Workgroup: IPv6 Operations
Internet-Draft:
draft-elkins-v6ops-eh-deepdive-fw-01
Published: 21 October 2022
Intended Status: Best Current Practice
Expires: 24 April 2023
Authors: N. Elkins M. Ackermann
INT Council INT Council
D. Dhody
India Internet Engineering Society

Deep Dive into IPv6 Extension Header Testing

Abstract

IPv6 Extension Header testing is a complex area. Studies have shown varying results. This document proposes a methodology for isolating the location and reasons for IPv6 Extension Headers blockage. This document outlines the basic framework and set of documents which will follow.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the IPv6 Operations Working Group mailing list (v6ops@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/v6ops/.

Source for this draft and an issue tracker can be found at https://github.com/dhruvdhody/draft-elkins-v6ops-eh-deepdive-fw.

Status of This Memo

Table of Contents

1. Introduction

1.1. Problem Description

IPv6 [RFC8200] Extension Headers (EHs) may be blocked at various points yet show the same symptom. That is, if an EH is blocked at a router, it will appear to the client or measurement technique that the packet is dropped. If an EH is blocked at a load balancer or CDN, the client will see the same symptom -- the packet is dropped.

This is a problem because when the same symptom can be the due to multiple factors, incorrect conclusions may be drawn from the results. That is, if, for example, loss of the sense of taste of

sweetness can be due either to a brain tumor or a minor neurological problem, then if someone has a tendency towards hypochondria, they may incorrectly bemoan their impending death from a brain tumor when it may really be only a minor issue.

This is the same for EH testing. If a packet is seen to be dropped in measurement, it may only be that there is a bug in the load balancer code (for example) and not that EH transmission does not work writ large.

This document proposes a framework to isolate the problem of where the packet is being dropped. This is, however, more easily said than done. There are many components and devices in a network. Each may require a different isolation technique.

We propose a methdology which starts with the simplest topology and grows towards more and more complex, real-world networks. For example, in today's networks, if one attempts to access a well-known site such as Facebook.com, one is likely to access a cache server managed by a Content Delivery Network (CDN) rather than the origin server managed by Facebook. It is important to isolate the testing so that we can determine the exact component which may be blocking the EH and why.

We additionally discuss the potential issues with the type of data sent as well as how the way the data is sent for testing may impact test results.

1.2. Fundamental Premises

Yet, there are some overriding principles.

The blockage may be:

- *in the source network
- *in the destination network
- *in a transit network

Then:

- *it may be blocked intentionally
- *it may be blocked unintentionally

Intentional blocks are easier to assess. The block is an administrator or network policy. Of course, the policy may come about as a result of a misunderstanding of the function of EHs or

lack of effective guidance but nonetheless, this is an understandable decision.

The more troubling causes of EH blocks include:

- *bugs
- *default configurations or policies by the vendor
- *lack of support of EH or IPv6 itself

As far as the blocking component itself (whether intentional or unintentional):

- *may be a router
- *may be a firewall
- *may be a load balancer
- *may be a proxy
- *may be a Host OS
- *may be a DNS or other configuration setting

Note: A DNS or other configuration setting may block IPv6 or prefer IPv4. If not tested correctly, lack of support for IPv6 itself may incorrectly appear as lack of support for EHs.

Finally, the blockage or source may differ based on the type of EH. Some EHs are limited to an administrative domain, some EHs are processed at every hop, still others are from the source and destination. Problem isolation techniques will differ for the various classes of EH.

Our final document in this series will be a BCP to indicate which EH should be allowed, blocked, encrypted, or authenticated and at which component or platform.

Note: [RFC9288] "Recommendations on the Filtering of IPv6 Packets Containing IPv6 Extension Headers at Transit Routers" focuses on the IPv6 EH handling at transit routers only. Our approach is to produce a final BCP with various recommendations across network segments, once the nature of the problems and techniques for isolation are well known.

