Stream: Internet Engineering Task Force (IETF)
RFC: 9098
Category: Informational
Published: September 2021
ISSN: 2070-1721

Authors:

F. Gont	N. Hilliard	G. Doering	W. Kumari	G. Huston
SI6 Networks	*INEX*	*SpaceNet AG*	*Google*	*APNIC*

W. Liu
Huawei Technologies

RFC 9098
Operational Implications of IPv6 Packets with Extension Headers

Abstract

This document summarizes the operational implications of IPv6 extension headers specified in the IPv6 protocol specification (RFC 8200) and attempts to analyze reasons why packets with IPv6 extension headers are often dropped in the public Internet.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

IPv6 extension headers (EHs) allow for the extension of the IPv6 protocol and provide support for core functionality such as IPv6 fragmentation. However, common implementation limitations suggest that EHs present a challenge for IPv6 packet routing equipment and middleboxes, and evidence exists that IPv6 packets with EHs are intentionally dropped in the public Internet in some circumstances.

This document has the following goals:

- Raise awareness about the operational and security implications of IPv6 extension headers specified in [RFC8200] and present reasons why some networks resort to intentionally dropping packets containing IPv6 extension headers.
- Highlight areas where current IPv6 support by networking devices may be suboptimal, such that the aforementioned support is improved.
- Highlight operational issues associated with IPv6 extension headers, such that those issues are considered in IETF standardization efforts.

Section 4 of this document provides background information about the IPv6 packet structure and associated implications. Section 5 summarizes previous work that has been carried out in the area of IPv6 extension headers. Section 6 discusses packet-forwarding engine constraints in contemporary routers. Section 7 discusses why intermediate systems may need to access Layer 4 information to make a forwarding decision. Finally, Section 8 discusses operational implications of IPv6 EHs.

2. Terminology

This document uses the term "intermediate system" to describe both routers and middleboxes when there is no need to distinguish between the two and where the important issue is that the device being discussed forwards packets.

3. Disclaimer

This document analyzes the operational challenges represented by packets that employ IPv6 extension headers and documents some of the operational reasons why these packets are often dropped in the public Internet. This document is not a recommendation to drop such packets, but rather an analysis of why they are currently dropped.

4. Background Information

It is useful to compare the basic structure of IPv6 packets against that of IPv4 packets and analyze the implications of the two different packet structures.

IPv4 packets have a variable-length header size that allows for the use of IPv4 "options" -- optional information that may be of use to nodes processing IPv4 packets. The IPv4 header length is specified in the "Internet Header Length" (IHL) field of the mandatory IPv4 header and must be in the range of 20 octets (the minimum IPv4 header size) to 60 octets, accommodating at most 40 octets of options. The upper-layer protocol type is specified via the "Protocol" field of the mandatory IPv4 header.

```
                 Protocol, IHL
                      +--------+
                      |        |
                      |        v
                 +------//-----+------------------------+
                 |             |                        |
                 |    IPv4     |       Upper-Layer      |
                 |    Header   |       Protocol         |
                 |             |                        |
                 +-----//------+------------------------+

                 variable length
                 <------------->
```

Figure 1: IPv4 Packet Structure

IPv6 took a different approach to the IPv6 packet structure. Rather than employing a variable-length header as IPv4 does, IPv6 employs a packet structure similar to a linked list, where a mandatory fixed-length IPv6 header is followed by an arbitrary number of optional extension headers, with the upper-layer header being the last header in the IPv6 header chain. Each extension header typically specifies its length (unless it is implicit from the extension header type) and the "next header" (NH) type that follows in the IPv6 header chain.

```
     NH          NH, EH-length      NH, EH-length
      +-------+      +------+            +-------+
      |       |      |      |            |       |
      |       v      |      v            |       v
+-------------+-------------+-//-+---------------+--------------+
|             |             |    |               |              |
|    IPv6     |    Ext.     |    |     Ext.      |  Upper-Layer |
|    header   |    Header   |    |     Header    |  Protocol    |
|             |             |    |               |              |
+-------------+-------------+-//-+---------------+--------------+

 fixed length    variable number of EHs & length
<------------> <-------------------------------->
```

