Network Working Group
Request for Comments: 4012
Updates: 2725, 2622
Category: Standards Track

L. Blunk
Merit Network
J. Damas
Internet Systems Consortium
F. Parent
Hexago
A. Robachevsky
RIPE NCC

March 2005

Язык описания правил маршрутизации следующего поколения (RPSLng)

Routing Policy Specification Language next generation (RPSLng)

Статус документа

В этом документе содержится спецификация протокола, предложенного сообществу Internet. Документ служит приглашением к дискуссии в целях развития и совершенствования протокола. Текущее состояние стандартизации протокола вы можете узнать из документа "Internet Official Protocol Standards" (STD 1). Документ может распространяться без ограничений.

Аннотация

В этом документе вводится набор простых расширений языка RPSL[1], позволяющих документировать правила маршрутизации для IPv6 и семейств групповых адресов, используемых в настоящее время в Internet.

Оглавление

1. Введение

В RFC 2622 [1] определён язык RPSL для протоколов индивидуальной (unicast) маршрутизации IPv4 и дано руководство по расширению языка. Кроме того, в RFC 2725 [2] описаны расширения безопасности для RPSL.

В этом документе описаны расширения языка RPSL, преследующие несколько целей:

- обеспечить расширение RPSL для новых семейств адресов (в частности, для документирования маршрутизации IPv6 и multicast);
- обеспечить совместимость с прежними версиями и минимизировать влияние на существующие инструменты и процессы в соответствии с рекомендациями раздела 10 в RFC 2622 [1] для расширения RPSL;

[1]Routing Policy Specification Language - язык описания правил маршрутизации.

- обеспечить ясность и однозначность - информация RPSL используется как людьми, так и программами;
- минимизировать дублирование информации (в частности для случаев, когда правила маршрутизации в разных семействах адресов совпадают).

Добавление поддержки IPv6 и групповой адресации в RPSL ведёт к появлению четырёх разных политик маршрутизации, которые требуется различать в данном документе - IPv4 {unicast|multicast}, Ipv6 {unicast|multicast}).

2. Задание политики маршрутизации для разных семейств адресов

Политика маршрутизации в настоящее время специфицирована в классе aut-num с использованием атрибутов «import:», «export:» и «default:». Иногда важно различать правила для разных семейств адресов, а также правила для индивидуальной и групповой (multicast) маршрутизации.

Хотя синтаксис существующих атрибутов import, export и default можно расширить, это создаст проблемы совместимости с ранними версиями и сделает выражения менее понятными.

С учётом того, что атрибуты import:, export: и default: явно заданы для правил маршрутизации индивидуальных адресов IPv4 и определены в RPSL, здесь вводятся новые мультипротокольные атрибуты (с префиксом mp-), которые описаны ниже.

2.1. Устранение неоднозначностей

К одним и те же партнерским отношениям могут быть привязано более одного атрибута мультипротокольной политики или комбинация мультипротокольных атрибутов (при задании политики для индивидуальной адресации IPv4) и ранее определённых атрибутов политики для индивидуальной адресации IPv4. В таких случаях реализациям следует пользоваться правилом specification-order, определенным в параграфе 6.4 RFC 2622 [1]. Для устранения неоднозначности используется действие, соответствующее первой партнерской спецификации.

2.2. Атрибут словаря afi

В этом параграфе вводится новый атрибут словаря:

> Идентификатор семейства адресов <afi>1 представляет собой RPSL-список семейств адресов, для которого следует вычислять данное выражение политики маршрутизации. Атрибут <afi> является опциональным в составе мультипротокольных атрибутов, вводимых для класса aut-num. Определен псевдоидентификатор any для более компактного выражения политики.

Возможные значения <afi> перечислены ниже:

```
ipv4.unicast
ipv4.multicast
ipv4 (эквивалент ipv4.unicast, ipv4.multicast)
ipv6.unicast
ipv6.multicast
ipv6 (эквивалент ipv6.unicast, ipv6.multicast)
any (эквивалент to ipv4, ipv6)
any.unicast (эквивалент ipv4.unicast, ipv6.unicast)
any.multicast (эквивалент ipv4.multicast, ipv6.multicast)
```

При включении этих значений в атрибуты им должно предшествовать ключевое слово afi.

Список <afi-list> определяется, как одно или множество значений afi, разделённых запятыми.

2.3. Расширения словаря RPSL

Для поддержки адресов IPv6 в атрибуте next-hop rp-attribute, в RPSL добавлен новый тип предопределённого словаря ipv6_address. Определение этого типа дано в параграфе 2.2 RFC 3513 [3].

