Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 4, 2016

S. Previdi, Ed.
C. Filsfils
Cisco Systems, Inc.
B. Field
Comcast
I. Leung
Rogers Communications
J. Linkova
Google
E. Aries
Facebook
T. Kosugi
NTT
E. Vyncke
Cisco Systems, Inc.
D. Lebrun
Universite Catholique de Louvain
October 2, 2015

IPv6 Segment Routing Header (SRH)
draft-previdi-6man-segment-routing-header-08

Abstract

Segment Routing (SR) allows a node to steer a packet through a controlled set of instructions, called segments, by prepending a SR header to the packet. A segment can represent any instruction, topological or service-based. SR allows to enforce a flow through any path (topological, or application/service based) while maintaining per-flow state only at the ingress node to the SR domain.

Segment Routing can be applied to the IPv6 data plane with the addition of a new type of Routing Extension Header. This draft describes the Segment Routing Extension Header Type and how it is used by SR capable nodes.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

Table of Contents

1. Segment Routing Documents

Segment Routing terminology is defined in [I-D.ietf-spring-segment-routing].

Segment Routing use cases are described in [I-D.ietf-spring-problem-statement] and [I-D.ietf-spring-ipv6-use-cases].

Segment Routing protocol extensions are defined in [I-D.ietf-isis-segment-routing-extensions], and [I-D.ietf-ospf-ospfv3-segment-routing-extensions].

2. Introduction

Segment Routing (SR), defined in [I-D.ietf-spring-segment-routing], allows a node to steer a packet through a controlled set of instructions, called segments, by prepending a SR header to the packet. A segment can represent any instruction, topological or service-based. SR allows to enforce a flow through any path (topological or service/application based) while maintaining per-flow state only at the ingress node to the SR domain. Segments can be derived from different components: IGP, BGP, Services, Contexts, Locators, etc. The list of segment forming the path is called the Segment List and is encoded in the packet header.

SR allows the use of strict and loose source based routing paradigms without requiring any additional signaling protocols in the infrastructure hence delivering an excellent scalability property.

The source based routing model described in [I-D.ietf-spring-segment-routing] is inherited from the ones proposed by [RFC1940] and [RFC2460]. The source based routing model offers the support for explicit routing capability.

2.1. Data Planes supporting Segment Routing

Segment Routing (SR), can be instantiated over MPLS ([I-D.ietf-spring-segment-routing-mpls]) and IPv6. This document defines its instantiation over the IPv6 data-plane based on the use-cases defined in [I-D.ietf-spring-ipv6-use-cases].

This document defines a new type of Routing Header (originally defined in [RFC2460]) called the Segment Routing Header (SRH) in order to convey the Segment List in the packet header as defined in [I-D.ietf-spring-segment-routing]. Mechanisms through which segment are known and advertised are outside the scope of this document.

A segment is materialized by an IPv6 address. A segment identifies a topological instruction or a service instruction. A segment can be either:

- global: a global segment represents an instruction supported by all nodes in the SR domain and it is instantiated through an IPv6 address globally known in the SR domain.

- local: a local segment represents an instruction supported only by the node who originates it and it is instantiated through an IPv6 address that is known only by the local node.

2.2. Segment Routing (SR) Domain

We define the concept of the Segment Routing Domain (SR Domain) as the set of nodes participating into the source based routing model. These nodes may be connected to the same physical infrastructure (e.g.: a Service Provider’s network) as well as nodes remotely connected to each other (e.g.: an enterprise VPN or an overlay).

A non-exhaustive list of examples of SR Domains is:

- o The network of an operator, service provider, content provider, enterprise including nodes, links and Autonomous Systems.

- o A set of nodes connected as an overlay over one or more transit providers. The overlay nodes exchange SR-enabled traffic with segments belonging solely to the overlay routers (the SR domain). None of the segments in the SR-enabled packets exchanged by the overlay belong to the transit networks

The source based routing model through its instantiation of the Segment Routing Header (SRH) defined in this document equally applies to all the above examples.

While the source routing model defined in [RFC2460] doesn’t mandate which node is allowed to insert (or modify) the SRH, it is assumed in this document that the SRH is inserted in the packet by its source. For example:

- o At the node originating the packet (host, server).

- o At the ingress node of a SR domain where the ingress node receives an IPv6 packet and encapsulates it into an outer IPv6 header followed by a Segment Routing header.

