requester. These filters are designed to operate by receiving a class type as operand. In the case of route objects, the attribute which is the class's primary key is where the route itself is defined.

It is therefore preferable to create a new route6 class than a multi-protocol class.

Each inter-AS ipv6 route originated by an AS is thus specified as:

```
route6:         [mandatory]  [single]     [primary/look-up key]
...  (rest an in the route class)

route6: 2001:610:240::/48
origin: AS3333
...
```

3.3 route-set

This class is used in <filter> expressions to specify a set of route prefixes.

A new attribute "mp-members:" is defined for this class with the following syntax:

```
mp-members: afi <afi-list> list of <address-prefix-range> or
            afi <afi-list> <route-set-name> or
            afi <afi-list> <route-set-name><range-operator>

route-set: rs-foo
mp-members: afi ipv6 rs-bar          # common members with afi constraint
mp-members: afi ipv6 rs-foo2, 3FFE:FFFF::/35 # v6 only members...
mp-members: afi ipv4 rs-foo3, 128.9.0.0/16
```

3.4 filter-set

The new "mp-filter:" attribute defines the set's policy filter. A policy filter is a logical expression which when applied to a set of routes returns a subset of these routes.

```
mp-filter:      afi <afi> <filter>
```

<filter> is defined in section Section 2.2.

The relevant parts of the new filter-set class are shown below:

```
filter-set: [mandatory] [ single] [class key]
mp-filter:  [optional] [multiple]
filter:     [optional] [multiple]
...
```

Note that according to this definition empty filters are possible and should be handled correctly.

3.5 peering-set

An "mp-peering:" attribute is introduced in this class.

```
mp-peering: afi <afi> <peering> Section 2.2
```

```
peering-set: [mandatory] [single] [class key]
peering:     [optional]  [multiple]
mp-peering:  [optional]  [multiple]
...

Example:

peering-set: prng-ebgp-peers
mp-peering:    afi ipv6 AS2 3FFE:FFFF::1 at 3FFE:FFFF::2
```

3.6 inet-rtr Class

This class gets two new attributes: "interface:" which allows the definition of generic interfaces, including the information previously contained in the "ifaddr:" attribute and new types such as tunnels.

mp-peer which includes and extends the functionality of the exisiting "peer:" attribute.

```
interface: afi <afi> <address> masklen <mask>
                     [ tunnel <remote-endpoint-address>,<encapsulation> ]
```

The new syntax allows native IPv4 and IPv6 interface definitions as well as the definition of tunnels as virtual interfaces.

Without the optional part, this attribute allows the same functionality as the "ifaddr:" attribute but extends it to allow IPv6 addresses.

In the case of the interface being a tunnel, the optional part describes the tunnel configuration as follows:

remote-endpoint-address indicates the IP address of the remote endpoint of the tunnel. The address family must match that of the local endpoint.

<encapsulation> denotes the encapsulation used in the tunnel and is one of {GRE,IPv6inIPv4,IPinIP,DVMRP}

Routing policies for these routers should be described in the appropriate classes (eg. peering and autnum).

```
   mp-peer: <protocol> afi <afi> <address> <options> |
            <protocol> <inet-rtr-name>     <options> |
            <protocol> <rtr-set-name>      <options> |
            <protocol> <peering-set-name>  <options>
```

3.7 rtr-set Class

```
   mp-members: list of <inet-rtr-name>          |
               <rtr-set-name>                        |
               afi <afi> list of <address-prefix>

   mp-members:     [optional]   [multiple]
```

4. Security Considerations

This document describes extensions to RPSL, a language for expressing routing policies. The extensions introduce ways of making the configurations currently available for describing IPv4 routing policies to IPv6. They introduce no additional security mechanisms or threats.

5. Acknowldegments

The authors wish to thank all the people who have contributed to this document through numerous discussions.

Particularly Ekaterina Petrusha for highly valuable discussions and suggestions. Shane Kerr, Engin Gunduz, Mark Blanchet and David Kessens participated constructively in many discussions. Finally Cengiz Alaettinoglu who is still the reference in all things RPSL.

References

[1] Alaettinoglu, C., Villamizar, C., Gerich, E., Kessens, D., Meyer, D., Bates, T., Karrenberg, D. and M. Terpstra, "Routing Policy Specification Language (RPSL)", RFC 2622, June 1999.

Authors' Addresses

Joao Damas
EMail: joao@psg.com

Florent Parent
Viagenie
EMail: Florent.Parent@viagenie.qc.ca

Andrei Robachevksi
RIPE NCC
EMail: andrei@ripe.net

Intellectual Property Statement

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.