Атрибут next-hop rp-attribute расширен в словаре следующим образом:

```
rp-attribute: # следующий маршрутизатор в статическом маршруте
              next-hop
              operator=(union ipv4_address, ipv6_address, enum[self])
```

В спецификацию словаря <protocol> добавлено новое значение:

```
MPBGP
```

MPBGP трактуется, как BGP4 с мультипротокольными расширениями (его часто называют BGP4+). Обозначение BGP4+ не используется в имени словаря, поскольку спецификация RPSL не допускает использования символа + в именах протоколов.

2.4. Типы IPv6 RPSL

В этом документе упоминаются три новых типа IPv6 RPSL, а именно <ipv6-address>, <ipv6-address-prefix> и <ipv6-address-prefix-range>. Типы <ipv6-address> и <ipv6-address-prefix> определены в параграфах 2.2 и 2.3 RFC 3513 [3]. Тип <ipv6-address-prefix-range> добавляет оператор диапазона к типу <ipv6-address-prefix>. Оператор диапазона определен в разделе 2 RFC 2622 [1].

2.5. mp-import, mp-export, mp-default

В класс aut-num добавлены три новых атрибута:

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

Эти атрибуты включают спецификацию afi (семейство адресов). Отметим, что спецификация afi не является обязательной. Если afi не задано, предполагается, что правило применимо для всех семейств протоколов -

ipv4.unicast, ipv4.multicast, ipv6.unicast и ipv6.multicast. Это является эквивалентом спецификации семейства адресов afi any. Атрибуты mp-import и mp-export имеют базовую спецификацию правил, дополненную более мощной структурированной спецификацией.

Синтаксис для атрибута mp-default и базовой спецификации для атрибутов mp-import и mp-export показан ниже.

Атрибут	*Значение*	*Тип*
mp-import	[protocol <protocol-1>] [into <protocol-2>] [afi <afi-list>] from <mp-peering-1> [action <action-1>; ... <action-N>;] . . . from <mp-peering-M> [action <action-1>; ... <action-N>;] accept <mp-filter> [;]	Необязательный, многозначный
mp-export	[protocol <protocol-1>] [into <protocol-2>] [afi <afi-list>] to <mp-peering-1> [action <action-1>; ... <action-N>;] . . . to <mp-peering-M> [action <action-1>; ... <action-N>;] announce <mp-filter> [;]	Необязательный, многозначный
mp-default	[afi <afi-list>] to <mp-peering> [action <action-1>; ... <action-N>;] [networks <mp-filter>]	Необязательный, многозначный

Правила mp-import и mp-export могут быть структурированными. В соответствии с RFC 2622 [1] структурирование правил рекомендуется только опытным пользователям RPSL. Синтаксис структурированного правила mp-import определен ниже. Отметим, что точка с запятой (;) в конце <import-factor> является обязательным символом, хотя в бесструктурных записях правил этот символ не обязателен. Синтаксис структурированного правила mp-export аналогичен синтаксису атрибута mp-import. Структурированный синтаксис позволяет вносить в правила исключения и уточнения с помощью ключевых слов except (исключить) и refine (улучшить). Более того, исключения и уточнения могут задавать необязательный список afi для ограничения воздействия правила определенным семейством адресов.

Отметим, что определение разрешает последовательность или «каскадирование» уточнений и исключений. В RFC 2622 [1] это некорректно названо «вложенностью» выражений (nested expressions). Синтаксис не разрешает действительно вложенных выражений.

```
<import-factor> ::=
     from <mp-peering-1> [action <action-1>; ... <action-M>;]
     . . .
     from <mp-peering-N> [action <action-1>; ... <action-K>;]
     accept <mp-filter>;

<import-term> :: = import-factor |
     {
     <import-factor-1>
     . . .
     <import-factor-N>
     }

<import-expression> ::= <import-term> |
     <import-term> EXCEPT <afi-import-expression> |
     <import-term> REFINE <afi-import-expression>

<afi-import-expression> ::= [afi <afi-list>] <import-expression>

mp-import: [protocol <protocol-1>] [into <protocol-2>]
     <afi-import-expression>
```

2.5.1. <mp-peering>

<mp-peering> указывает AS (и маршрутизатор, если присутствует) и определяется следующим образом:

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

где <as-expression> - выражение из номеров и наборов AS с использованием операторов AND, OR и EXCEPT, а <mp-router-expression> - выражение из адресов маршрутизаторов ipv4-addresses или ipv6-addresses, имён inet-rtr и rtr-set с использованием операторов AND, OR, EXCEPT. Двоичный оператор EXCEPT является оператором исключения (вычитания) множества и имеет такой же приоритет исполнения, как оператор AND. Семантически оператор EXCEPT эквивалентен комбинации AND NOT, т. е. (AS65001 OR AS65002) EXCEPT AS65002 равно AS65001.