2.2.1. SR Domain in a Service Provider Network

The following figure illustrates an SR domain consisting of an operator’s network infrastructure.

```
     (-------------------------- Operator 1 -----------------------)
     (                                                             )
     (  (-----AS 1-----)  (-------AS 2-------)  (----AS 3-------)  )
     (  (              )  (                  )  (               )  )
 A1--(--(--11---13--14-)--(-21---22---23--24-)--(-31---32---34--)--)--Z1
     (  ( /|\  /|\  /| )  ( |\  /|\  /|\  /| )  ( |\  /|\  /| \ )  )
 A2--(--(/ | \/ | \/ | )  ( | \/ | \/ | \/ | )  ( | \/ | \/ |  \)--)--Z2
     (  (  | /\ | /\ | )  ( | /\ | /\ | /\ | )  ( | /\ | /\ |   )  )
     (  (  |/  \|/  \| )  ( |/  \|/  \|/  \| )  ( |/  \|/  \|   )  )
 A3--(--(--15---17--18-)--(-25---26---27--28-)--(-35---36---38--)--)--Z3
     (  (              )  (                  )  (               )  )
     (  (--------------)  (------------------)  (---------------)  )
     (                                                             )
     (-------------------------------------------------------------)
```

Figure 1: Service Provider SR Domain

Figure 1 describes an operator network including several ASes and delivering connectivity between endpoints. In this scenario, Segment

Routing is used within the operator networks and across the ASes boundaries (all being under the control of the same operator). In this case segment routing can be used in order to address use cases such as end-to-end traffic engineering, fast re-route, egress peer engineering, data-center traffic engineering as described in [I-D.ietf-spring-problem-statement], [I-D.ietf-spring-ipv6-use-cases] and [I-D.ietf-spring-resiliency-use-cases].

Typically, an IPv6 packet received at ingress (i.e.: from outside the SR domain), is classified according to network operator policies and such classification results into an outer header with an SRH applied to the incoming packet. The SRH contains the list of segment representing the path the packet must take inside the SR domain. Thus, the SA of the packet is the ingress node, the DA (due to SRH procedures described in Section 4) is set as the first segment of the path and the last segment of the path is the egress node of the SR domain.

The path may include intra-AS as well as inter-AS segments. It has to be noted that all nodes within the SR domain are under control of the same administration. When the packet reaches the egress point of the SR domain, the outer header and its SRH are removed so that the destination of the packet is unaware of the SR domain the packet has traversed.

The outer header with the SRH is no different from any other tunneling encapsulation mechanism and allows a network operator to implement traffic engineering mechanisms so to efficiently steer traffic across his infrastructure.

2.2.2. SR Domain in a Overlay Network

The following figure illustrates an SR domain consisting of an overlay network over multiple operator’s networks.

```
       (--Operator 1---)  (-----Operator 2-----)  (--Operator 3---)
       (               )  (                     )  (               )
   A1--(--11---13--14--)--(--21---22---23--24--)--(-31---32---34--)--C1
       ( /|\  /|\  /|  )  (  |\  /|\  /|\  /|  )  ( |\  /|\  /| \ )
   A2--(/ | \/ | \/ |  )  (  | \/ | \/ | \/ |  )  ( | \/ | \/ |  \)--C2
       (  | /\ | /\ |  )  (  | /\ | /\ | /\ |  )  ( | /\ | /\ |   )
       (  |/  \|/  \|  )  (  |/  \|/  \|/  \|  )  ( |/  \|/  \|   )
   A3--(--15---17--18--)--(--25---26---27--28--)--(-35---36---38--)--C3
       (               )  (  |    |         |  )  (               )
       (---------------)  (--|----|---------|--)  (---------------)
                             |    |         |
                             B1   B2        B3
```

Figure 2: Overlay SR Domain

Figure 2 describes an overlay consisting of nodes connected to three different network operators and forming a single overlay network where Segment routing packets are exchanged.

The overlay consists of nodes A1, A2, A3, B1, B2, B3, C1, C2 and C3. These nodes are connected to their respective network operator and form an overlay network.

Each node may originate packets with an SRH which contains, in the segment list of the SRH or in the DA, segments identifying other overlay nodes. This implies that packets with an SRH may traverse operator's networks but, obviously, these SRHs cannot contain an address/segment of the transit operators 1, 2 and 3. The SRH originated by the overlay can only contain address/segment under the administration of the overlay (e.g. address/segments supported by A1, A2, A3, B1, B2, B3, C1,C2 or C3).

In this model, the operator network nodes are transit nodes and, according to [RFC2460], MUST NOT inspect the routing extension header since there are not the DA of the packet.

It is a common practice in operators networks to filter out, at ingress, any packet whose DA is the address of an internal node and it is also possible that an operator would filter out any packet destined to an internal address and having an extension header in it.

This common practice does not impact the SR-enabled traffic between the overlay nodes as the intermediate transit networks do never see a destination address belonging to their infrastructure. These SR-enabled overlay packets will thus never be filtered by the transit operators.

In all cases, transit packets (i.e.: packets whose DA is outside the domain of the operator's network) will be forwarded accordingly without introducing any security concern in the operator's network. This is similar to tunneled packets.

