A Path Verification Solution based on SRv6

draft-jliu-tpp-srv6-00

Brisbane Meeting @ IETF 119, March 2024
Jun Liu
Tsinghua University
Contents

• What is Path Verification for?
• Existing works.
• Path Verification Solution on SRv6.
• Core Use Case.
What is Path Verification for?

• Packet tampering and source spoofing.

• Path deviation.
  ○ Router hop pass.
  ○ Router addition.
  ○ Out-of-order forwarding.

• Denial of Service (DoS).

• Privacy leak.
Existing works-Two ways

• Verifying the planned paths– if it is trusted and authorized.

• Verifying the traversed paths-if the packet has actually traversed.

New Challenge: Ensure that the integrity of the original path is not destroyed when the network flow is forwarded by untrusted nodes
Terminology

• MAC: Message Authentication Code, a technology to confirm the integrity and conduct certification.

• SRH: [RFC8754] Source Routing Head.


• Dos: Denial of Service.

• Tag: Mark a packet as part of a class or group of packets. This field initialized with a timestamp value to represent a unique session.

• Segments Left: Number of remaining routing segments, defined in [RFC8200] Section 4.4.

• SID: the label of the intermediate router.

• SL: Segments List, an ordered list of SID.

• IR: Intermediate Router, routers participating in packet forwarding in the path.
Path Verification Solution on SRv6

Handle the predetermined forwarding path risks caused by untrusted intermediate routers. The specific functional goals are as follows:

• **Source and path verification.**
  - Extends the existing SRv6 routing header.
  - Lightweight operations.

• **Privacy protection.**

• **Fault localization.**
  - Provides distributed path verification and centralization-based fault localization.

Reduce the header overhead and introduce privacy protection in the path verification mechanism.
Initialization

• **Path Initialization.**
  - Use the local key of each SR router and the path creation time to generate the session key using hash function with session initiation timestamp.
  - Notify the sender of the session key and IP address of the entire path.

• **Package Initialization.**
  - Initialize the Tag field of the SRH when the path was created.
  - Generate its SID and write into the Segment List(SL).
  - Calculate MAC(SID) and insert into SL for subsequent SID generation.

---

**SRv6 header format**

<table>
<thead>
<tr>
<th>Next Header</th>
<th>Hdr Ext Len</th>
<th>Routing Type</th>
<th>Segments Left</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPv6 Header</td>
<td></td>
<td>SRH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Segment list[0]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Segment list[n]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Path Initialization Process**

1. Start
2. Select nodes in the path
3. Generate session keys in order
4. Judge whether it is the last node?
   - Yes: Send the session key and IP address of the path to the source
   - No: Go to Step 2
Verification

- **Path Initialization Key generation.**
  - Intermediate Router (IR) computes the session key using the IR’s local key and timestamp.
  - The key generation is stateless.

- **Upstream verification.**
  - IR generates a MAC with the partial segment list from the packet Header and the payload.
  - Verifies SL and update by upstream router.

- **Obtain downstream IP address.**
  - IR then parses the SID with K to obtain the IPv6 address of its downstream router.

- **Replace header.**
  - Router replaces the source and destination filed of IPv6 Header with itself IPv6 address and its downstream router’s IPv6 address.
Fault location & Security Analysis

• Fault location.
  - Payload alteration by the upstream router.
  - Malicious redirection by other routers.

• Security Analysis.
  - Packet alteration.
  - Path deviation.
  - Denial-of-Service (DoS).
  - Privacy leaks.
Core Use Case: Proof of TPP-processing

**Core Use Case**

**Proof of TPP-processing**

**Path Initialization Model**

**Fault localization Model**

**Data Generation Model**

**Communication Model**

**SRH Generation Model**

**Packet Verification Model**

**Forwarding Model**

**Fault Management Model**

**Communication Model**

**Data Packet Verification Model**

**Sender**

**SR Router**

**Receiver**

**SR controller**

**DELL XPS (IntelCorei 79700, 3GHZ, 8core, 16GB, Ubuntu)**
Normative References


Path Verification solution on SRv6
Thank you! Questions?

Brisbane Meeting @ IETF 119, March 2024
Jun Liu
juneliu@tsinghua.edu.cn
draft-jliu-tpp-srv6-00