2.5.2. <mp-filter>

Выражение фильтра <mp-filter> произведено на основе выражения RPSL <filter>, определённого в параграфе 5.4 RFC 2622 [1]. Фильтр <mp-filter> расширяет выражение <filter>, позволяя указывать префиксы и диапазоны префиксов IPv6. В частности, выражение Address-Prefix Set в <mp-filter> может включать префиксы и диапазоны префиксов IPv4 и IPv6. В остальном <mp-filter> идентично выражению RPSL <filter>. Множества Address-Prefix указываются в фигурных скобках {}. Фильтру соответствует множество маршрутов, для которых адресные префиксы получателей входят в указанное множество. Например,

```
{ 192.0.2.0/24, 2001:0DB8::/32 }
{ 2001:0DB8:0100::/48^+, 2001:0DB8:0200::/48^64 }
```

2.5.3. Примеры правил

Семейство адресов может быть задано в последующем уточнении или исключении, которое применяется только в рамках содержащего его правила.

В примере

```
aut-num:    AS65534
mp-import: afi any.unicast from AS65001 accept as-foo;
             except afi any.unicast {
               from AS65002 accept AS65226;
             } except afi ipv6.unicast {
                 from AS65003 accept {2001:0DB8::/32};
               }
```

последний оператор except используется только для индивидуальных адресов IPv6, тогда как остальные выражения для импорта рассчитываются для индивидуальных адресов IPv6 и IPv4.

Проверка выражения правила выполняется путём проверки каждой из его компонент. Проверка peering-sets и filter-sets ограничивается семейством адресов. Такие ограничения могут приводить к NOT ANY <mp-filter> или недействительному <mp-peering> в зависимости от явного или неявного указания семейства адресов. Конфликты с явным или неявным указанием семейства разрешаются в процессе проверки выражения правила. Реализация проверки RPSL может выдавать предупреждения при обнаружении NOT ANY <mp-filter>. Приведённый ниже пример mp-import содержит фильтр <mp-filter>, который при проверке даёт NOT ANY.

```
aut-num: AS65002
mp-import: afi ipv6.unicast from AS65001 accept {192.0.2.0/24}
```

3. Класс route6

Класс route6 является эквивалентом класса route для IPv6. Как и для класса route, ключ класса route6 задаётся парой атрибутов route6 и origin. Кроме атрибута route6 класс route6 включает те же атрибуты, которые используются для класса route. Хотя имена атрибутов совпадают, атрибуты inject, components, exports-comps, holes и mnt-routes должны задавать адреса и префиксы IPv6, а не IPv4. Это требование выражается указанием <ipv6-router-expression>, <ipv6-filter> и <ipv6-address-prefix>, как показано ниже. Определение <ipv6-address-prefix> приведено выше. Фильтр <ipv6-filter> связан с <mp-filter>, как описано в параграфе 2.5.2, но может включать только тип <ipv6-address-prefix>. Аналогично, выражение <ipv6-router-expression> связано с <mp-router-expression>, как определено выше в параграфе 2.5.1, но может включать только тип <ipv6-address>.

Атрибут	*Значение*	*Тип*
route6	<ipv6-address-prefix>	Обязательный, ключ класса, однозначный
origin	<as-number>	Обязательный, ключ класса, однозначный
member-of	список <route-set-name>	Необязательный, многозначный
inject	[at <ipv6-router-expression>] ... [action <action>] [upon <condition>]	Необязательный, многозначный
components	[ATOMIC] [[<ipv6-filter>] [protocol <protocol> <ipv6-filter> ...]]	Необязательный, однозначный
aggr-bndry	<as-expression>	Необязательный, однозначный
aggr-mtd	inbound или outbound [<as-expression>]	Необязательный, однозначный
export-comps	<ipv6-filter>	Необязательный, однозначный
holes	список <ipv6-address-prefix>	Необязательный, многозначный
mnt-lower	список <mntner-name>	Необязательный, многозначный
mnt-routes	список <mntner-name> [{список <ipv6-address-prefix-range>} или ANY]	Необязательный, многозначный