2.3. Illustration

In the context of Figure 3 we illustrate an example of how segment routing an be used within a SR domain in order to engineer traffic. Let's assume that the SR domain is configured as a single AS and the IGP (OSPF or IS-IS) is configured using the same cost on every link. Let's also assume that a packet P enters the SR domain at an ingress edge router I and that the operator requests the following requirements for packet P:

- o The local service S offered by node B must be applied to packet P.
- o The links AB and CE cannot be used to transport the packet P.
- o Any node N along the journey of the packet should be able to determine where the packet P entered the SR domain and where it will exit. The intermediate node should be able to determine the paths from the ingress edge router to itself, and from itself to the egress edge router.
- o Per-flow State for packet P should only be created at the ingress edge router.
- o The operator can forbid, for security reasons, anyone outside the operator domain to exploit its intra-domain SR capabilities.

```
                 S
         I---A---B---C---E
              \  |  / \ /
               \ | /   F
                \|/
                 D
```

Figure 3: An illustration of SR properties

All these properties may be realized by instructing the ingress SR edge router I to create a SRH with the list of segments the packet must traverse: D, B, S, F, E. Therefore, the ingress router I creates an outer header where:

- o the SA is the IPv6 address of I

- o the final destination of the packet is the SR egress node E however, D being the first segment of the path, the DA is set to D IPv6 address.

- o the SRH is inserted with the segment list consisting of following IPv6 addresses: D, B, S, F, E

The SRH contains a source route encoded as a list of segments (D, B, S, F, E). The ingress and egress nodes are identified in the packet respectively by the SA and the last segment of the segment list.

The packet P reaches the ingress SR node I. Node I pushes the newly created outer header and SRH with the Segment List as illustrated above (D, B, S, F, E)

D is the IPv6 address of node D and it is recognized by all nodes in the SR domain as the forwarding instruction "forward to D according to D route in the IPv6 routing table". The routing table being built through IGPs (OSPF or IS-IS) it is equivalent to say "forward according to shortest path to D".

Once at D, the next segment is inspected and executed (segment B).

B is an instruction recognized by all the nodes in the SR domain which causes the packet to be forwarded along the shortest path to B.

Once at B, the next segment is executed (segment S).

S is an instruction only recognized by node B which causes the packet to receive service S.

Once the service S is applied, the next segment is executed (segment F) which causes the packet to be forwarded along the shortest path to F.

Once at F, the next segment is executed (segment E).

E is an instruction recognized by all the nodes in the SR domain which causes the packet to be forwarded along the shortest path to E.

E being the destination of the packet, removes the outer header and the SRH. Then, it inspects the inner packet header and forwards the packet accordingly.

All of the requirements are met:

- o First, the packet P has not used links AB and CE: the shortest-path from I to D is I-A-D, the shortest-path from D to B is D-B, the shortest-path from B to F is B-C-F and the shortest-path from F to E is F-E, hence the packet path through the SR domain is I-A-D-B-C-F-E and the links AB and CE have been avoided.

- o Second, the service S supported by B has been applied on packet P.

- o Third, any node along the packet path is able to identify the service and topological journey of the packet within the SR domain by inspecting the SRH and SA/DA fields of the packet header.

- o Fourth, only node I maintains per-flow state for packet P. The entire program of topological and service instructions to be executed by the SR domain on packet P is encoded by the ingress edge router I in the SR header in the form of a list of segments where each segment identifies a specific instruction. No further per-flow state is required along the packet path. Intermediate nodes only hold states related to the global node segments and their local segments. These segments are not per-flow specific and hence scale very well. Typically, an intermediate node would maintain in the order of 100's to 1000's global node segments and in the order of 10's to 100 of local segments.

- o Fifth, the SR header (and its outer header) is inserted at the entrance to the domain and removed at the exit of the operator domain. For security reasons, the operator can forbid anyone outside its domain to use its intra-domain SR capability (e.g. configuring ACL that deny any packet with a DA towards its infrastructure segment).

3. IPv6 Instantiation of Segment Routing

3.1. Segment Identifiers (SIDs)

Segment Routing, as described in [I-D.ietf-spring-segment-routing], defines Node-SID and Adjacency-SID. When SR is used over IPv6 data-plane the following applies.

3.1.1. Node-SID

The Node-SID identifies a node. With SR-IPv6 the Node-SID is an IPv6 address that the operator configured on the node and that is used as the node identifier. Typically, in case of a router, this is the IPv6 address of the node loopback interface. Therefore, SR-IPv6 does not require any additional SID advertisement for the Node Segment. The Node-SID is in fact the IPv6 address of the node.

3.1.2. Adjacency-SID

Adjacency-SIDs can be either globally scoped IPv6 addresses or IPv6 addresses known locally by the node but not advertised in any control plane (in other words an Adjacency-SID may well be any 128-bit identifier). Obviously, in the latter case, the scope of the Adjacency-SID is local to the router and any packet with the a such Adjacency-SID would need first to reach the node through the node's Segment Identifier (i.e.: Node-SID) prior for the node to process the Adjacency-SID. In other words, two segments (SIDs) would then be required: the first is the node's Node-SID that brings the packet to the node and the second is the Adjacency-SID that will make the node to forward the packet through the interface the Adjacency-SID is allocated to.

In the SR architecture defined in [I-D.ietf-spring-segment-routing] a node may advertise one (or more) Adj-SIDs allocated to the same interface as well as a node can advertise the same Adj-SID for multiple interfaces. Use cases of Adj-SID advertisements are described in [I-D.ietf-spring-segment-routing]The semantic of the Adj-SID is:

> Send out the packet to the interface this Adj-SID is allocated to.

