Journal of Ubiquitous Systems & Pervasive Networks
Volume 18, No. 1 (2023) pp. 15-21

Segment Routing v6 - Security Issues and Experimental Results

David Lo Bascio [a]*, Flavio Lombardi [bc]

[a]*Dept. of Information Engineering, Electronics and Telecommunications (DIET) "Sapienza" University of Rome, Italy*
[b]*Istituto per le Applicazioni del Calcolo, Consiglio Nazionale delle Ricerche (IAC-CNR), Rome, Italy*
[c]*Member of the INdAM-GNCS research group*

Abstract

SRv6 can provide hybrid cooperation between a centralized network controller and network nodes. IPv6 routers maintain multi-hop ECMP-aware segments, whereas the controller establishes a source-routed path through the network. Since the state of the flow is defined at the ingress to the network and then is contained in a specific packet header, called Segment Routing Header (SRH), the importance of such a header itself is vital. Motivated by the need to study and investigate this technology, this paper discusses some security-related issues of Segment Routing. A SRv6 capable experimental testbed is built and detailed. Finally, an experimental test campaign is performed and results are evaluated and discussed.

Keywords: *Segment Routing, Networking, Security*

1. Introduction

Today's pervasive networks are increasingly smart and flexible [1],[2],[3]. This is also due to the advanced technologies, together with IPv6 support, that are deployed on network devices (both physical ones and virtual ones).

Traffic Engineering (TE) in IP carrier networks is one of the functions that can benefit from the Software Defined Networking (SDN) paradigm [4],[5]. Nevertheless, traditional per-flow routing requires a direct interaction between the SDN controller and each node that is involved in the traffic paths. Segment Routing (SR) is one technology that can help simplify route enforcement by delegating all the configuration and per-flow state at the border of the network.

In the traditional routing approach [6] a distributed intelligence is used: each decision on the traffic path is taken on the packet by each node of the network. In fact, conventional routers in the network determine the path incrementally based on the packet destination. New networking paradigms such as SDN have introduced a centralized optimization but require maintaining a per-flow state on each node.

A Segment Routing (SR) architecture allows including a list of instructions (i.e. segments) in the packet header [7]. This can provide hybrid centralized/distributed cooperation between the controller and the network, where the network maintains the multi-hop ECMP-aware [8] segments while the centralized controller combines them to form a source-routed path through the network. In SR the state is removed from the network and it is only present at the ingress to the network and then in the packet header itself.

The IPv6 protocol has many features including the expanded addressing capability, auto-configuration mechanism, simplification of the header format, improved support for extensions and options (see [9] and [10]), extensions for authentication and privacy, flow labeling capability and so on.

For these reasons SR can be instantiated over the IPv6 data plane, in what is Segment Routing v6 (SRv6) [11], using a new type of Routing Extension Header called the Segment Routing Header (SRH).

The main goal of our work is to address the lack of a detailed and comprehensive discussion and experimental validation and evaluation of the potential SRv6 security issues. As such, this paper aims at introducing and discussing the technological context, at building a realistic experimental testbed and at providing some initial experimental results validating the idea that SRv6 technology can be potentially misused and cause security and performance issues. Among the potential stakeholders of our work, researchers in network and security, network admins and

[*] Corresponding author. Tel.: 39-06-44585365
Fax: +39-06-44585632; E-mail: david.lobascio@uniroma1.it

DOI: 10.5383/JUSPN.18.01.003

engineers that want a better view on the pros and cons of using SRv6 technology in practice. Present work is relevant to such stakeholders in that it induces a better knowledge and awareness of the specific network functionality and flows that have to be monitored and further studied.

1.1. Main Contributions and Layout

This paper builds on the findings in [12] and extends and improves over previous paper in that it:

- enriches and updates the Related Work section;
- implements a fully functional SRv6 experimental testbed;
- details how to create an SRv6 scenario leveraging freely available and open source software;
- performs a preliminary experimental campaign;
- evaluates and discusses the collected experimental results.

This paper is organized as follows: Section 2. provides backgound information shedding light on the considered technologies; Section 3. surveys related work on SRv6 technology and on its performance and security; Section 4. provides details of the implemented testbed including the leveraged software tools; Section 5. shows the test plan and discusses the results of the collected experimental efforts; Finally Section 6. draws some conclusions and provides hints for future work.