Пример

```
route6:   2001:0DB8::/32
origin:   AS65001
```

4. Обновление имеющихся классов для поддержки расширений

4.1. Класс as-set

Класс as-set определяет множество автономных систем (AS), задаваемое путём прямого перечисления, с помощью ссылки на другое множество as-set или с помощью mbrs-by-ref. Важно отметить, что «В контексте, который предполагает множество маршрутов (например, атрибут members в классе route-set), [...] множество (as-set) AS-X определяет набор маршрутов, исходящих из автономных систем в составе AS-X.» (параграф 5.3 RFC 2622 [1]).

Класс, следовательно, as-set служит для задания множества маршрутных префиксов, которое может быть ограничено конкретным семейством адресов.

В существующий класс as-set не требуется вносить какие-либо изменения. Значения класса могут фильтроваться по семействам адресов с использованием обычных механизмов фильтрации с целью использования в современных системах маршрутных реестров IRR[1].

[1]Internet Routing Registry.

4.2. Класс route-set

Этот класс служит для задания множества маршрутных префиксов.

Для этого класса определен новый атрибут mp-members, позволяющий задавать диапазоны адресных префиксов IPv4 и IPv6.

Атрибут	*Значение*	*Тип*
mp-members	список (<ipv4-address-prefix-range> или <ipv6-address-prefix-range> или <route-set-name> или <route-set-name><range-operator>)	Необязательный, многозначный

Пример

```
route-set:  rs-foo
mp-members: rs-bar
mp-members: 2001:0DB8::/32  # префикс v6
mp-members: 192.0.2.0/24    # префикс v4
```

4.3. Класс filter-set

Новый атрибут mp-filter: определяет фильтр правил, представляющий собой логическое выражение, применяемое к множеству маршрутов и возвращающее некое подмножество этих маршрутов. Имеющие отношение к таким фильтрам компоненты обновлённого класса filter-set показаны ниже.

Атрибут	*Значение*	*Тип*
filter-set	<object-name>	Обязательный, ключ класса, однозначный
filter	<filter>	Необязательный, однозначный
mp-filter	<mp-filter>	Необязательный, однозначный

Определение <mp-filter> приведено выше в параграфе 2.5.2. Хотя атрибуты filter: и mp-filter: не являются обязательными (тип optional), выражение filter-set должно включать один из этих двух атрибутов. Реализациям следует отвергать экземпляры, содержащие в одном объекте оба атрибута, поскольку интерпретация таких filter-set становится неопределённой.

4.4. Класс peering-set

Для класса peering-set (множество партнёров) добавлен атрибут mp-peering:.

Атрибут	*Значение*	*Тип*
peering-set	<object-name>	Обязательный, ключ класса, однозначный
peering	<peering>	Необязательный, многозначный
mp-peering	<mp-peering>	Необязательный, многозначный

Пример

```
peering-set:   prng-ebgp-peers
mp-peering:    AS65002 2001:0DB8::1 at 2001:0DB8::2
```

Определение <mp-peering> приведено выше в параграфе 2.5.1. Хотя атрибуты peering: и mp-peering: не являются обязательными (тип optional), в peering-set должен присутствовать хотя бы один из этих атрибутов.

4.5. Класс inet-rtr

Для класса inet-rtr добавлены два новых атрибута - interface:, позволяющий определить интерфейсы, включая информацию, ранее содержавшуюся в атрибуте ifaddr:, а также поддержку определения туннелей и mp-peer:, который включает и расширяет функциональность существующего атрибута peer:. Синтаксис определения interface: приведён ниже.

Атрибут	*Значение*	*Тип*
interface	<ipv4-address> или <ipv6-address> masklen <mask> [action <action>] [tunnel <remote-endpoint-address>,<encapsulation>]	Необязательный, многозначный

Синтаксис позволяет определять естественные интерфейсы IPv4 и IPv6, а также туннели в качестве виртуальных интерфейсов. Без учёта поддержки определений туннельных интерфейсов функциональность этого атрибута совпадает с функциональностью атрибута ifaddr:, расширенной поддержкой адресов IPv6.

Синтаксис определения туннельных интерфейсов описан ниже.