Advertisement of Adj-SID may be done using multiple mechanisms among which the ones described in ISIS and OSPF protocol extensions: [I-D.ietf-isis-segment-routing-extensions] and [I-D.ietf-ospf-ospfv3-segment-routing-extensions]. The distinction between local and global significance of the Adj-SID is given in the encoding of the Adj-SID advertisement.

3.2. Segment Routing Extension Header (SRH)

A new type of the Routing Header (originally defined in [RFC2460]) is defined: the Segment Routing Header (SRH) which has a new Routing Type, (suggested value 4) to be assigned by IANA.

The Segment Routing Header (SRH) is defined as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header   |  Hdr Ext Len  | Routing Type  | Segments Left |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| First Segment |            Flags              |  HMAC Key ID  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
```

```
    |             Segment List[0] (128 bits ipv6 address)           |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                                                               |
                                  ...
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |             Segment List[n] (128 bits ipv6 address)           |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |             Policy List[0] (optional)                         |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |             Policy List[1] (optional)                         |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |             Policy List[2] (optional)                         |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |             Policy List[3] (optional)                         |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                                                               |
    |                                                               |
    |                        HMAC (256 bits)                        |
    |                         (optional)                            |
    |                                                               |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- o Next Header: 8-bit selector. Identifies the type of header immediately following the SRH.

- o Hdr Ext Len: 8-bit unsigned integer, is the length of the SRH header in 8-octet units, not including the first 8 octets.

- o Routing Type: TBD, to be assigned by IANA (suggested value: 4).

- o Segments Left. Defined in [RFC2460], it contains the index, in the Segment List, of the next segment to inspect. Segments Left is decremented at each segment.

- o First Segment: contains the index, in the Segment List, of the first segment of the path which is in fact the last element of the Segment List.

- o Flags: 16 bits of flags. Following flags are defined:

```
                     1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|C|P|R|R|    Policy Flags       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

C-flag: Clean-up flag. Set when the SRH has to be removed from the packet when packet reaches the last segment.

P-flag: Protected flag. Set when the packet has been rerouted through FRR mechanism by a SR endpoint node.

R-flags. Reserved and for future use.

Policy Flags. Define the type of the IPv6 addresses encoded into the Policy List (see below). The following have been defined:

Bits 4-6: determine the type of the first element after the segment list.

Bits 7-9: determine the type of the second element.

Bits 10-12: determine the type of the third element.

Bits 13-15: determine the type of the fourth element.

The following values are used for the type:

0x0: Not present. If value is set to 0x0, it means the element represented by these bits is not present.

0x1: SR Ingress.

0x2: SR Egress.

0x3: Original Source Address.

0x4 to 0x7: currently unused and SHOULD be ignored on reception.

- HMAC Key ID and HMAC field, and their use are defined in Section 5.

- Segment List[n]: 128 bit IPv6 addresses representing the nth segment in the Segment List. The Segment List is encoded starting from the last segment of the path. I.e., the first element of the segment list (Segment List [0]) contains the last segment of the path while the last segment of the Segment List (Segment List[n]) contains the first segment of the path. The index contained in "Segments Left" identifies the current active segment.

- Policy List. Optional addresses representing specific nodes in the SR path such as:

  SR Ingress: a 128 bit generic identifier representing the ingress in the SR domain (i.e.: it needs not to be a valid IPv6 address).

  SR Egress: a 128 bit generic identifier representing the egress in the SR domain (i.e.: it needs not to be a valid IPv6 address).

  Original Source Address: IPv6 address originally present in the SA field of the packet.