Figure 2: IPv6 Packet Structure

This packet structure has the following implications:

- [RFC8200] requires the entire IPv6 header chain to be contained in the first fragment of a packet, therefore limiting the IPv6 header chain to the size of the path MTU.
- Other than the path MTU constraints, there are no other limits to the number of IPv6 EHs that may be present in a packet. Therefore, there is no upper limit regarding how deep into the IPv6 packet the upper-layer protocol header may be found.
- The only way for a node to obtain the upper-layer protocol type or find the upper-layer protocol header is to parse and process the entire IPv6 header chain, in sequence, starting from the mandatory IPv6 header until the last header in the IPv6 header chain is found.

5. Previous Work on IPv6 Extension Headers

Some of the operational and security implications of IPv6 extension headers have been discussed in the IETF:

- [OPERATORS] discusses a rationale for which operators drop IPv6 fragments.
- [HEADERS] discusses possible issues arising from "long" IPv6 header chains.
- [PARSING] describes how inconsistencies in the way IPv6 packets with extension headers are parsed by different implementations could result in evasion of security controls and presents guidelines for parsing IPv6 extension headers, with the goal of providing a common and consistent parsing methodology for IPv6 implementations.
- [IPV6-EH] analyzes the security implications of IPv6 EHs, as well as the operational implications of dropping packets that employ IPv6 EHs and associated options.
- [RFC7113] discusses how some popular Router Advertisement Guard (RA-Guard) implementations are subject to evasion by means of IPv6 extension headers.
- [RFC8900] analyzes the fragility introduced by IP fragmentation.

A number of recent RFCs have discussed issues related to IPv6 extension headers and have specified updates to RFC 2460 [RFC2460] (an earlier version of the IPv6 standard). Many of these updates have now been incorporated into the current IPv6 core standard [RFC8200] or the IPv6 node requirements [RFC8504]. Namely,

- [RFC5095] discusses the security implications of Routing Header Type 0 (RHT0) and deprecates it.
- [RFC5722] analyzes the security implications of overlapping fragments and provides recommendations in this area.
- [RFC7045] clarifies how intermediate nodes should deal with IPv6 extension headers.
- [RFC7112] discusses the issues arising in a specific fragmentation case where the IPv6 header chain is fragmented into two or more fragments and formally forbids such fragmentation.
- [RFC6946] discusses a flawed (but common) processing of the so-called IPv6 "atomic fragments" and specifies improved processing of such packets.
- [RFC8021] deprecates the generation of IPv6 atomic fragments.
- [RFC8504] clarifies processing rules for packets with extension headers and also allows hosts to enforce limits on the number of options included in IPv6 EHs.
- [RFC7739] discusses the security implications of predictable fragment Identification values and provides recommendations for the generation of these values.
- [RFC6980] analyzes the security implications of employing IPv6 fragmentation with Neighbor Discovery for IPv6 and formally recommends against such usage.

Additionally, [RFC8200] has relaxed the requirement that "all nodes must examine and process the Hop-by-Hop Options header" from [RFC2460], by specifying that only nodes that have been explicitly configured to process the Hop-by-Hop Options header are required to do so.

A number of studies have measured the extent to which packets employing IPv6 extension headers are dropped in the public Internet:

- [PMTUD-Blackholes] and [Linkova-Gont-IEPG90] present some preliminary measurements regarding the extent to which packets containing IPv6 EHs are dropped in the public Internet.
- [RFC7872] presents more comprehensive results and documents the methodology used to obtain these results.
- [Huston-2017] and [Huston-2020] measure packet drops resulting from IPv6 fragmentation when communicating with DNS servers.

6. Packet-Forwarding Engine Constraints

Most contemporary carrier-grade routers use dedicated hardware, e.g., Application-Specific Integrated Circuits (ASICs) or Network Processing Units (NPUs), to determine how to forward packets across their internal fabrics (see [IEPG94-Scudder] and [APNIC-Scudder] for details). One common method of handling next-hop lookups is to send a small portion of the ingress packet to a lookup engine with specialized hardware, e.g., ternary content-addressable memory (TCAM) or reduced latency dynamic random-access memory (RLDRAM), to determine the packet's next hop. Technical constraints mean that there is a trade-off between the amount of data sent to the lookup engine and the overall packet-forwarding rate of the lookup engine. If more data is sent, the lookup engine can inspect further into the packet, but the overall packet-forwarding rate of the system will be reduced. If less data is sent, the overall packet-forwarding rate of the router will be increased, but the packet lookup engine may not be able to inspect far enough into a packet to determine how it should be handled.