<remote-endpoint-address> показывает адрес IPv4 или IPv6 на удалённой стороне туннеля. Семейство адресов должно соответствовать адресу локальной стороны туннеля. <encapsulation> указывает применяемый в туннеле метод инкапсуляции и может принимать одно из двух значений {GRE, IPinIP} (отметим, что версии протокола IP для внутреннего и внешнего заголовков можно определить по контексту интерфейса — например, инкапсуляция IPv6-in-IPv4 будет просто IPinIP). Правила маршрутизации для таких маршрутизаторов следует описывать в подходящих классах (например, aut-num).

Атрибут mp-peer: описан ниже. Он отличается от атрибута peer: лишь поддержкой адресов IPv6.

Атрибут	*Значение*	*Тип*

mp-peer	<protocol> <ipv4-address> <options> или <protocol> <ipv6-address> <options> или <protocol> <inet-rtr-name> <options> или <protocol> <rtr-set-name> <options> или <protocol> <peering-set-name> <options>	Необязательный, многозначный

<protocol> указывает имя протокола, <options> представляет собой список разделённых запятыми опций партнёрства для <protocol>, как указано в словаре RPSL.

4.6. Класс rtr-set

Класс rtr-set расширен новым атрибутом mp-members:, который добавляет в прототип members: поддержку адресов IPv6. Атрибут показан ниже.

Атрибут	*Значение*	*Тип*
mp-members	список(<inet-rtr-name> или <rtr-set-name> или <ipv4-address> или <ipv6-address>)	Необязательный, многозначный

5. Расширения RFC 2725

В RFC 2725 [2] предложена модель проверки полномочий (authorization) для контроля целостности правил в маршрутных реестрах. Для поддержки этой модели были определены два новых атрибута mnt-routes и mnt-lower.

В RPSLng эти атрибуты расширены для классов route6 и inet6num (см. ниже). Кроме того, синтаксис имеющегося атрибута mnt-routes был изменён для поддержки необязательных списков диапазонов адресных префиксов IPv6 в объектах классов inet6num, route6 и aut-num. Эти списки содержат разделённые запятыми диапазоны префиксов и весь список заключается в фигурные скобки. Для класса aut-num диапазоны префиксов IPv6 могут смешиваться с диапазонами префиксов IPv4. Вместо указания диапазона может также использоваться ключевое слово ANY (любые). Для объектов inet6num и route6 слово ANY указывает все более специфические префиксы по сравнению с префиксом в поле ключа класса. Для класса aut-num слово ANY указывает все префиксы. По умолчанию при отсутствии дополнительных элементов принимается значение ANY. Сокращённое определение класса aut-num с обновлённым синтаксисом для атрибута mnt-routes представлено ниже.

Атрибут	*Значение*	*Тип*
aut-num	<as-number>	Обязательный, однозначный, ключ класса
mnt-routes	список<mntner-name> [{список (<ipv6-address-prefix-range> или <ipv4-address-prefix-range>)} или ANY]	Необязательный, многозначный

Ниже приведён пример использования mnt-routes. Это пример предоставляет MAINT-65001 полномочия создания объектов route6 с исходной AS 65002 для адресных префиксов IPv6 из диапазона 2001:0DB8::/32^+ и объектов route с исходной AS 65002 для префиксов IPv4 из диапазона 192.0.2.0/24^+.

```
aut-num: AS65002
mnt-routes: MAINT-AS65001 {2001:0DB8::/32^+, 192.0.2.0/24^+}
```

Отметим, что включение диапазонов префиксов IPv6 в атрибут mnt-routes объектов aut-num может приводить к конфликтам с имеющимися реализациями RPSL, которые поддерживают только диапазоны префиксов IPv4. Однако с учётом малой распространённости таких необязательных списков префиксов было принято решение о целесообразности расширения имеющегося атрибута mnt-routes в классе aut-num, а не создания нового типа атрибута.

Атрибут	*Значение*	*Тип*
inet6num	<ipv6-address-prefix>	Обязательный, однозначный, ключ класса
netname	<netname>	Обязательный, однозначный
descr	<free-form>	Обязательный, многозначный
country	<country-code>	Обязательный, многозначный
admin-c	<nic-handle>	Обязательный, многозначный
tech-c	<nic-handle>	Обязательный, многозначный
remarks	<free-form>	Необязательный, многозначный
notify	<email-address>	Необязательный, многозначный
mnt-lower	список <mntner-name>	Необязательный, многозначный
mnt-routes	список <mntner-name> [{список <ipv6-address-prefix-range>} или ANY]	Необязательный, многозначный
mnt-by	список <mntner-name>	Обязательный, многозначный
changed	<email-address> <date>	Обязательный, многозначный
source	<registry-name>	Обязательный, однозначный

Значение <country-code> должно быть корректным двухсимвольным идентификатором страны ISO 3166. <netname> указывает символьное имя для заданного адресного блока IPv6. Ограничений на резервные префиксы не накладывается. Определения взяты из RIPE Database Reference Manual [4].