2. Technological Background

A **source-routing** architecture seeks the right balance between distributed intelligence and centralized optimization. Source routing allows the sender of a packet to partially or completely specify the route the packet takes through the network. Two main options exist: *Loose source routing* uses a source routing option in IP to record the set of routers a packet must visit; *Strict source routing* where every step of the route is decided in advance when the packet is sent.

The **Segment Routing (SR)** architecture is based on the loose source routing paradigm. A node steers a packet through an ordered list of instructions, called "segments". The list of segments represents an SR policy instantiated at the ingress node to the SR domain. A segment is often referred to by its Segment Identifier (SID), it can represent any kind of instruction. A segment associated with a **topological** instruction can be:

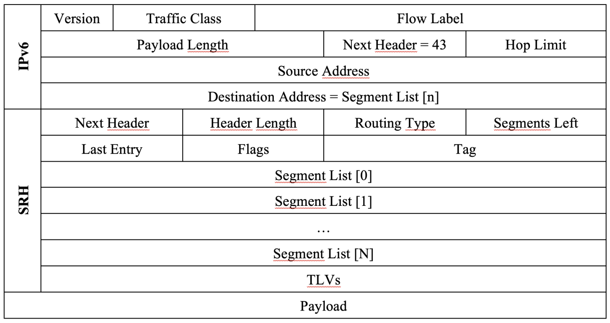
- a topological *local* segment, which may instruct a node to forward the packet via a specific outgoing interface;
- a topological *global* segment, which may instruct an SR domain to forward the packet via a specific path to a destination.

A segment can also be **service-based** – e.g., the packet should be processed by a container or Virtual Machine (VM) associated with the segment – or may be associated with a QoS treatment – e.g., shape the packets received with this segment at *x* Mbps. The SR architecture supports any type of instruction associated with a segment.

The SR architecture supports any type of **control plane**: distributed, centralized, or hybrid. In a *distributed* control plane segments are allocated and signaled by Intermediate System to Intermediate System (IS-IS) or Open Shortest Path First (OSPF) or Border Gateway Protocol (BGP): a node individually computes the SR Policy and decides on its own to steer packets based on that policy. In a *centralized* control plane, segments are allocated and instantiated by an SR controller: the SR controller computes the source-routed policies and decides which nodes need to steer which packets on those policies. The SR architecture does not restrict how the controller programs the network. A *hybrid* scenario complements a base distributed control plane with a centralized controller.

The SR architecture can be instantiated on various **data planes**: SR over Multi Protocol Label Switching (SR-MPLS) and SR over IPv6 (SRv6). SR can be directly applied to the *MPLS* architecture with no change to the forwarding plane: a segment is encoded as an MPLS label and an SR Policy is instantiated as a stack of labels. The segment to process (the active segment) is on the top of the stack. Upon completion of a segment, the related label is popped from the stack.

If SR uses an *IPv6* data plane, each instruction is associated with a segment and encoded as an IPv6 address. An SRv6 segment is also called an SRv6 SID. An SR Policy is instantiated as an ordered list of SRv6 SIDs in a new type of routing header called the SR Header (SRH); so, when a packet is steered on an SR Policy, the related SRH is added to the packet by a headend node – the Source SR node – that is a SR-capable router. SR Header (SRH) is created with Segment list in reversed order of the path; the active segment is indicated by the Destination Address (DA) of the packet and it is set to the first segment. The packet is sent according to the IP DA, through a normal IPv6 forwarding. The next active segment is indicated by the Segments Left (SL) pointer in the SRH. When a SRv6 SID is completed, the SL is decremented and the next segment is copied to the DA. The SRH is shown in Figure 1.



IPv6	Version	Traffic Class	Flow Label	
	Payload Length		Next Header = 43	Hop Limit
	Source Address			
	Destination Address = Segment List [n]			
SRH	Next Header	Header Length	Routing Type	Segments Left
	Last Entry	Flags	Tag	
	Segment List [0]			
	Segment List [1]			
	…			
	Segment List [N]			
	TLVs			
Payload				

Fig. 1. SRv6 Header

A Transit node forwards the packet containing the SR header as a normal IPv6 packet, so the Transit nodes do not need to be SRv6-aware. A transit node executes plain IPv6 forwarding, solely based on IPv6 DA; it doesn't inspect or update the SRH.