> NOTE:
>
> For example, some contemporary high-end routers are known to inspect up to 192 bytes, while others are known to parse up to 384 bytes of header.

If a hardware-forwarding engine on a contemporary router cannot make a forwarding decision about a packet because critical information is not sent to the lookup engine, then the router will normally drop the packet. Section 7 discusses some of the reasons for which a contemporary router might need to access Layer 4 information to make a forwarding decision.

Historically, some packet-forwarding engines punted packets of this kind to the control plane for more in-depth analysis, but this is unfeasible on most contemporary router architectures as a result of the vast difference between the hardware-based forwarding capacity of the router and the processing capacity of the control plane and the size of the management link that connects the control plane to the forwarding plane. Other platforms may have a separate software-based forwarding plane that is distinct both from the hardware-based forwarding plane and the control plane. However, the limited CPU resources of this software-based forwarding plane, as well as the limited bandwidth of the associated link, results in similar throughput constraints.

If an IPv6 header chain is sufficiently long such that it exceeds the packet lookup capacity of the router, the router might be unable to determine how the packet should be handled and thus could resort to dropping the packet.

6.1. Recirculation

Although type-length-value (TLV) chains are amenable to iterative processing on architectures that have packet lookup engines with deep inspection capabilities, some packet-forwarding engines manage IPv6 header chains using recirculation. This approach processes extension headers one at a time: when processing on one extension header is completed, the packet is looped back through the processing engine again. This recirculation process continues repeatedly until there are no more extension headers left to be processed.

Recirculation is typically used on packet-forwarding engines with limited lookup capability, because it allows arbitrarily long header chains to be processed without the complexity and cost associated with packet-forwarding engines, which have deep lookup capabilities. However, recirculation can impact the forwarding capacity of hardware, as each packet will pass through the processing engine multiple times. Depending on configuration, the type of packets being processed, and the hardware capabilities of the packet-forwarding engine, the data-plane throughput performance on the router might be negatively affected.

7. Requirement to Process Layer 3 / Layer 4 Information in Intermediate Systems

The following subsections discuss some of the reasons for which intermediate systems may need to process Layer 3 / Layer 4 information to make a forwarding decision.

7.1. ECMP and Hash-Based Load Sharing

In the case of Equal Cost Multipath (ECMP) load sharing, the intermediate system needs to make a decision regarding which of its interfaces to use to forward a given packet. Since round-robin usage of the links is usually avoided to prevent packet reordering, forwarding engines need to use a mechanism that will consistently forward the same data streams down the same forwarding paths. Most forwarding engines achieve this by calculating a simple hash using an n-tuple gleaned from a combination of Layer 2 through to Layer 4 protocol header information. This n-tuple will typically use the src/dst Media Access Control (MAC) addresses, src/dst IP addresses, and, if possible, further Layer 4 src/dst port information.

In the IPv6 world, flows are expected to be identified by means of the IPv6 "Flow Label" [RFC6437]. Thus, ECMP and hash-based load sharing should be possible without the need to process the entire IPv6 header chain to obtain upper-layer information to identify flows. [RFC7098] discusses how the IPv6 Flow Label can be used to enhance Layer 3/4 load distribution and balancing for large server farms.

Historically, many IPv6 implementations failed to set the Flow Label, and hash-based ECMP/load-sharing devices also did not employ the Flow Label for performing their task. While support of [RFC6437] is currently widespread for current versions of all popular host implementations, there is still only marginal usage of the IPv6 Flow Label for ECMP and load balancing [Almeida-2020]. A contributing factor could be the issues that have been found in host implementations and middleboxes [Jaeggli-2018].