5.1. Модель проверки полномочий для объектов route6

Удаление и обновление объектов route6 не отличается от аналогичных операций, описанных в RFC 2725 [2]. Правила создания объектов route6 реплицированы из соответствующих правил для route с RFC 2725 [2] (параграф 9.9).

При добавлении объектов route6 должны быть выполнены два аутентификационных требования. Аутентификация должна выполняться по методу, указанному в объекте aut-num в соответствии со спецификацией объекта route6 или, при отсутствии подходящего объекта route6, в соответствии с объектом inet6num.

Добавляемый объект представляется с номером AS и префиксом IPv6 в качестве ключа. Если для добавляемого объекта route6 не существует объекта aut-num, добавление будет отвергнуто. Если объект aut-num имеется, представление проверяется на предмет применимости поддерживающих (maintainer). Выполняется поиск для префикса сначала на предмет точного соответствия, а при отсутствии совпадения ищется наиболее длинный префикс, соответствующий указанному. Если поиск дал результат, будет возвращён один или множество объектов route6. Подача должна соответствовать применимому поддерживающему по крайней мере для одного из возвращённых объектов route6. Если при поиске не было возвращено объектов route6, выполняется поиск объекта inet6num, точно соответствующего префиксу, или наиболее специфичного inet6num, который менее специфичен, нежели подаваемый объект route6.

После того, как найден объект aut-num и список объектов route6 или объект inet6num, для этих объектов должна быть выполнена проверка полномочий. Подходящим объектом maintainer будет любой из указанных в атрибутах mnt-routes. Если в объекте присутствует один или несколько атрибутов mnt-routes, атрибуты mnt-by и mnt-lower не принимаются во внимание. При отсутствии а данном объекте атрибутов mnt-routes используется первый из атрибутов mnt-lower (только в случаях, когда данный объект является inet6num и менее специфичен, чем добавляемый объект route6). Если подходящего атрибута mnt-lower не найдено, используется атрибут mnt-by для этого объекта. Аутентификация должна соответствовать одной из проверок полномочий в каждом из двух объектов.

6. Вопросы безопасности

Этот документ описывает расширения RFC 2622 [1] и RFC 2725 [2], предназначенные для снятия ограничений, связанных с IPv6 и групповой адресацией. Расширения не создают новых угроз и связанной с безопасностью функциональности.

Хотя предложенные расширения не создают новых угроз, следует отметить, что исходный стандарт RPSL RFC 2622 [1] включал несколько слабых и/или уязвимых механизмов. Во-первых, это схема MAIL-FROM, которую легко обмануть путём подмены адреса отправителя в сообщении электронной почты, во-вторых, схема CRYPT-PW, которая может быть атакована с использованием словарей или перехвата паролей, если объекты RPSL отправляются через незашифрованный канал типа электронной почты, а в-третьих, механизм NONE, который не обеспечивает защиты объектов.

7. Благодарности

Авторы благодарят всех участников связанных с этим документом обсуждения и, в частности, Ekaterina Petrusha за ценные замечания и предложения. Shane Kerr, Engin Gunduz, Marc Blanchet и David Kessens внесли множество конструктивных предложений, а с Cengiz Alaettinoglu связано все, что относится к RPSL.

8. Литература

8.1. Нормативные документы

[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.

[2] Villamizar, C., Alaettinoglu, C., Meyer, D., and S. Murphy, "Routing Policy System Security", RFC 2725, December 1999.

[3] Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, April 2003.

8.2. Дополнительная литература

[4] Damas, J. and A. Robachevsky, "RIPE Database Reference Manual", August 2002.

Адреса авторов

Larry Blunk
Merit Network
EMail: ljb@merit.edu

Joao Damas
Internet Systems Consortium
EMail: Joao_Damas@isc.org

Florent Parent
Hexago
EMail: Florent.Parent@hexago.com

Andrei Robachevsky
RIPE NCC
EMail: andrei@ripe.net

Перевод на русский язык
Николай Малых
nmalykh@protokols.ru

Полное заявление авторских прав

Подтверждение

Финансирование функций RFC Editor обеспечено Internet Society.