SR Endpoints are SR-capable nodes whose address is in the IP DA. They inspect the SRH and update the DA in the IPv6 Header according to the Segment Left and the Segment List specified by the SRH. After processing, the packet is forwarded according to the new IP DA.

In Segment Routing v6 it is possible to consider two kinds of nodes (routers and hosts):

- nodes belonging to a single SR domain where all nodes are trusted;

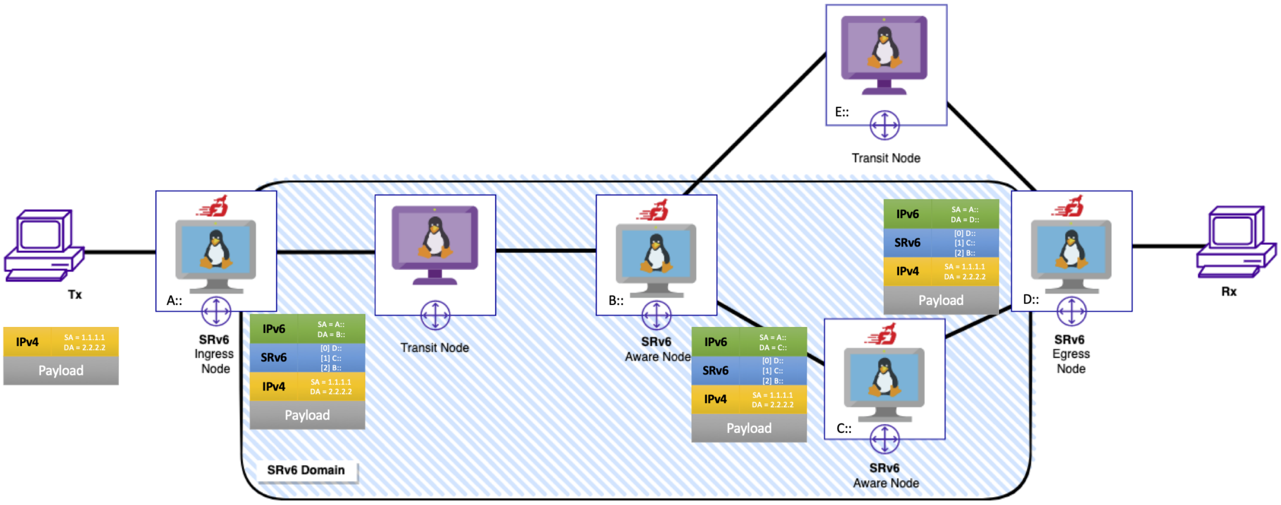


Fig. 2. SRv6 Packet Manipulation when Traveling among Nodes

- nodes outside of the SR domain, that cannot be trusted.

SRv6 is quite protected in a single administrative domain with trusted nodes, but its potentialities are limited. Further, SRv6 nodes ignore SRH created by external nodes, making the RFC 5095 attacks far more complex to perform.

In is worth noting that the security-related fields in SRH feature are:

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

The HMAC field is used to verify the validity of the SRH. Nevertheless, some tampering could still be possible due to the limited key length [13].

The SRv6 packet processing over contiguous nodes is shown in Figure 2. A a useful survey to better delve into Segment Routing can be found here [14].

3. Related Work

Interestingly, Mavani and Ragha [15] investigate the threats due to misusing IPv6 destination option and fragmentation extension headers. In particular fragmentation attack, overlapping fragmentation attack, and flooding attack are tested. The very same authors wrote another stimulating work [16] where covert channel using destination option extension header of IPv6 is discussed.

Djellali and Adda [17] investigate network attack detection using AI. This is out of the scope of our present work but will be useful for further network traffic analysis.

Some IPv6-related security issues are reported in [18]. The Segment Routing Header is an extension header of IPv6 used by an IPv6 source to list one or more intermediate nodes to be passed through by the packet on the path to a destination. One security issue comes from the fact that an attacker can detour the access list of security system, for example firewalls, and then he can access the protected internal system by using SRH. An enlightening article on LWN [19] mentions HMAC as a mitigating approach to the Segment Routing header tampering problem.

Using a SRH is a form of source routing, therefore it has some well-known security issues as described in RFC4942 [20] and RFC5095 [21] as explained in [22]:

- *amplification attacks*: where a packet is forged in such a way as to introduce loops among a set of SR-enabled routers, yielding unnecessary traffic, hence a Denial of Service (DoS) [23] against bandwidth;
- *reflection attack*: where an attacker forces an intermediate node to appear as the immediate attacker, hence hiding the real attacker from naive forensic;
- *bypass attack*: where an intermediate node is used as a step stone (for example in a De-Militarized Zone) to attack another host (for example in the data center or any back-end server).