Clearly, widespread support of [RFC6437] would relieve intermediate systems from having to process the entire IPv6 header chain, making Flow Label-based ECMP and load sharing [RFC6438] feasible.

If an intermediate system cannot determine consistent n-tuples for calculating flow hashes, data streams are more likely to end up being distributed unequally across ECMP and load-shared links. This may lead to packet drops or reduced performance.

7.2. Enforcing Infrastructure ACLs

Infrastructure Access Control Lists (iACLs) drop unwanted packets destined to a network's infrastructure. Typically, iACLs are deployed because external direct access to a network's infrastructure addresses is operationally unnecessary and can be used for attacks of different sorts against router control planes. To this end, traffic usually needs to be differentiated on the basis of Layer 3 or Layer 4 criteria to achieve a useful balance of protection and functionality. For example, an infrastructure may be configured with the following policy:

- Permit some amount of ICMP echo (ping) traffic towards a router's addresses for troubleshooting.
- Permit BGP sessions on the shared network of an exchange point (potentially differentiating between the amount of packets/second permitted for established sessions and for connection establishment), but do not permit other traffic from the same peer IP addresses.

If a forwarding router cannot determine consistent n-tuples for calculating flow hashes, data streams are more likely to end up being distributed unequally across ECMP and load-shared links. This may lead to packet drops or reduced performance.

If a network cannot deploy infrastructure ACLs, then the security of the network may be compromised as a result of the increased attack surface.

7.3. DDoS Management and Customer Requests for Filtering

The case of customer Distributed Denial-of-Service (DDoS) protection and edge-to-core customer protection filters is similar in nature to the iACL protection. Similar to iACL protection, Layer 4 ACLs generally need to be applied as close to the edge of the network as possible, even though the intent is usually to protect the customer edge rather than the provider core. Application of Layer 4 DDoS protection to a network edge is often automated using BGP Flowspec [RFC8955] [RFC8956].

For example, a website that normally only handles traffic on TCP ports 80 and 443 could be subject to a volumetric DDoS attack using NTP and DNS packets with a randomized source IP address, thereby rendering source-based remote triggered black hole [RFC5635] mechanisms useless. In this situation, ACLs that provide DDoS protection could be configured to block all UDP traffic at the network edge without impairing the web server functionality in any way. Thus, being able to block arbitrary protocols at the network edge can avoid DDoS-related problems both in the provider network and on the customer edge link.

7.4. Network Intrusion Detection and Prevention

Network Intrusion Detection Systems (NIDS) examine network traffic and try to identify traffic patterns that can be correlated to network-based attacks. These systems generally attempt to inspect application-layer traffic (if possible) but, at the bare minimum, inspect Layer 4 flows. When attack activity is inferred, the operator is notified of the potential intrusion attempt.

Network Intrusion Prevention Systems (IPS) operate similarly to NIDSs, but they can also prevent intrusions by reacting to detected attack attempts by e.g., triggering packet filtering policies at firewalls and other devices.

Use of extension headers can be problematic for NIDS/IPS, since:

- Extension headers increase the complexity of resulting traffic and the associated work and system requirements to process it.
- Use of unknown extension headers can prevent a NIDS or IPS from processing Layer 4 information.
- Use of IPv6 fragmentation requires a stateful fragment-reassembly operation, even for decoy traffic employing forged source addresses (see, e.g., [nmap]).

As a result, in order to increase the efficiency or effectiveness of these systems, packets employing IPv6 extension headers are often dropped at the network ingress point(s) of networks that deploy these systems.

7.5. Firewalling

Firewalls enforce security policies by means of packet filtering. These systems usually inspect Layer 3 and Layer 4 traffic but can often also examine application-layer traffic flows.

As with a NIDS or IPS (Section 7.4), use of IPv6 extension headers can represent a challenge to network firewalls, since:

- Extension headers increase the complexity of resulting traffic and the associated work and system requirements to process it, as outlined in [Zack-FW-Benchmark].
- Use of unknown extension headers can prevent firewalls from processing Layer 4 information.
- Use of IPv6 fragmentation requires a stateful fragment-reassembly operation, even for decoy traffic employing forged source addresses (see, e.g., [nmap]).