1.3. Diagnostic Methodology Overview

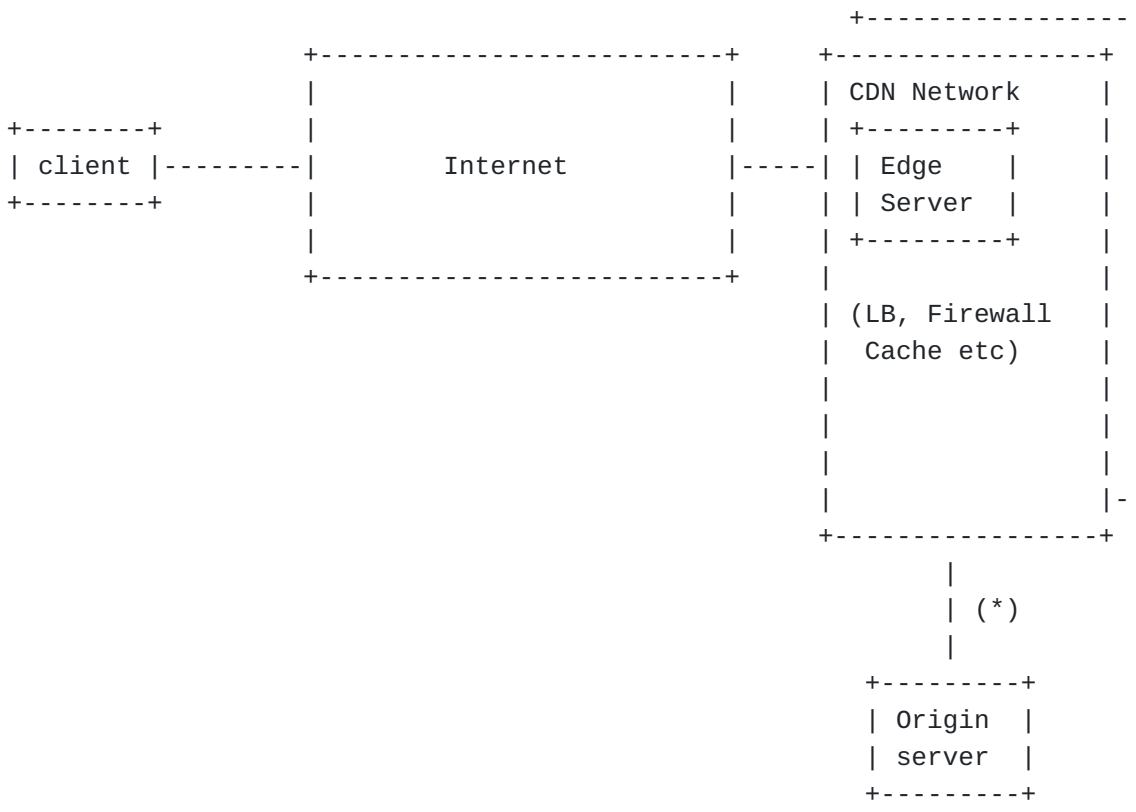
The diagnostic methodology to follow depends on what is being tested. For example, the problem isolation for a CDN would be different than the problem isolation for an internal network.

This framework proposes the following set of documents:

- *EH problem isolation for owned client / server
- *EH problem isolation in a network using a CDN
- *EH problem isolation in a network using a cloud provider
- *EH problem isolation for routers
- *EH problem isolation for load balancers
- *EH problem isolation for proxies
- *EH problem isolation for host OSs
- *EH problem isolation for transit networks
- *EH problem isolation for ISPs (multiple components / networks)
- *BCP for EH Permissions, Encryption and Authentication

Note that the server can be owned and operated by the administrator themselves (on-prem) Figure 1, or they could hosted behind a CDN Figure 2, or hosted by the cloud provider Figure 3.

Figure 1: Owned client / server



(*) - can be over the internet

Figure 2: Server behind CDN

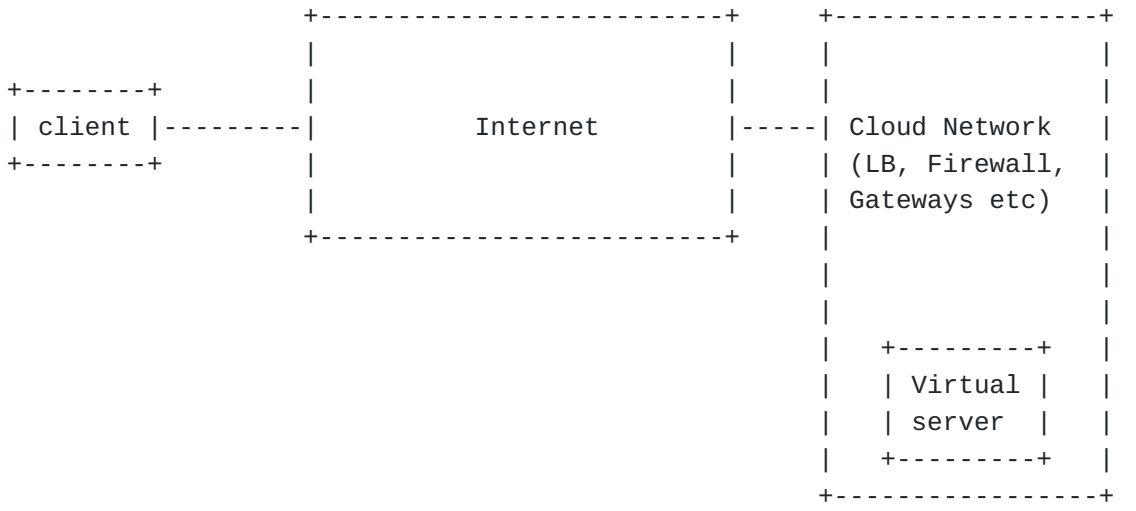


Figure 3: Cloud-hosted server

2. EH Enabled Server / Client / Router

The first step for all testing is to have a test server, client and / or router enabled to send EHs. There are two basic options to do this, each with its own set of advantages and drawbacks.