RFC2460 [24] defines an IPv6 extension header called Routing Header, in particular a Routing Header subtype denoted as Type 0 a.k.a. RH0 is defined that may contain multiple intermediate node addresses, including repetitions. This allows a packet to be constructed such that it will oscillate between two RH0-processing hosts or routers many times. This property can be used to cause congestion and DoSes [25]. This attack is particularly serious in that it affects the entire path between the two exploited nodes, not only the nodes themselves or their local networks. Analogous functionality can be found in the IPv4 source route option, but the opportunities for abuse are greater with RH0 due to the ability to specify more intermediate node addresses in each packet. The severity of this threat was considered to be sufficient to warrant deprecation of RH0 entirely.

The above work inspired our effort aimed at investigating potential similar issues in SRv6.

Ally [26] is a platform allowing to sequester cores to run packet processing software appliances (e.g. for Deep Packet Inspection, DPI). Ally is aimed at low overhead packet interception, that can be very useful to perform sophisticated high performance packet tampering.

SRv6 security has been discussed in several works, we mention just a few here below.

Li and Xie [27] describe some threats and security concerns related to SRv6. Unfortunately do not consider some relevant security issues as they consider SR networks as "trusted domains". This document assumes that the SR-capable routers and transit IPv6 routers within the SRv6 trusted domains are trustworthy. Hence,the SRv6 packets are treated as normal IPv6 packets in transit nodes and the SRH will not bring new security problem. The question here is how strong and realistic the assumption of having trusted domains is.

Barton and Henry [28] show how a path computation element of a network configured for segment routing receives, from a plurality of path computation clients in the network, segment identifiers identifying a destination segment. They show how the above element also receives fatigue states for segments of the network to allow rerouting to proactively mitigate overloaded segments in the network.

Filsfils and Garvia [29] show how Segment Routing network nodes protect IPv6 Segment Routing (SRv6) using Security Segment Identifiers providing origin authentication, integrity of information and antireplay protection. Nevertheless, this is a patented approach with limited applicability.

Vyncke, Previdi and Lebrun [30] propose SR-TPP, a mechanism based on SRv6 to support network path verification while hiding both-end and path information. Unfortunately, the SR-TPP approach is distributed and this opens up some issues related to state transmission and potential further attacks to the distributed system.

Cravel et al. [31] discuss the impact of IP packet header modifications, present some techniques for detection, and define strategies to add tamper-evident protection into the Linux TCP stack. We believe this is interesting work and it will be investigated further in the future to help provide countermeasures to packet header tampering.

Wongang et al. [32] investigate an Advanced Encryption Standard (AES)-based routing algorithm (AODV-Wormhole Attack Detection Reaction) for securing AODV-based eMANETs against wormhole attacks. The paper studies the performance of the algorithm on devices that are incompatible with AES and introducing hash codes in the data packets to help data integrity.

The main limitation of the above cited previous work is the lack of a through investigation of the potential security issues of SRv6 and of an experimental validation of such issues. Present paper builds on the ground of above work and on [12] to help fill the existing gap on SRv6 security knowledge.

4. Experimental Testbed

This section provides an overview, motivation and details of the experimental campaign performed for this present work. Firstly, some relevant SRv6 issues are summarized. Secondly, the testbed configuration details are given, together with the objectives of the experimental campaign itself.

4.1. SRv6 Security Issues

Some of the most relevant attack scenarios for SRv6 are the following[12]:

- **Ingress SRv6 Node Attack**: this attack is based on having a compromised router at the beginning of the SR domain, i.e. the router responsible of SRH encapsulation.
- **Transit Node Attack**: this attack is based on having an SR-unaware router compromised. Such compromised node is passed through by a packet in the plain forwarding operation. This node is supposed not inspect or update the SRH. Nevertheless, it could try to alter IPv6 DA with a rogue SID, manipulating the SR policy;
- **SR Endpoint Attack**: this attack is based on having a SR-capable router compromised. This SR-aware node is supposed to inspect the SRH, update the Segment Left field in SRH, update tha DA in the IPv6 Header, and in case process the payload. Malicious actions on this node would have a great impact on SR policy and traffic redirecting.