Additionally, a common firewall filtering policy is the so-called "default deny", where all traffic is blocked (by default), and only expected traffic is added to an "allow/accept list".

As a result, packets employing IPv6 extension headers are often dropped by network firewalls, either because of the challenges represented by extension headers or because the use of IPv6 extension headers has not been explicitly allowed.

Note that although the data presented in [Zack-FW-Benchmark] was several years old at the time of publication of this document, many contemporary firewalls use comparable hardware and software architectures; consequently, the conclusions of this benchmark are still relevant, despite its age.

8. Operational and Security Implications

8.1. Inability to Find Layer 4 Information

As discussed in Section 7, intermediate systems that need to find the Layer 4 header must process the entire IPv6 header chain. When such devices are unable to obtain the required information, the forwarding device has the option to drop the packet unconditionally, forward the packet unconditionally, or process the packet outside the normal forwarding path. Forwarding packets unconditionally will usually allow for the circumvention of security controls (see, e.g., Section 7.5), while processing packets outside of the normal forwarding path will usually open the door to Denial-of-Service (DoS) attacks (see, e.g., Section 6). Thus, in these scenarios, devices often simply resort to dropping such packets unconditionally.

8.2. Route-Processor Protection

Most contemporary carrier-grade routers have a fast hardware-assisted forwarding plane and a loosely coupled control plane, connected together with a link that has much less capacity than the forwarding plane could handle. Traffic differentiation cannot be performed by the control plane because this would overload the internal link connecting the forwarding plane to the control plane.

The Hop-by-Hop Options header has been particularly challenging since, in most circumstances, the corresponding packet is punted to the control plane for processing. As a result, many operators drop IPv6 packets containing this extension header [RFC7872]. [RFC6192] provides advice regarding protection of a router's control plane.

8.3. Inability to Perform Fine-Grained Filtering

Some intermediate systems do not have support for fine-grained filtering of IPv6 extension headers. For example, an operator that wishes to drop packets containing RHT0 may only be able to filter on the extension header type (Routing Header). This could result in an operator enforcing a coarser filtering policy (e.g., "drop all packets containing a Routing Header" vs. "only drop packets that contain a Routing Header Type 0").

8.4. Security Concerns Associated with IPv6 Extension Headers

The security implications of IPv6 extension headers generally fall into one or more of these categories:

- Evasion of security controls
- DoS due to processing requirements
- DoS due to implementation errors
- Issues specific to the extension header type

Unlike IPv4 packets where the upper-layer protocol can be trivially found by means of the IHL field of the IPv4 header, the structure of IPv6 packets is more flexible and complex. This can represent a challenge for devices that need to find this information, since locating upper-layer protocol information requires that all IPv6 extension headers be examined. In turn, this presents implementation difficulties, since some packet-filtering mechanisms that require upper-layer information (even if just the upper-layer protocol type) can be trivially circumvented by inserting IPv6 extension headers between the main IPv6 header and the upper-layer protocol header. [RFC7113] describes this issue for the RA-Guard case, but the same techniques could be employed to circumvent other IPv6 firewall and packet-filtering mechanisms. Additionally, implementation inconsistencies in packet-forwarding engines can result in evasion of security controls [PARSING] [Atlasis2014] [BH-EU-2014].

Sometimes, packets with IPv6 extension headers can impact throughput performance on intermediate systems. Unless appropriate mitigations are put in place (e.g., packet dropping and/or rate limiting), an attacker could simply send a large amount of IPv6 traffic employing IPv6 extension headers with the purpose of performing a DoS attack (see Sections 6.1 and 8 for further details). The extent to which performance is affected on these devices is implementation dependent.

> NOTE:
>
> In the most trivial case, a packet that includes a Hop-by-Hop Options header might go through the slow forwarding path, to be processed by the router's CPU. Alternatively, a router configured to enforce an ACL based on upper-layer information (e.g., upper-layer protocol type or TCP Destination Port) may need to process the entire IPv6 header chain in order to find the required information, thereby causing the packet to be processed in the slow path [Cisco-EH-Cons]. We note that, for obvious reasons, the aforementioned performance issues can affect devices such as firewalls, NIDSs, etc. [Zack-FW-Benchmark].