The options fall broadly into two categories:

* Modifications to the OS / interface / socket / driver to send EHs with application data

* Using a package which crafts EHs (multiple exist)

If crafting packets, then the question arises of which packets to craft. This will be discussed in Section 2.2.

For either methodology, the rate of sending may influence results. This will be discussed in Section 3.

2.1. Modifications to send EHs with application data

Possibly the best way to isolate problems with EHs may be to have a server, client or router which has modifications to the OS, interface, driver or other to send real EHs with real application data. This carries the initial cost of creating such modifications since, unfortunately, the current set of host OSs do not natively support this.

The reason we find this to be the best method is because the problem of what packet to craft is a field which is rife with land mines. See Section 2.2 for a further discussion of the exact nature of the aforesaid land mines.

2.2. Crafting packets with EH headers

A number of packages exist which can craft a packet with an EH header. The more interesting and fraught problem may be which exact packet to craft and then which EH to craft.

Let's discuss first the options for the type of packet to craft. These include:

* A TCP packet

* A UDP packet

* An ICMPv6 packet

* A QUIC packet

If crafting a TCP packet, then it is likely that some middlebox will drop a TCP packet which does not have the appropriate ACK and SEQUENCE numbers. One may get around this problem by sending a packet with the SYN flag on and directed to some well-known port such as 443 or 80. But, if many firewalls and other devices such as IDS / IPS, and even some OSs have SYN Flood protection. So, if more

than a certain number of these packets, say 10, in some short interval are sent, then they are likely to be dropped for reasons other than blockage of EH headers.

One may choose to use UDP, QUIC, or ICMPv6 to attempt to bypass the complexities of TCP. Some enterprise networks are likely to drop UDP and / or QUIC. Testing should be done without EH to make sure that such packets do indeed pass.

Certainly, if you have access to all the middleboxes in your domain, you may be able to bypass or stop SYN Flood, UDP, ICMPv6 or other transport layer blocks at all the middleboxes. But, it may be a more difficult effort than might be imagined.

You may wish to test in a lab environment first to validate your approach.

2.3. How to Add EH

There are two ways to add an EH to a packet:

- *use a "real" EH
- *craft an EH

The best method may be to use an EH which is actually processed at the client, server, and any associated router. This leaves us with a sparse set to choose from. The problem with crafting an EH is that if the data is not "real", one may not be able to fault a network component for blocking it.

Having said that, let's discuss crafting an EH. An EH may be crafted by simply using multiple PADN options. One should be careful not to use too many PADNs because then this type of header may be dropped by a middlebox or OS as being a flawed packet. This is likely to distort the results.

An EH with all 0's or other data patterns that could be perceived as not "real" may also be dropped by middleboxes which are trying to be helpful.

It is also possible that the length of the EH may have some effect on what EHs get dropped and where.

You may wish to try this in a lab environment first. If the test suceeds, then you may test on your network.

2.4. Which EH to Use

Next, there is a consideration that the type of EH, for example, Destination Options, Hop-by-hop, etc. could get processed differently and may be dropped at different frequencies, by different devices. You may wish to test one at a time and note the results.

You may wish to try this in a lab environment first. If the test suceeds, then you may test on your network.

3. Rate of sending and sampling

Whether you have chosen to send real application data or to craft packets, the rate of sending and sampling may create a false indication of blockage. That is, if you send a great deal of data at frequent intervals, some middlebox in the network is likely to see this traffic as a Denial of Service (DoS) attack and block it. This is more likely if a great many crafted packets are sent but even with real application data such as FTP or HTTP, overly aggressive sending is likely to be counterproductive.

You may wish to send only a few packets or one or two CURL or FTPs at any one test.

4. Security Considerations

This document has no security considerations.

5. Privacy Considerations

This document has no privacy considerations.

6. IANA Considerations

This document has no IANA actions.

7. References

7.1. Normative References

[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/rfc/rfc8200>.

7.2. Informative References

[RFC9288] Gont, F. and W. Liu, "Recommendations on the Filtering of IPv6 Packets Containing IPv6 Extension Headers at Transit

Routers", RFC 9288, DOI 10.17487/RFC9288, August 2022,
<https://www.rfc-editor.org/rfc/rfc9288>.

Acknowledgments

TODO acknowledge.

Contributors

TODO contributors.

Authors' Addresses

Nalini Elkins
Industry Network Technology Council
United States of America

Phone: +1 831 234 4232
Email: nalini.elkins@insidethestack.com

Michael Ackermann
Industry Network Technology Council
United States of America

Phone: +1 248 703 3600
Email: mackermann@bcbsm.com
URI: https://www.bcbsm.com

Dhruv Dhody
India Internet Engineering Society
India

Email: dhruv.ietf@gmail.com