In this paper we will focus on some experimental scenario related to the **Transit Node Attack** above, shown in Figure 3.

4.2. Testbed

The main motivation for implementing a fully functional testbed is that of observing and understanding the behavior of current SRv6 implementations with and without actual attacks being performed. This allowed to get better acquainted with the technology and to evaluate current implementation status to potentially evince weaknesses.

In order to evaluate SRv6 functionality and performance under different conditions, we implemented a simulation testbed similarly to what provided in [33]. We leveraged the EVE-NG [1] network emulation software as it allowed performing accurate simulation of real world scenarios by making use of actual software implementations inside network nodes (i.e. VMs).

The main motivations for choosing EVE-NG come from its scalability, flexibility and support for building network topologies of virtual nodes that allow performing complex network activities in a fully controlled realistic environment.

Our testbed is built on top of a HP Z4 G4 Workstation equipped with the following resources:

- CPU: 1 x Intel Xeon W-2123 4-code/8-thread CPU @ 3.60GHz;
- Memory: 40GB DDR4-2666 ECC Reg RAM;

The workstation is a physical server dedicated to emulating the virtual environment: EVE-NG acts as a platform, since it hosts and runs VMs, without the need to separately install an operating system and virtualization software, given that the EVE-NG platform leverages KVM [34]. The version in use is EVE-NG Community Edition Version 2.0.3-112.

Inside the EVE-NG environment, the logical topology is built as shown in Figure 2. VMs are configured as routers thanks to enabling packet forwarding among their 100Mb/s interfaces; each node is equipped with 8 GB RAM, while the guest OS is Ubuntu 18.04.6 LTS.

The Vector Packet Processing (VPP) platform is used on nodes (i.e. VMs) to support high performance packet-processing. VPP is the core technology behind the FD.io Project[2]. VPP is an open source stack that can run on commodity CPUs. Such technology

[1] https://www.eve-ng.net/

[2] https://fd.io/

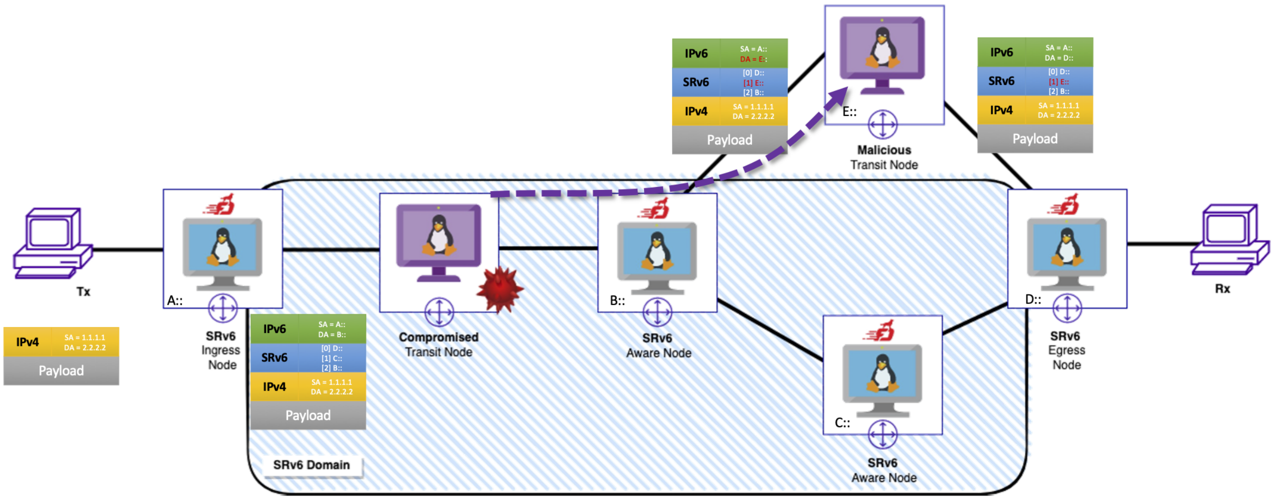


Fig. 3. Transit Node Attack Scenario

allows programming L2/L3 instructions without the need to change core/kernel code: the engine runs in pure userspace. Our aim is to define SRv6 policies and instructions on VMs, in order to easily configure them as SRv6-aware commodity routers. The VPP version currently in use is 21.10.1.

