D. Lu
M. Chen
Li. Su
China Mobile
Wei. Pan
Cheng. Li
Huawei Technologies
June 28, 2021

SRH and IP header protection
draft-chen-spring-srv6-srh-security-01

Abstract

This document proposes a method to protect SRH and IP header using signature which stored in the TLV, this scheme can apply to SRv6 and G-SRv6.

Status of This Memo

This Internet-Draft will expire on December 30, 2021.

Copyright Notice

Table of Contents

1. Introduction

SRv6 is a protocol for forwarding IPv6 packets over a network based on the concept of source routing. By inserting a Segment Routing Header (SRH) into the IPv6 packet, an explicit IPv6 address stack is pressed into the SRH, and the destination address and offset address stack are constantly updated by the intermediate node to complete hop-by-hop forwarding, SRH is defined in RFC8754 [RFC8754]

G-SRv6 is generalized Segment Routing over IPv6 which can reduce the overhead of SRv6 by encoding the Generalized SIDs in SID list, the compression solution is designed in the draft [I-D.cl-spring-generalized-srv6-for-cmpr].

As an emerging source routing protocol, SRv6 is confronted with various threat of source routing attacks. By defining SRH, attackers can construct various source routing attacks, such as bypassing key detection nodes of network and constructing malicious loops.

SRv6 networks generally define SRv6 trust domains for basic security protection, which is also mentioned in the draft [I.D.li-spring-srv6-security-consideration] and RFC 8754 [RFC8754]. Firstly, the address space in the SRv6 trust domain is defined to avoid SRv6 trust domain address leakage. Then ACL filtering is enabled at the boundary of the trust domain, and packets whose destination address is SRv6 trust domain are discarded to avoid source routing attack on SRV6 trust domain by attacking packets.

SRv6 trust domains use Segment Bingding technology for basic security. RFC8754 defines SRv6 HMAC TLV for IPv6 source address and SRH integrity protection which based on SRv6 trust domain, identity

authentication based on the shared key, to prevent illegal access and tamper header, so as to prevent various source routing attacks. However, there is a problem with this scheme, HMAC verification is based on symmetric key verification, that means all network nodes that need to be verified have to share the same key, there may exist a problems.

Secret key leak problem: when a single point's key was leaked, then all the trust domain was compromised.

In this document we present an alternative method for Segment Routing Header protection.

2. Terminology

This document uses the terminology defined in [RFC8754].

3. New TLV Type for Signature

This section describes how to use the certificate to authenticate the header. The source address field in IP header and several fields in SRH are protected by signature, and the result of signature is stored in TLV, the TLV format is consistent with the HMAC TLV defined in RFC8754, we describe this in Figure 1.

By defining a new type of TLV which the Type is 6 and we call it Auth TLV, indicates that the TLV is used for signature protection based on asymmetric secret keys. Auth TLV is described in Figure 1.

```
+--------+---------------------------+
|  Type  |  Length |D| RESERVED      |
+--------+---------------------------+
|          AUTH Key ID(4 octets)     |
+------------------------------------+
|                                    |
|          AUTH(variable)            |
|                                    |
+------------------------------------+
```

Figure 1: Auth TLV format

Type: 6.

Length: The length of the variable-length data in bytes.

D: 1 bit. 1 indicates that the Destination Address verification is disabled due to use of a reduced Segment List.

RESERVED: 15 bits. MUST be 0 on transmission.

AUTH Key ID: A 4-octet opaque number that uniquely identifies the hash algorithm, signature algorithm, and certificate serial number used for signature authentication.

AUTH: the content of the signature that protects the field, in multiples of 8 octets, at most 32 octets.

The AUTH TLV is used to protect IPv6 source address, SRH header for signature protection. Which fields are in the range of the signature check? they are described in Figure 2 and Figure 3, Figure 2 is for SRv6 and Figure 5 is for G-SRv6.