IPv6 implementations, like all other software, tend to mature with time and wide-scale deployment. While the IPv6 protocol itself has existed for over 20 years, serious bugs related to IPv6 extension header processing continue to be discovered (see, e.g., [Cisco-Frag], [Microsoft-SA], and [FreeBSD-SA]). Because there is currently little operational reliance on IPv6 extension headers, the corresponding code paths are rarely exercised, and there is the potential for bugs that still remain to be discovered in some implementations.

The IPv6 Fragment Header is employed for the fragmentation and reassembly of IPv6 packets. While many of the security implications of the fragmentation/reassembly mechanism are known from the IPv4 world, several related issues have crept into IPv6 implementations. These range from DoS attacks to information leakages, as discussed in [RFC7739], [Bonica-NANOG58], and [Atlasis2012].

9. IANA Considerations

This document has no IANA actions.

10. Security Considerations

The security implications of IPv6 extension headers are discussed in Section 8.4. This document does not introduce any new security issues.

11. References

11.1. Normative References

[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, <https://www.rfc-editor.org/info/rfc5095>.

[RFC5722] Krishnan, S., "Handling of Overlapping IPv6 Fragments", RFC 5722, DOI 10.17487/RFC5722, December 2009, <https://www.rfc-editor.org/info/rfc5722>.

[RFC6946] Gont, F., "Processing of IPv6 "Atomic" Fragments", RFC 6946, DOI 10.17487/RFC6946, May 2013, <https://www.rfc-editor.org/info/rfc6946>.

[RFC6980] Gont, F., "Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery", RFC 6980, DOI 10.17487/RFC6980, August 2013, <https://www.rfc-editor.org/info/rfc6980>.

[RFC7112] Gont, F., Manral, V., and R. Bonica, "Implications of Oversized IPv6 Header Chains", RFC 7112, DOI 10.17487/RFC7112, January 2014, <https://www.rfc-editor.org/info/rfc7112>.

[RFC8021] Gont, F., Liu, W., and T. Anderson, "Generation of IPv6 Atomic Fragments Considered Harmful", RFC 8021, DOI 10.17487/RFC8021, January 2017, <https://www.rfc-editor.org/info/rfc8021>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8504] Chown, T., Loughney, J., and T. Winters, "IPv6 Node Requirements", BCP 220, RFC 8504, DOI 10.17487/RFC8504, January 2019, <https://www.rfc-editor.org/info/rfc8504>.

11.2. Informative References

[Almeida-2020] Almeida, R., Cunha, I., Teixeira, R., Veitch, D., and C. Diot, "Classification of Load Balancing in the Internet", IEEE INFOCOM 2020, DOI 10.1109/INFOCOM41043.2020.9155387, July 2020, <https://homepages.dcc.ufmg.br/~cunha/papers/almeida20infocom-mca.pdf>.

[APNIC-Scudder] Scudder, J., "Modern router architecture and IPv6", APNIC Blog, June 2020, <https://blog.apnic.net/2020/06/04/modern-router-architecture-and-ipv6/>.

[Atlasis2012] Atlasis, A., "Attacking IPv6 Implementation Using Fragmentation", Black Hat Europe 2012, March 2012, <https://void.gr/kargig/ipv6/bh-eu-12-Atlasis-Attacking_IPv6-Slides.pdf>.

[Atlasis2014] Atlasis, A., "A Novel Way of Abusing IPv6 Extension Headers to Evade IPv6 Security Devices", May 2014, <http://www.insinuator.net/2014/05/a-novel-way-of-abusing-ipv6-extension-headers-to-evade-ipv6-security-devices/>.

[BH-EU-2014] Atlasis, A., Rey, E., and R. Schaefer, "Evasion of High-End IDPS Devices at the IPv6 Era", Black Hat Europe 2014, 2014, <https://www.ernw.de/download/eu-14-Atlasis-Rey-Schaefer-briefings-Evasion-of-HighEnd-IPS-Devices-wp.pdf>.

[Bonica-NANOG58] Bonica, R., "IPv6 Fragmentation: The Case For Deprecation", NANOG 58, June 2013, <https://www.nanog.org/sites/default/files/mon.general.fragmentation.bonica.pdf>.