  The segments in the Policy List are encoded after the segment list and they are optional. If none are in the SRH, all bits of the Policy List Flags MUST be set to 0x0.

3.2.1. SRH and RFC2460 behavior

The SRH being a new type of the Routing Header, it also has the same properties:

SHOULD only appear once in the packet.

Only the router whose address is in the DA field of the packet header MUST inspect the SRH.

Therefore, Segment Routing in IPv6 networks implies that the segment identifier (i.e.: the IPv6 address of the segment) is moved into the DA of the packet.

The DA of the packet changes at each segment termination/completion and therefore the original DA of the packet MUST be encoded as the last segment of the path.

As illustrated in Section 2.3, nodes that are within the path of a segment will forward packets based on the DA of the packet without inspecting the SRH. This ensures full interoperability between SR-capable and non-SR-capable nodes.

4. SRH Procedures

In this section we describe the different procedures on the SRH.

4.1. Segment Routing Node Functions

SR packets are forwarded to segments endpoints (i.e.: the segment endpoint is the node representing the segment and whose address is in the segment list and in the DA of the packet when traveling in the segment). The segment endpoint, when receiving a SR packet destined to itself, does:

- o Inspect the SRH.
- o Determine the next active segment.
- o Update the Segments Left field (or, if requested, remove the SRH from the packet).
- o Update the DA.
- o Forward the packet to the next segment.

The procedures applied to the SRH are related to the node function. Following nodes functions are defined:

Source SR Node.

SR Domain Ingress Node.

Transit Node.

SR Endpoint Node.

4.1.1. Source SR Node

A Source SR Node can be any node originating an IPv6 packet with its IPv6 and Segment Routing Headers. This include either:

A host originating an IPv6 packet

A SR domain ingress router encapsulating a received IPv6 packet into an outer IPv6 header followed by a SRH

The mechanism through which a Segment List is derived is outside of the scope of this document. As an example, the Segment List may be obtained through:

Local path computation.

Local configuration.

Interaction with a centralized controller delivering the path.

Any other mechanism.

The following are the steps of the creation of the SRH:

Next Header and Hdr Ext Len fields are set according to [RFC2460].

Routing Type field is set as TBD (SRH).

The Segment List is built with the FIRST segment of the path encoded in the LAST element of the Segment List. Subsequent segments are encoded on top of the first segment. Finally, the LAST segment of the path is encoded in the FIRST element of the Segment List. In other words, the Segment List is encoded in the reverse order of the path.

The original DA of the packet is encoded as the last segment of the path (encoded in the first element of the Segment List).

The DA of the packet is set with the value of the first segment (found in the last element of the segment list).

The Segments Left field is set to n-1 where n is the number of elements in the Segment List.

The First Segment field is set to n-1 where n is the number of elements in the Segment List.

The packet is sent out towards the first segment (i.e.: represented in the packet DA).

HMAC and HMAC Key ID may be set according to Section 5.

4.1.2. SR Domain Ingress Node

The SR Domain Ingress Node is the node where ingress policies are applied and where the packet path (and processing) is determined.

After policies are applied and packet classification is done, the result may be instantiated into a Segment List representing the path the packet should take. In such case, the SR Domain Ingress Node instantiate a new outer IPv6 header to which the SRH is appended (with the computed Segment List). The procedures for the creation and insertion of the new SRH are described in Section 4.1.1.

4.1.3. Transit Node

According to [RFC2460], the only node who is allowed to inspect the Routing Extension Header (and therefore the SRH), is the node corresponding to the DA of the packet. Any other transit node MUST NOT inspect the underneath routing header and MUST forward the packet towards the DA and according to the IPv6 routing table.

In the example case described in Section 2.2.2, when SR capable nodes are connected through an overlay spanning multiple third-party infrastructure, it is safe to send SRH packets (i.e.: packet having a Segment Routing Header) between each other overlay/SR-capable nodes as long as the segment list does not include any of the transit provider nodes. In addition, as a generic security measure, any service provider will block any packet destined to one of its internal routers, especially if these packets have an extended header in it.

4.1.4. SR Segment Endpoint Node

The SR segment endpoint node is the node whose address is in the DA. The segment endpoint node inspects the SRH and does:

```
1.   IF DA = myself (segment endpoint)
2.      IF Segments Left > 0 THEN
           decrement Segments Left
           update DA with Segment List[Segments Left]
3.         IF Segments Left == 0 THEN
              IF Clean-up bit is set THEN remove the SRH
4.      ELSE give the packet to next PID (application)
             End of processing.
5.   Forward the packet out
```

5. Security Considerations

This section analyzes the security threat model, the security issues and mitigation techniques of SRH.

SRH is simply another type of the routing header as described in RFC 2460 [RFC2460] and is:

- o added to a new outer IP header by the ingress router when entering the SR domain or by the originating node itself. The source host can be outside the SR domain;

- o inspected and acted upon when reaching the destination address of the IP header per RFC 2460 [RFC2460].

Per RFC2460 [RFC2460], routers on the path that simply forward an IPv6 packet (i.e. the IPv6 destination address is none of theirs) will never inspect and process the content of any routing header (including SRH). Routers whose one interface IPv6 address equals the destination address field of the IPv6 packet MUST to parse the SRH and, if supported and if the local configuration allows it, MUST act accordingly to the SRH content.

According to RFC2460 [RFC2460], non SR-capable (or non SR-configured) router upon receipt of an IPv6 packet with SRH destined to an address of its:

- o must ignore the SRH completely if the Segment Left field is 0 and proceed to process the next header in the IPv6 packet;

- o must discard the IPv6 packet if Segment Left field is greater than 0 and send a Parameter Problem ICMP message back to the Source Address.

5.1. Threat model

5.1.1. Source routing threats

Using a SRH is a specific case of loose source routing, therefore it has some well-known security issues as described in RFC4942 [RFC4942] section 2.1.1 and RFC5095 [RFC5095]:

- o amplification attacks: where a packet could be forged in such a way to cause looping among a set of SR-enabled routers causing unnecessary traffic, hence a Denial of Service (DoS) against bandwidth;

- o reflection attack: where a hacker could force an intermediate node to appear as the immediate attacker, hence hiding the real attacker from naive forensic;

- o bypass attack: where an intermediate node could be used as a stepping stone (for example in a De-Militarized Zone) to attack another host (for example in the datacenter or any back-end server).

5.1.2. Applicability of RFC 5095 to SRH

First of all, the reader must remember this specific part of section 1 of RFC5095 [RFC5095], "A side effect is that this also eliminates benign RH0 use-cases; however, such applications may be facilitated by future Routing Header specifications.". In short, it is not forbidden to create new secure type of Routing Header; for example, RFC 6554 (RPL) [RFC6554] also creates a new Routing Header type for a specific application confined in a single network.

The main use case for SR consists of the single administrative domain (or cooperating administrative domains) where only trusted nodes with SR enabled and explicitely configured participate in SR: this is the same model as in RFC6554 [RFC6554]. All non-trusted nodes do not participate as either SR processing is not enabled by default or because they only process SRH from nodes within their domain.

Moreover, all SR routers SHOULD ignore SRH created by outsiders based on topology information (received on a peering or internal interface) or on presence and validity of the HMAC field. Therefore, if intermediate SR routers ONLY act on valid and authorized SRH (such as within a single administrative domain), then there is no security threat similar to RH-0. Hence, the RFC 5095 [RFC5095] attacks are not applicable.

5.1.3. Service stealing threat

Segment routing is used for added value services, there is also a need to prevent non-participating nodes to use those services; this is called 'service stealing prevention'.

5.1.4. Topology disclosure

The SRH may also contains IPv6 addresses of some intermediate SR routers in the path towards the destination, this obviously reveals those addresses to the potentially hostile attackers if those attackers are able to intercept packets containing SRH. On the other hand, if the attacker can do a traceroute whose probes will be forwarded along the SR path, then there is little learned by intercepting the SRH itself. The clean-bit of SRH can help by removing the SRH before forwarding the packet to potentially a non-trusted part of the network; if the attacker can force the generation of an ICMP message during the transit in the SR domain, then the ICMP will probably contain the SRH header (totally or partially) depending on the ICMP-generating router behavior.

5.1.5. ICMP Generation

Per section 4.4 of RFC2460 [RFC2460], when destination nodes (i.e. where the destination address is one of theirs) receive a Routing Header with unsupported Routing Type, the required behavior is:

- o If Segments Left is zero, the node must ignore the Routing header and proceed to process the next header in the packet.

- o If Segments Left is non-zero, the node must discard the packet and SHOULD send an ICMP Parameter Problem, Code 0, message to the packet's Source Address, pointing to the unrecognized Routing Type.

This required behavior could be used by an attacker to force the generation of ICMP message by any node. The attacker could send packets with SRH (with Segment Left different than 0) destined to a node not supporting SRH. Per RFC2460 [RFC2460], the destination node must then generate an ICMP message per RFC 2460, causing a local CPU utilization and if the source of the offending packet with SRH was spoofed could lead to a reflection attack without any amplification.

It must be noted that this is a required behavior for any unsupported Routing Type and not limited to SRH packets. So, it is not specific to SRH and the usual rate limiting for ICMP generation is required anyway for any IPv6 implementation and has been implemented and deployed for many years.

5.2. Security fields in SRH

This section summarizes the use of specific fields in the SRH. They are based on a key-hashed message authentication code (HMAC).

The security-related fields in SRH are:

- HMAC Key-id, 8 bits wide;
- HMAC, 256 bits wide (optional, exists only if HMAC Key-id is not 0).

The HMAC field is the output of the HMAC computation (per RFC 2104 [RFC2104]) using a pre-shared key and hashing algorithm identified by HMAC Key-id and of the text which consists of the concatenation of:

- the source IPv6 address;
- First Segment field;
- an octet whose bit-0 is the clean-up bit flag and others are 0;
- HMAC Key-id;
- all addresses in the Segment List.

The purpose of the HMAC field is to verify the validity, the integrity and the authorization of the SRH itself. If an outsider of the SR domain does not have access to a current pre-shared secret, then it cannot compute the right HMAC field and the first SR router on the path processing the SRH and configured to check the validity of the HMAC will simply reject the packet.

The HMAC field is located at the end of the SRH simply because only the router on the ingress of the SR domain needs to process it, then all other SR nodes can ignore it (based on local policy) because they trust the upstream router. This is to speed up forwarding operations because SR routers which do not validate the SRH do not need to parse the SRH until the end.

The HMAC Key-id field allows for the simultaneous existence of several hash algorithms (SHA-256, SHA3-256 ... or future ones) as well as pre-shared keys. This allows for pre-shared key roll-over when two pre-shared keys are supported for a while when all SR nodes converged to a fresher pre-shared key. The HMAC Key-id field is opaque, i.e., it has neither syntax not semantic except as an index to the right combination of pre-shared key and hash algorithm and except that a value of 0 means that there is no HMAC field. It could also allow for interoperation among different SR domains if allowed by local policy and assuming a collision-free Key Id allocation which is out of scope of this memo.

When a specific SRH is linked to a time-related service (such as turbo-QoS for a 1-hour period), then it is important to refresh the shared-secret frequently as the HMAC validity period expires only when the HMAC Key-id and its associated shared-secret expires.

5.2.1. Selecting a hash algorithm

The HMAC field in the SRH is 256 bits wide. Therefore, the HMAC MUST be based on a hash function whose output is at least 256 bits. If the output of the hash function is 256, then this output is simply inserted in the HMAC field. If the output of the hash function is larger than 256 bits, then the output value is truncated to 256 by taking the least-significant 256 bits and inserting them in the HMAC field.

SRH implementations can support multiple hash functions but MUST implement SHA-2 [FIPS180-4] in its SHA-256 variant.

NOTE: SHA-1 is currently used by some early implementations used for quick interoperations testing, the 160-bit hash value must then be right-hand padded with 96 bits set to 0. The authors understand that this is not secure but is ok for limited tests.

5.2.2. Performance impact of HMAC

While adding a HMAC to each and every SR packet increases the security, it has a performance impact. Nevertheless, it must be noted that:

- the HMAC field SHOULD be used only when SRH is inserted by a device (such as a home set-up box) which is outside of the segment routing domain. If the SRH is added by a router in the trusted segment routing domain, then, there is no need for a HMAC field, hence no performance impact.

- when present, the HMAC field MUST be checked and validated only by the first router of the segment routing domain, this router is named 'validating SR router'. Downstream routers may not inspect the HMAC field.

- this validating router can also have a cache of <IPv6 header + SRH, HMAC field value> to improve the performance. It is not the same use case as in IPsec where HMAC value was unique per packet, in SRH, the HMAC value is unique per flow.

- o Last point, hash functions such as SHA-2 have been optmized for security and performance and there are multiple implementations with good performance.

With the above points in mind, the performance impact of using HMAC is minimized.

5.2.3. Pre-shared key management

The field HMAC Key-id allows for:

- o key roll-over: when there is a need to change the key (the hash pre-shared secret), then multiple pre-shared keys can be used simultaneously. The validating routing can have a table of <HMAC Key-id, pre-shared secret, hash algorithm> for the currently active and future keys.

- o different algorithm: by extending the previous table to <HMAC Key-id, hash function, pre-shared secret>, the validating router can also support simultaneously several hash algorithms (see section Section 5.2.1)

The pre-shared secret distribution can be done:

- o in the configuration of the validating routers, either by static configuration or any SDN oriented approach;

- o dynamically using a trusted key distribution such as [RFC6407]

The intent of this document is NOT to define yet-another-key-distribution-protocol.

5.3. Deployment Models

5.3.1. Nodes within the SR domain

The routers inside a SR domain can be trusted to generate the outer IP header and the SRH and to process SRH received on interfaces that are part of the SR domain. These nodes MUST drop all SRH packets received on any interface that is not part of the SR domain and containing a SRH whose HMAC field cannot be validated by local policies. This includes obviously packet with a SRH generated by a non-cooperative SR domain.

If the validation fails, then these packets MUST be dropped, ICMP error messages (parameter problem) SHOULD be generated (but rate limited) and SHOULD be logged.

5.3.2. Nodes outside of the SR domain

Nodes outside of the SR domain cannot be trusted for physical security; hence, they need to obtain by some trusted means (outside of the scope of this document) a complete SRH for each new connection (i.e. new destination address). The received SRH MUST include a HMAC Key-id and HMAC field which has been computed correctly (see Section 5.2).

When a outside the SR domain sends a packet with a SRH and towards a SR domain ingress node, the packet MUST contain the HMAC Key-id and HMAC field and the the destination address MUST be an address of a SR domain ingress node .

The ingress SR router, i.e., the router with an interface address equals to the destination address, MUST verify the HMAC field with respect to the HMAC Key-id.

If the validation is successful, then the packet is simply forwarded as usual for a SR packet. As long as the packet travels within the SR domain, no further HMAC check needs to be done. Subsequent routers in the SR domain MAY verify the HMAC field when they process the SRH (i.e. when they are the destination).

If the validation fails, then this packet MUST be dropped, an ICMP error message (parameter problem) SHOULD be generated (but rate limited) and SHOULD be logged.

5.3.3. SR path exposure