The actual SRv6 configuration is detailed below. First of all, a local SID has to be associated to a Segment Routing or function on each node. In our case, we have two possible behaviors:

- END: it is the most basic behavior, since it simply activates the next SID in the current packet, by decrementing the Segments Left value and updating the IPv6 Destination Address;
- END.DX4: this behavior decapsulates packets and forwards the decapsulated IPv4 packets over the Layer 3 interface bound to the SID.

The END behavior is associated to Ingress/Egress Nodes of SRv6 Domain, while the END.DX4 behavior is associated to Transit Nodes. These behaviors match the definitions of the SRv6 network programming architecture [35].

Secondly, an SR Policy has to be defined by the set of Segment Lists, that are SR-aware routers through which the packet will have to pass. The SID list has an impact on the Destination Address field on the IPv6 outer header. Furthermore, an IPv6 source address must be specified for the encapsulated traffic on the same header. Finally, traffic is steered into SR policy based on the destination network IP address.

In order to reliably perform test measurements an open source tool such as iPerf3[3] has been deployed.

The iPerf3 tool is a well known software for active network performance measurements, including obtaining the maximum achievable bandwidth and other data. It supports tuning of various parameters related to timing, buffers and protocols (TCP, UDP, etc.). For each test it reports the bandwidth, loss, and other data.

5. Test Plan, Results and Discussion

Having an actual implementation of the experimental testbed helps showing the feasibility and impact of enabling SRv6 on the above topology. An experimental validation of the collected results comes from having tested the testbed components in heterogeneous configurations and scenarios with and without SRv6 with convincing results. Our experimental campaign also serves as the basis for measuring and evaluating the impact of the attacks on the functionality and performance of the network. In future work, it will also allow evaluating and comparing different mitigation approaches.

The main metrics used here are the end-to-end throughput and the receiving and transmitting node CPU usage, the measured end-to-end packet loss and the jitter in the packet arrival times. We believe they are adequate for our present experimental activity. The initial feasibility tests have been performed on the basic topology shown in Figure 2 without SRv6 encapsulation. The VPP dataplane is used on all nodes, but normal IPv6 traffic is sent end-to-end, in order to establish the network performance baseline. We have generated two different kinds of traffic flows: in the first case it is a 100 Mbit/s data transfer on TCP connection, using different SID Lists; in the second one we have a UDP traffic flow test on the same scenarios.

Table 1. SRv6 ground truth | TCP flow - 60 sec. test

#SID in the Segment List	Throughput (Mbit/s)	CPU Tx Usage (%)	CPU Rx Usage (%)
No SRv6	94.14	0.4	8.44
1	95.44	0.4	9.12
3	95.54	0.4	8.38

The collected results allow us to find some performance ground truth on the testbed subject to no disturbance/attacks. Such data is presented in Table 1 as regards TCP based traffic and in Table 2 concerning UDP traffic simulating real-time traffic scenarios. The outcome for both bulk traffic (i.e. large TCP data transfer)

[3] https://iperf.fr/

Table 2. SRv6 ground truth | 100Mbit/s UDP flow - 60 sec. test

#SID in the SID List	Throughput (Mbit/s)	CPU Tx Usage (%)	CPU Rx Usage (%)	Jitter (ms)	Packet Loss (%)
No SRv6	95.3	13.58	5.16	0.0976	0.0482
1	95.76	14.4	4.84	0.096	0.042
3	95.76	13.96	5.94	0.0822	0.029

and potentially real time traffic using the basic UDP protocol is quite good. In fact, for the TCP tests, on average, the throughput is always close to the maximum obtainable on the network links and the insertion of a small number of SIDs does not appear to affect performance. Furthermore, the CPU occupancy is quite low for both the sender and the receiver, negliglible for the former, whereas the latter is not surpassing 10% in any condition. As regards UDP, the average throughput is also close to the maximum allowance for the link capacity, but in this case CPU usage appears higher, both for the receiver (topping at 6%) and especially for the sender (topping at 15%) and this is a surprising result we need to investigate further.