The AUTH Key ID field is opaque--i.e.,it has neither syntax nor semantic except as an identifier of the right combination of hash algorithm, signature algorithm and certificate serial number

Hash Algorithm indicates the hash algorithm used in the header, such as SHA256, and we do not recommend using SHA1.

Signature Algorithm indicates the asymmetric signature algorithm used, such as ECDSA and RAS2048.

Certificate Serial number used to identify certificate that issued by CA, if a custom certificate is used, the Certificate Serial number represents the identity of the custom certificate.

4. SRH protection used in SRv6 and G-SRv6

Segment routing header is defined in RFC8754, when user choose to use the method proposed in this draft, the complete SRv6 header with Auth TLV is show as figure 2, and figure 3 is for G-SRV6.

```
   +--------------+----------------+----------------------------+
   |  Version     |  Traffic class | Flow Label                 |
   +--------------+--------------------------------+------------+
   |    Payload Length            |  Next=43      |  Hop Limit |
   +-------------------------------+---------------+------------+
   |               Source Address                               |
   +------------------------------------------------------------+
   |               Destination Address                          |
   +--------------+---------------------------------------------+
   | Next Header  | Hdr Ext Len    |Routing Type=4 |Segment Left|
   +------------------------------------------------------------+
   | Last Entry   | Flags          |       Tag                  |
   +--------------+----------------+----------------------------+
   |                 Segment List[0]                            |
   +------------------------------------------------------------+
   |                 Segment List[1]                            |
   +------------------------------------------------------------+
   |                 Segment List[2]                            |
   +--------------+----------------+---+------------------------+
   |  Type=6      |  Length        | D |    Reserved            |
   +--------------+----------------+---+------------------------+
   |                   Auth Key ID                              |
   +------------------------------------------------------------+
   |                  Auth(variable)                            |
   +------------------------------------------------------------+
   |                IPv6 Payload                                |
   +-------------------------------------------------------------
```

Figure 2: Complete SRv6 header with Auth TLV

Figure 5 is the detailed structure for G-SRv6

```
+--------------+----------------+----------------------------+
|  Version     |  Traffic class | Flow Label                 |
+--------------+--------------------------------+------------+
|    Payload Length            |  Next=43      |  Hop Limit |
+-------------------------------+---------------+------------+
|               Source Address                               |
+------------------------------------------------------------+
|               Destination Address                          |
+--------------+---------------------------------------------+
| Next Header  | Hdr Ext Len    |Routing Type=4 |Segment Left|
+------------------------------------------------------------+
| Last Entry   | Flags          |       Tag                  |
+--------------+----------------+----------------------------+
|                 G-SID Container[0]                         |
+------------------------------------------------------------+
|                 G-SID Container[1]                         |
+------------------------------------------------------------+
|                 G-SID Container[2]                         |
+--------------+----------------+---+------------------------+
|  Type=6      |  Length        | D |    Reserved            |
+--------------+----------------+---+------------------------+
|                   Auth Key ID                              |
+------------------------------------------------------------+
|                  Auth(variable)                            |
+------------------------------------------------------------+
|                IPv6 Payload                                |
+-------------------------------------------------------------
```

Figure 3: Complete SRv6 header with Auth TLV

Signature check those fields that need to be protected will be signed, the range of signatures includes IPv6 Source address, SRH Last Entry, SRH Flags, SRH Segment List, AUTH TLV D, AUTH TLV Reserved, AUTH TLV Auth Key ID.

5. signing and verifying process

First, need the CA center to issue a root certificate to the controller that will generate controller's public and private key, or the controller use custom certificate, it depends on the detail implementation. How to preset and update a CA certificate on a device is out of scope in this document. The process described in this document uses CA certificates by default.

SRv6 controller uses the private key of the certificate to hash the SRH and IP header, and encapsulates the digital signature generated by SRv6 header and controller in the SRV6 source node. The signature process is divided into three steps.