[Cisco-EH-Cons] Cisco, "IPv6 Extension Headers Review and Considerations", October 2006, <http://www.cisco.com/en/US/technologies/tk648/tk872/technologies_white_paper0900aecd8054d37d.pdf>.

[Cisco-Frag] Cisco, "Cisco IOS XR Software Crafted IPv6 Packet Denial of Service Vulnerability", June 2015, <http://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20150611-iosxr>.

[FreeBSD-SA] The FreeBSD Project, "IPv6 Hop-by-Hop options use-after-free bug", September 2020, <https://www.freebsd.org/security/advisories/FreeBSD-SA-20:24.ipv6.asc>.

[HEADERS] Kumari, W., Jaeggli, J., Bonica, R. P., and J. Linkova, "Operational Issues Associated With Long IPv6 Header Chains", Work in Progress, Internet-Draft, draft-wkumari-long-headers-03, 16 June 2015, <https://datatracker.ietf.org/doc/html/draft-wkumari-long-headers-03>.

[Huston-2017] Huston, G., "Dealing with IPv6 fragmentation in the DNS", APNIC Blog, August 2017, <https://blog.apnic.net/2017/08/22/dealing-ipv6-fragmentation-dns/>.

[Huston-2020] Huston, G., "Measurement of IPv6 Extension Header Support", NPS/CAIDA 2020 Virtual IPv6 Workshop, June 2020, <https://www.cmand.org/workshops/202006-v6/slides/2020-06-16-xtn-hdrs.pdf>.

[IEPG94-Scudder] Petersen, B. and J. Scudder, "Modern Router Architecture for Protocol Designers", IEPG 94, November 2015, <http://www.iepg.org/2015-11-01-ietf94/IEPG-RouterArchitecture-jgs.pdf>.

[IPV6-EH] Gont, F. and W. Liu, "Recommendations on the Filtering of IPv6 Packets Containing IPv6 Extension Headers at Transit Routers", Work in Progress, Internet-Draft, draft-ietf-opsec-ipv6-eh-filtering-08, 3 June 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-opsec-ipv6-eh-filtering-08>.

[Jaeggli-2018] Jaeggli, J., "IPv6 flow label: misuse in hashing", APNIC Blog, January 2018, <https://blog.apnic.net/2018/01/11/ipv6-flow-label-misuse-hashing/>.

[Linkova-Gont-IEPG90] Linkova, J. and F. Gont, "IPv6 Extension Headers in the Real World v2.0", IEPG 90, July 2014, <http://www.iepg.org/2014-07-20-ietf90/iepg-ietf90-ipv6-ehs-in-the-real-world-v2.0.pdf>.

[Microsoft-SA] Microsoft, "Windows TCP/IP Remote Code Execution Vulnerability", CVE-2021-24094, February 2021, <https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-24094>.

[nmap] Lyon, G., "Firewall/IDS Evasion and Spoofing", Chapter 15. Nmap Reference Guide, <https://nmap.org/book/man-bypass-firewalls-ids.html>.

[OPERATORS] Jaeggli, J., Colitti, L., Kumari, W., Vyncke, E., Kaeo, M., and T. Taylor, Ed., "Why Operators Filter Fragments and What It Implies", Work in Progress, Internet-Draft, draft-taylor-v6ops-fragdrop-02, 3 December 2013, <https://datatracker.ietf.org/doc/html/draft-taylor-v6ops-fragdrop-02>.

[PARSING] Kampanakis, P., "Implementation Guidelines for Parsing IPv6 Extension Headers", Work in Progress, Internet-Draft, draft-kampanakis-6man-ipv6-eh-parsing-01, 5 August 2014, <https://datatracker.ietf.org/doc/html/draft-kampanakis-6man-ipv6-eh-parsing-01>.

[PMTUD-Blackholes] De Boer, M. and J. Bosma, "Discovering Path MTU black holes on the Internet using RIPE Atlas", University of Amsterdam, MSc. Systems & Network Engineering, July 2012, <http://www.nlnetlabs.nl/downloads/publications/pmtu-black-holes-msc-thesis.pdf>.