The simulated Transit Node Attack is represented in Figure 3 where we compromise a router through which IP packets flow: the router is not part of the SRv6 policy inserted by the SRv6 Ingress Node, so it is out of the management of the SRv6 Domain. As we have shown, even a commodity host, properly configured, can act as a MITM router. This compromised Transit Node succeeded in altering the SID List held in the SR6 Header: in our case, it substituted the second SID, thus modifying the routing path and redirecting traffic to another Malicious Transit Node before arriving to the Egress Node. It is significant to have succeeded in altering the SID list of in transit packets. This opens up novel possibilities, especially if each SRv6 Node that is part of the SID List represents a network function acting on traffic flows. Nevertheless, further experimental activity will be the objective of future work, involving increasingly complex header manipulations and network configurations.

6. Conclusion

This paper has reviewed Segment Routing and in particular the SRv6 architecture in IP networks. Some relevant security issues have been discussed, showing some preliminary attacks. The actual implementation of the testbed has been motivated and detailed, and first experimental results have been discussed. The take home message of present paper is that SRv6 technology weaknesses can actually be abused to cause issues. The stakeholders of our work, i.e. researchers, network admins and engineers can benefit from the increased knowledge and awareness of the specific network issues. Nevertheless, the results presented here are limited, and a further more thorough experimental campaign will be the objective of future work.

Acknowledgments

We would especially like to thank Simone Nafra for his invualuable contribution in the testbed setup and help with the experimental campaign.

References

[1] Partha Pratim Ray and Neeraj Kumar. SDN/NFV architectures for Edge-Cloud oriented IoT: A systematic review. *Computer Communications*, 169:129–153, 2021. ISSN 0140-3664.

[2] Maurantonio Caprolu, Roberto Di Pietro, Flavio Lombardi, and Simone Raponi. Edge computing perspectives: Architectures, technologies, and open security issues. In *2019 IEEE International Conference on Edge Computing (EDGE)*, pages 116–123, 2019.

[3] Jabril Abdelaziza, Mehdi Adda, and Hamid Mcheick. An architectural model for fog computing. *Journal of Ubiquitous Systems and Pervasive Networks*, 10(1):21–25, March 2018.

[4] Patricia A. Morreale and James M. Anderson. *Software Defined Networking: Design and Deployment*. CRC Press, Inc., USA, 2014. ISBN 1482238632.

[5] Mohammed Alabbad and Ridha Khedri. Dynamic segmentation, configuration, and governance of sdn. *Journal of Ubiquitous Systems and Pervasive Networks*, 16 (1):7–22, January 2022.

[6] Singh Vishal Krishna, Verma Saurabh, and Kumar Manish. Evaluation of privacy preserving in-network aggregation for different routing structures in wsns. *Journal of Ubiquitous Systems and Pervasive Networks*, 9(2):15–19, March 2017.

[7] Clarence Filsfils, Stefano Previdi, Les Ginsberg, Bruno Decraene, Stephane Litkowski, and Rob Shakir. Segment Routing Architecture. RFC 8402, July 2018.

[8] Christian Hopps. Analysis of an Equal-Cost Multi-Path Algorithm. RFC 2992, November 2000.

[9] JaeDeok Lim and YoungKi Kim. Protection Algorithm against security holes of IPv6 routing header. In *2006 8th International Conference Advanced Communication Technology*, volume 3, pages 2004–2007, 2006.

[10] Mario Smith and Norbert Kottapalli. In-Flight IPv6 Extension Header Insertion Considered Harmful, 2020.

[11] Clarence Filsfils, Darren Dukes, Stefano Previdi, John Leddy, Satoru Matsushima, and Daniel Voyer. IPv6 Segment Routing Header (SRH). RFC 8754, March 2020.

[12] David Lo Bascio and Flavio Lombardi. On srv6 security. *Procedia Computer Science*, 201:406–412, 2022. ISSN 1877-0509. 13th Intl Conf on Ambient Systems, Networks and Technologies (ANT).

[13] Gaëtan Leurent, Thomas Peyrin, and Lei Wang. New generic attacks against hash-based macs. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology -*

ASIACRYPT 2013, pages 1–20, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-42045-0.

[14] Pier Ventre, Stefano Salsano, Marco Polverini, Antonio Cianfrani, Ahmed Abdelsalam, Clarence Filsfils, Pablo Camarillo, and Francois Clad. Segment Routing: A Comprehensive Survey of Research Activities, Standardization Efforts, and Implementation Results. *IEEE Communications Surveys & Tutorials*, PP, 2020.

[15] Monali Mavani and Leena Ragha. Security implication and detection of threats due to manipulating ipv6 extension headers. In *2013 Annual IEEE India Conference (INDICON)*, pages 1–6, 2013.