As the intermediate SR nodes addresses appears in the SRH, if this SRH is visible to an outsider then he/she could reuse this knowledge to launch an attack on the intermediate SR nodes or get some insider knowledge on the topology. This is especially applicable when the path between the source node and the first SR domain ingress router is on the public Internet.

The first remark is to state that 'security by obscurity' is never enough; in other words, the security policy of the SR domain SHOULD assume that the internal topology and addressing is known by the attacker.

IPsec Encapsulating Security Payload [RFC4303] cannot be use to protect the SRH as per RFC4303 the ESP header must appear after any routing header (including SRH).

When the SRH is not generated by the actual source node but by an SR domain ingress router, it is added after a new outer IP header, this means that a normal traceroute will not reveal the routers in the SR domain (pretty much like in a MPLS network) and that if ICMP are generated by routers in the SR domain they will be sent to the ingress router of the SR domain without revealing anything to the outside of the SR domain.

To prevent a user to leverage the gained knowledge by intercepting SRH, it it recommended to apply an infrastructure Access Control List (iACL) at the edge of the SR domain. This iACL will drop all packets from outside the SR-domain whose destination is any address of any router inside the domain. This security policy should be tuned for local operations.

5.3.4. Impact of BCP-38

BCP-38 [RFC2827], also known as "Network Ingress Filtering", checks whether the source address of packets received on an interface is valid for this interface. The use of loose source routing such as SRH forces packets to follow a path which differs from the expected routing. Therefore, if BCP-38 was implemented in all routers inside the SR domain, then SR packets could be received by an interface which is not expected one and the packets could be dropped.

As a SR domain is usually a subset of one administrative domain, and as BCP-38 is only deployed at the ingress routers of this administrative domain and as packets arriving at those ingress routers have been normally forwarded using the normal routing information, then there is no reason why this ingress router should drop the SRH packet based on BCP-38. Routers inside the domain commonly do not apply BCP-38; so, this is not a problem.

6. IANA Considerations

TBD but should at least require a new type for routing header

7. Manageability Considerations

TBD should we talk about traceroute? about SRH in ICMP replies?

8. Contributors

The authors would like to thank Dave Barach, John Leddy, John Brzozowski, Pierre Francois, Nagendra Kumar, Mark Townsley, Christian Martin, Roberta Maglione, James Connolly, Aloys Augustin and Fred Baker for their contribution to this document.

9. Acknowledgements

TBD

10. References

10.1. Normative References

[FIPS180-4]
National Institute of Standards and Technology, "FIPS 180-4 Secure Hash Standard (SHS)", March 2012, <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <http://www.rfc-editor.org/info/rfc4303>.

[RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, DOI 10.17487/RFC5095, December 2007, <http://www.rfc-editor.org/info/rfc5095>.

[RFC6407] Weis, B., Rowles, S., and T. Hardjono, "The Group Domain of Interpretation", RFC 6407, DOI 10.17487/RFC6407, October 2011, <http://www.rfc-editor.org/info/rfc6407>.

10.2. Informative References

[I-D.ietf-isis-segment-routing-extensions]
Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and J. Tantsura, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-05 (work in progress), June 2015.

[I-D.ietf-ospf-ospfv3-segment-routing-extensions]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPFv3 Extensions for Segment Routing", draft-ietf-ospf-ospfv3-segment-routing-extensions-03 (work in progress), June 2015.

[I-D.ietf-spring-ipv6-use-cases]
Brzozowski, J., Leddy, J., Leung, I., Previdi, S., Townsley, W., Martin, C., Filsfils, C., and R. Maglione, "IPv6 SPRING Use Cases", draft-ietf-spring-ipv6-use-cases-05 (work in progress), September 2015.

[I-D.ietf-spring-problem-statement]
Previdi, S., Filsfils, C., Decraene, B., Litkowski, S., Horneffer, M., and R. Shakir, "SPRING Problem Statement and Requirements", draft-ietf-spring-problem-statement-04 (work in progress), April 2015.

[I-D.ietf-spring-resiliency-use-cases]
Francois, P., Filsfils, C., Decraene, B., and R. Shakir, "Use-cases for Resiliency in SPRING", draft-ietf-spring-resiliency-use-cases-01 (work in progress), March 2015.

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and r. rjs@rob.sh, "Segment Routing Architecture", draft-ietf-spring-segment-routing-05 (work in progress), September 2015.

[I-D.ietf-spring-segment-routing-mpls]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Shakir, R., Tantsura, J., and E. Crabbe, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-01 (work in progress), May 2015.

[RFC1940] Estrin, D., Li, T., Rekhter, Y., Varadhan, K., and D. Zappala, "Source Demand Routing: Packet Format and Forwarding Specification (Version 1)", RFC 1940, DOI 10.17487/RFC1940, May 1996, <http://www.rfc-editor.org/info/rfc1940>.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <http://www.rfc-editor.org/info/rfc2104>.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <http://www.rfc-editor.org/info/rfc2827>.

[RFC4942] Davies, E., Krishnan, S., and P. Savola, "IPv6 Transition/ Co-existence Security Considerations", RFC 4942, DOI 10.17487/RFC4942, September 2007, <http://www.rfc-editor.org/info/rfc4942>.

[RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, DOI 10.17487/RFC6554, March 2012, <http://www.rfc-editor.org/info/rfc6554>.

Authors’ Addresses

Stefano Previdi (editor)
Cisco Systems, Inc.
Via Del Serafico, 200
Rome 00142
Italy

Email: sprevidi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Brussels
BE

Email: cfilsfil@cisco.com

Brian Field
Comcast
4100 East Dry Creek Road
Centennial, CO 80122
US

Email: Brian_Field@cable.comcast.com

Ida Leung
Rogers Communications
8200 Dixie Road
Brampton, ON L6T 0C1
CA

Email: Ida.Leung@rci.rogers.com

Jen Linkova
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

Email: furry@google.com

Ebben Aries
Facebook
US

Email: exa@fb.com

Tomoya Kosugi
NTT
3-9-11, Midori-Cho Musashino-Shi,
Tokyo 180-8585
JP

Email: kosugi.tomoya@lab.ntt.co.jp

Eric Vyncke
Cisco Systems, Inc.
De Kleetlaann 6A
Diegem 1831
Belgium

Email: evyncke@cisco.com

David Lebrun
Universite Catholique de Louvain
Place Ste Barbe, 2
Louvain-la-Neuve, 1348
Belgium

Email: david.lebrun@uclouvain.be