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

[RFC5635] Kumari, W. and D. McPherson, "Remote Triggered Black Hole Filtering with Unicast Reverse Path Forwarding (uRPF)", RFC 5635, DOI 10.17487/RFC5635, August 2009, <https://www.rfc-editor.org/info/rfc5635>.

[RFC6192] Dugal, D., Pignataro, C., and R. Dunn, "Protecting the Router Control Plane", RFC 6192, DOI 10.17487/RFC6192, March 2011, <https://www.rfc-editor.org/info/rfc6192>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/info/rfc6437>.

[RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <https://www.rfc-editor.org/info/rfc6438>.

[RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <https://www.rfc-editor.org/info/rfc7045>.

[RFC7098] Carpenter, B., Jiang, S., and W. Tarreau, "Using the IPv6 Flow Label for Load Balancing in Server Farms", RFC 7098, DOI 10.17487/RFC7098, January 2014, <https://www.rfc-editor.org/info/rfc7098>.

[RFC7113] Gont, F., "Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard)", RFC 7113, DOI 10.17487/RFC7113, February 2014, <https://www.rfc-editor.org/info/rfc7113>.

[RFC7739] Gont, F., "Security Implications of Predictable Fragment Identification Values", RFC 7739, DOI 10.17487/RFC7739, February 2016, <https://www.rfc-editor.org/info/rfc7739>.

[RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <https://www.rfc-editor.org/info/rfc7872>.

[RFC8900] Bonica, R., Baker, F., Huston, G., Hinden, R., Troan, O., and F. Gont, "IP Fragmentation Considered Fragile", BCP 230, RFC 8900, DOI 10.17487/RFC8900, September 2020, <https://www.rfc-editor.org/info/rfc8900>.

[RFC8955] Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", RFC 8955, DOI 10.17487/RFC8955, December 2020, <https://www.rfc-editor.org/info/rfc8955>.

[RFC8956] Loibl, C., Ed., Raszuk, R., Ed., and S. Hares, Ed., "Dissemination of Flow Specification Rules for IPv6", RFC 8956, DOI 10.17487/RFC8956, December 2020, <https://www.rfc-editor.org/info/rfc8956>.

[Zack-FW-Benchmark] Zack, E., "Firewall Security Assessment and Benchmarking IPv6 Firewall Load Tests", IPv6 Hackers Meeting #1, June 2013, <https://www.ipv6hackers.org/files/meetings/ipv6-hackers-1/zack-ipv6hackers1-firewall-security-assessment-and-benchmarking.pdf>.

Acknowledgements

The authors would like to thank (in alphabetical order) Mikael Abrahamsson, Fred Baker, Dale W. Carder, Brian Carpenter, Tim Chown, Owen DeLong, Gorry Fairhurst, Guillermo Gont, Tom Herbert, Lee Howard, Tom Petch, Sander Steffann, Eduard Vasilenko, Éric Vyncke, Rob Wilton, Jingrong Xie, and Andrew Yourtchenko for providing valuable comments on earlier draft versions of this document.

Fernando Gont would like to thank Jan Zorz / Go6 Lab <https://go6lab.si/>, Jared Mauch, and Sander Steffann <https://steffann.nl/> for providing access to systems and networks that were employed to perform experiments and measurements involving packets with IPv6 extension headers.

Authors' Addresses

Fernando Gont
SI6 Networks
Segurola y Habana 4310, 7mo Piso
Villa Devoto
Ciudad Autonoma de Buenos Aires
Argentina
Email: fgont@si6networks.com
URI: https://www.si6networks.com

Nick Hilliard
INEX
4027 Kingswood Road
Dublin
24
Ireland
Email: nick@inex.ie

Gert Doering
SpaceNet AG
Joseph-Dollinger-Bogen 14
D-80807 Muenchen
Germany
Email: gert@space.net

Warren Kumari
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
United States of America
Email: warren@kumari.net

Geoff Huston
Email: gih@apnic.net
URI: https://www.apnic.net

Will (Shucheng) Liu
Huawei Technologies
Bantian, Longgang District
Shenzhen
518129
China
Email: liushucheng@huawei.com