[16] Monali Mavani and Leena Ragha. Covert channel in ipv6 destination option extension header. In *2014 International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA)*, pages 219–224, 2014.

[17] Choukri Djellali and Mehdi adda. An enhanced deep learning model to network attack detection, by using parameter tuning, hidden markov model and neural network. *Journal of Ubiquitous Systems and Pervasive Networks*, 15(01):35–41, March 2021.

[18] Johanna Ullrich, Katharina Krombholz, Heidelinde Hobel, Adrian Dabrowski, and Edgar Weippl. IPv6 Security: Attacks and Countermeasures in a Nutshell. In *8th USENIX Workshop on Offensive Technologies (WOOT 14)*, pages 1–20, San Diego, CA, August 2014. USENIX Association.

[19] David Lebrun. IPv6 segment routing, 2020.

[20] Suresh Krishnan, Elwyn B. Davies, and Pekka Savola. IPv6 Transition/Co-existence Security Considerations. RFC 4942, September 2007.

[21] Juan Abley Afilias, Pablo Savola, and Guillermo Neville-Neil. Deprecation of Type 0 Routing Headers in IPv6. RFC 5095, 2005.

[22] Éric Vyncke, Stefano Previdi, and David Lebrun. IPv6 Segment Routing Security Considerations. Internet-Draft draft-vyncke-6man-segment-routing-security-02, Internet Engineering Task Force, February 2015.

[23] Lubna Fayez Eliyan and Roberto Di Pietro. DoS and DDoS attacks in Software Defined Networks: A survey of existing solutions and research challenges. *Future Generation Computer Systems*, 122:149–171, 2021. ISSN 0167-739X.

[24] Bob Hinden and Dr. Steve E. Deering. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, December 1998.

[25] P. Biondi and A. Ebalard. IPv6 Routing Header Security. In *CanSecWest Security Conference 2007*, pages 1–61, 2007.

[26] Jen-Cheng Huang, Matteo Monchiero, Yoshio Turner, and Hsien-Hsin S. Lee. Ally: Os-transparent packet inspection using sequestered cores. In *2011 ACM/IEEE Seventh Symposium on Architectures for Networking and Communications Systems*, pages 1–11, 2011.

[27] Cheng Li, Zhenbin Li, Chongfeng Xie, Hui Tian, and Jianwei Mao. Security Considerations for SRv6 Networks. Internet-Draft draft-li-spring-srv6-security-consideration-07, Internet Engineering Task Force, October 2021.

[28] Robert Edgard Barton and Jerome Henry. Fatigue - based segment routing, Jul 2019. Patent H04L 29/06 (20060101); H04L 12/707 (20060101), Jul. 2019.

[29] Clarence Filsfils, Pablo Camarillo Garvia, and Francois Clad. Providing processing and network efficiencies in protecting internet protocol version 6 segment routing packets and functions using security segment identifiers, Dec U.S. Patent 11019075, May. 2021.

[30] Jiang Zhou, Hewu Li, Qian Wu, Zeqi Lai, and Jun Liu. SR-TPP: Extending IPv6 Segment Routing to enable Trusted and Private Network Paths. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6, 2020.

[31] Ryan Craven, Robert Beverly, and Mark Allman. Techniques for the detection of faulty packet header modifications, 2014-03-12.

[32] Isaac Woungang, Sanjay Kumar Dhurandher, Vincent Koo, and Issa Traore. Comparison of two security protocols for preventing packet dropping and message tampering attacks on aodv-based mobile ad hoc networks. In *2012 IEEE Globecom Workshops*, pages 1037–1041, 2012.

[33] Wan Nor Ashiqin Wan Ali, Abidah Hj Mat Taib, Naimah Mohd Hussin, and Jamal Othman. IPv6 attack scenarios testbed. In *2012 IEEE Symposium on Humanities, Science and Engineering Research*, pages 927–932, 2012.

[34] Flavio Lombardi and Roberto Di Pietro. *Virtualization and Cloud Security: Benefits, Caveats, and Future Developments*, pages 237–255. Springer International Publishing, Cham, 2014. ISBN 978-3-319-10530-7.

[35] Clarence Filsfils, Pablo Camarillo, John Leddy, Daniel Voyer, Satoru Matsushima, and Zhenbin Li. Segment Routing over IPv6 (SRv6) Network Programming. RFC 8986, February 2021.