Step1: Preset certificates, include private keys and controller certificates on SRv6 controllers, and CA root certificates on key network devices;

Step2: After the secure connection is established between the controller and the network device on the control plane, perform public key certificate distribution and signature algorithm selection, and inform the key node the selection result.

Step3: SRv6 controller uses the private key, the hash algorithm and the asymmetric algorithm selected in the step2 to sign the packet header which generated according to the routing result, and store the signature results in the TLV, finally sends the routing result which include the signature to the source node, the source node wraps and forwards an SRv6 packet with a signature, the SRv6 network structure is described in Figure 4.

```
                  +------------+ Public key
                  | Controller | Private key
                  +-----^------+
                        |
    +------------+------+-----+------------+
    |            |            |             |
    |            |            |             |
    |            |certificate |             |
    |            |            |             |
    |            |            |             |
+---+--+     +---v--+      +--+---+     +---+--+
| RT A +-----+ RT B +------+ RT C +-----+ RT D |
+------+     +------+      +------+     +------+
```

Figure 4: SRv6 network structure

Signature verification is required at key network nodes, it's also divided into three steps.

Step1: Enable signature verification at the key nodes.

Step2: Request a public key certificate from the controller.

Step3: calculate the hash value according to the header, and use the public key to decrypt the signature in the message, compare the decryption result with the hash value, if verify successful, forward the message, otherwise, the message is discarded.

6. verifying optimization process

When asymmetric key is used to verify the signature of the forward message on the data plane, the processing efficiency of the forward message is reduced. An efficient lookup table forwarding mechanism for signature verification can be considered, which verifies the signature of the first packet of the data, and records the hash result and signature of the packet header into the hash table. The subsequent packets can directly find the hash table and compare the signature result, no more need to decrypt, also can divide into three steps.

Step1: When the interface of signature verification is opened and the SRV6 message is received, the hash value of the message header is calculated and finds if the local hash table is hit, the local hash table contains hash value and signature value, and they are bound.

Step2: If the local hash table is not hit, the controller's public key is used to decrypt the signature and compare whether the decrypted result is consistent with the calculated hash value. If not, the message is discarded. If the hash value and decrypted result are consistently then recorded to the local hash table, and the processing packet is forwarded.

Step3: If the local hash table is hit, the signature value in message is compared with hash table's signature value, if yes then forwarded to process the message, if not then discarded.

7. Security Considerations

SRv6 is threatened by various source routing attacks. By defining SRH, an attacker can construct various source routing attacks, such as bypassing the key detection nodes of the network and constructing malicious loops, in this draft we propose a method, it can prevent a single device from being compromised and exposes the network's shared key, then the entire network is under threat.

8. IANA Considerations

This document does not require any action from IANA.

9. Acknowledgement

TBD

10. Normative References

[I-D.cl-spring-generalized-srv6-for-cmpr]
Cheng, W., Li, Z., Li, C., Clad, F., Liu, A., Xie, C., Liu, Y., and S. Zadok, "Generalized SRv6 Network Programming for SRv6 Compression", draft-cl-spring-generalized-srv6-for-cmpr-03 (work in progress), April 2021.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

Authors' Addresses

Dongjie Lu
China Mobile
32, Xuanwumen West
BeiJing, BeiJing 100053
China

Email:
ludongjie@chinamobile.com

Meiling Chen
China Mobile
32, Xuanwumen West
BeiJing, BeiJing 100053
China

Email:
chenmeiling@chinamobile.com

Li Su
China Mobile

32, Xuanwumen West

BeiJing

100053

China

Email:
suli@chinamobile.com

Wei Pan
Huawei Technologies

101 Software Avenue, Yuhuatai District

Nanjing

China

Email:
william.panwei@huawei.com

Cheng Li
Huawei Technologies

Huawei Campus, No. 156 Beiqing Rd.

Beijing

100095

China

Email:
c.l@huawei.com