Implementing IPv6 Segment Routing in the Linux Kernel

David Lebrun, Olivier Bonaventure
ICTEAM, UCLouvain

Work supported by ARC grant 12/18-054 (ARC-SDN) and a Cisco grant
Agenda

• IPv6 Segment Routing

• Implementation in the Linux kernel

• Performance evaluation
What is Segment Routing?

- The return of Source Routing
  - Each packet contains a loose route to encode any path inside the network
IPv6 Segment Routing

• Basic principles
  – IGP distributes IPv6 prefixes and router loopback addresses
  – Loose source route encoded inside IPv6 extension header containing a list of segments

– Main types of segments
  • Node segment (router's loopback address)
  • Adjacency segment (router outgoing interface)
  • Virtual function (operator defined function)

http://www.segment-routing.net

The IPv6 Segment Routing Header

The IPv6 Segment Routing Header is used to specify the path that IPv6 packets should take through a network. Each segment is one IPv6 address, and the header includes fields for the segment list, index of the last segment, and remaining segments. The header also includes fields for the next header, header extension length, routing type, and segments left.

Extensibility is provided by optional type length value (TLV) objects, which can be included in the header as needed.

Remaining segments
Index of last segment
Each segment is one IPv6 address
Extensibility
IPv6 Segment Routing use cases

- Paths controlled by the endhosts

  Source adds SRH to all packets

  Destination removes SRH from all packets
Network Function Virtualisation

- Force packets to pass through NFV
Encap and decap

- Routers can also tunnel SRH packets

Ingress router encaps to R6 with SRH

Egress router decaps and removes SRH
Security: Learning from the past

• How to avoid past failures of source routing?

Security Problems in the TCP/IP Protocol Suite

S.M. Bellovin*
smb@ulysses.att.com
AT&T Bell Laboratories
Murray Hill, New Jersey 07974

ABSTRACT

The TCP/IP protocol suite, which is very widely used today, was developed under the sponsorship of the Department of Defense. Despite that, there are a number of serious security flaws inherent in the protocols, regardless of the correctness of any implementations. We describe a variety of attacks based on these flaws, including sequence number spoofing, routing attacks, source address spoofing, and authentication attacks. We also present defenses against these attacks, and conclude with a discussion of broad-spectrum defenses such as encryption.

Network Working Group
Request for Comments: 5095
Updates: 2460, 4294
Category: Standards Track

J. Abley
Afilias
P. Savola
CSC/FUNET
G. Neville-Neil
Neville-Neil Consulting
December 2007

Deprecation of Type 0 Routing Headers in IPv6

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

The functionality provided by IPv6's Type 0 Routing Header can be exploited in order to achieve traffic amplification over a remote path for the purposes of generating denial-of-service traffic. This document updates the IPv6 specification to deprecate the use of IPv6 Type 0 Routing Headers, in light of this security concern.
The IPv6 SRH HMAC TLV

Different keys and different hash functions can be used
Utilisation of the HMAC TLV

• All routers are configured with an HMAC key

• Clients receive SRH with HMAC key
  – E.g. from SDN controlled

• Trusted servers configured with HMAC key
Agenda

• IPv6 Segment Routing

• Implementation in the Linux kernel

• Performance evaluation
Basics of Linux packet processing

Packet received for local process

Forwarded packet

Packet sent by local process
Packet forwarding with IPv6 SR

- Router is one of the segments in the list

SRH updated, Packet forwarded to next segment
Packet forwarding with IPv6 SR

• Egress router receiving encapsulated packet

SRH processed encapsulation removed
How to configure IPv6 SR?

- IPv6 SR implementation extends iproute2
  - Commands passed through rtnetlink
  - Example

```bash
ip -6 route add fc42::/64
    encap seg6 mode encaps
    segs fc00::1,2001:db8::1,fc10::7
    dev eth0
```

- Destination match

- SRv6 encapsulation

- Segments added in the encapsulated packet
SRH usage by applications

- Endhosts can control the SRH on a per flow basis through the socket API

```c
struct ipv6_sr_hdr *srh;
int srh_len;

srh_len = build_srh(&srh);
fds = socket(AF_INET6, SOCK_STREAM, IPPROTO_TCP);
setsockopt(fd, IPPROTO_IPV6, IPV6_RTHDR, srh, srh_len);
```
HMAC processing

• Three modes of operations can be configured
  – Ignore
    • All packets are forwarded independently of the HMACs
  – Verify
    • Packets containing an HMAC are processed if HMAC is valid
    • Packets without HMAC are processed
  – Enforce
    • Packets containing an HMAC are processed if HMAC is valid
    • Packets without HMAC are processed
Agenda

- IPv6 Segment Routing
- Implementation in the Linux kernel
- Performance evaluation
Lab measurements

- Lab setup
  - Intel Xeon X3440 processors (4 cores 8 threads at 2.53 GHz)
  - 16 GB of RAM
  - Two Intel 82599 10 Gbps Ethernet
    - One queue per CPU, one IRQ per queue
  - Linux kernel 4.11-rc3, TSO and GRO disabled

- Traffic generator
  - Pktgen, in-kernel module sending UDP packets
First measurements with one CPU

Why this gap?

Baseline
Plain IPv6 forwarding

No difference between SRH forwarding and encap+forwarding
Performance limitations of the first implementation

• Route lookup
  – Destination cache was implemented for locally generated packets but not forwarded ones
    • Fixed with a dest cache

• Issue with memory allocation
  – Forced free to take a slow path involving spinlocks in case packet was processed by different CPU than NIC IRQ
    • Fixed with a better utilisation of the skb
Improved performance on one CPU

IPv6 SRH forwarding and encap are now close to plain IPv6 packet forwarding performance.
Does packet size affect performance?
Cost of HMAC

Performances comparison (HMAC)

Throughput (Kpps)

Encap  |  HMAC (generic)  |  HMAC (ssse3)

Pure C code

Special intel instructions
Leveraging multiple cores

Performance scales well with the number of physical CPUs
Conclusion

• IPv6 Segment Routing has matured
  – Stable specification
  – Various use cases

• Implementation in the Linux kernel 4.11+
  – Endhost functions for clients and servers
  – Router functions

• Performance evaluation
  – Good forwarding and encap/decap performance
  – Unsurprisingly HMAC TLV affects performance

http://www.segment-routing.org