Operational Implications of IPv6 Packets with Extension Headers

Fernando Gont

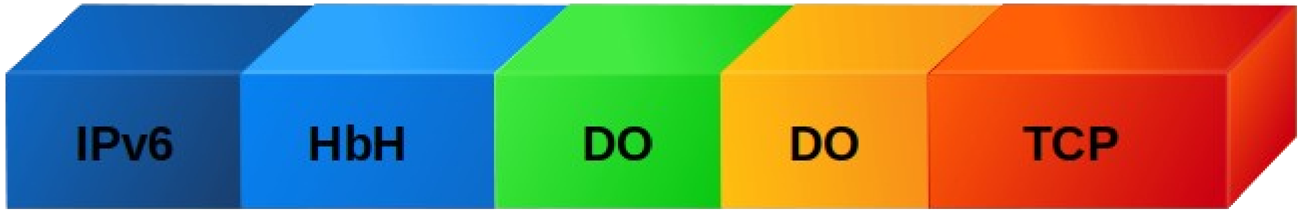
edgeuno.com

NANOG 83 Minneapolis
November 1-3, 2021

Introduction

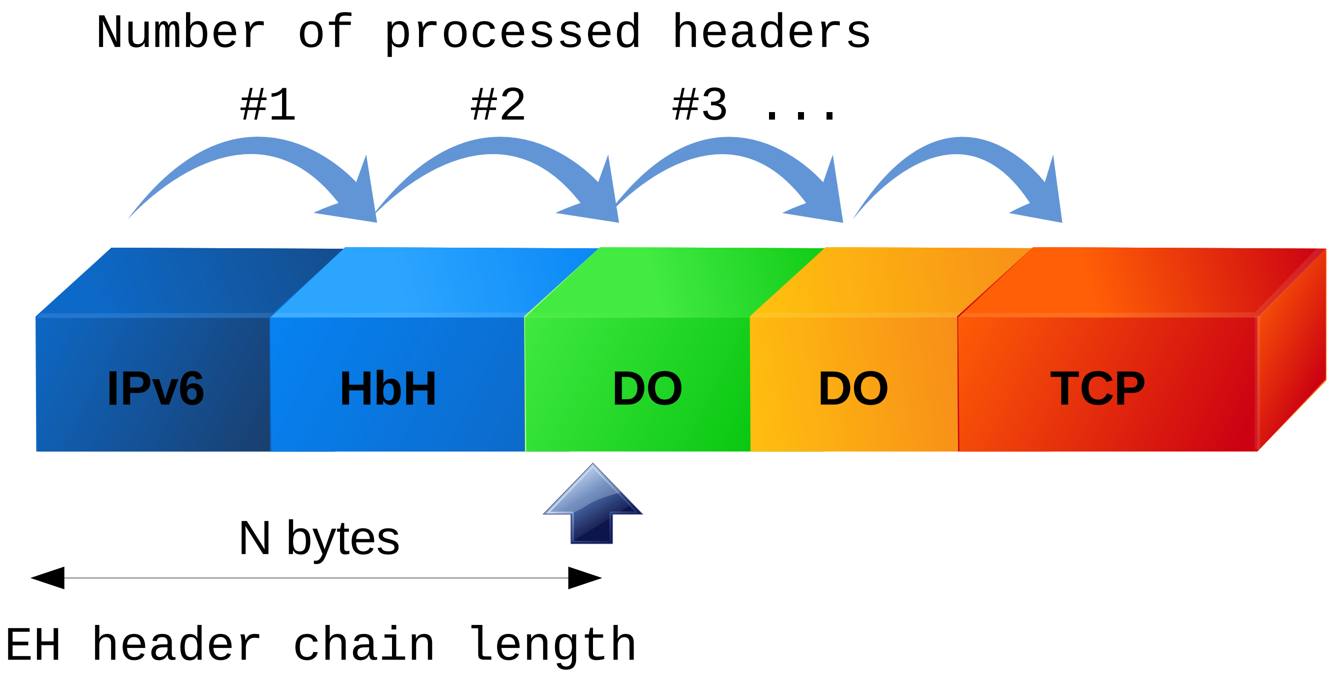
IPv6 Extension Headers

- IPv6 options are included in “extension headers”
  - These headers sit between the IPv6 header and the upper-layer protocol
  - There may be multiple instances, of multiple extension headers, each with multiple options
- Hence, IPv6 follows a “header chain” type structure. e.g.,



Processing the IPv6 header chain

- It is harder to spot e.g. layer-4 information (if at all possible)

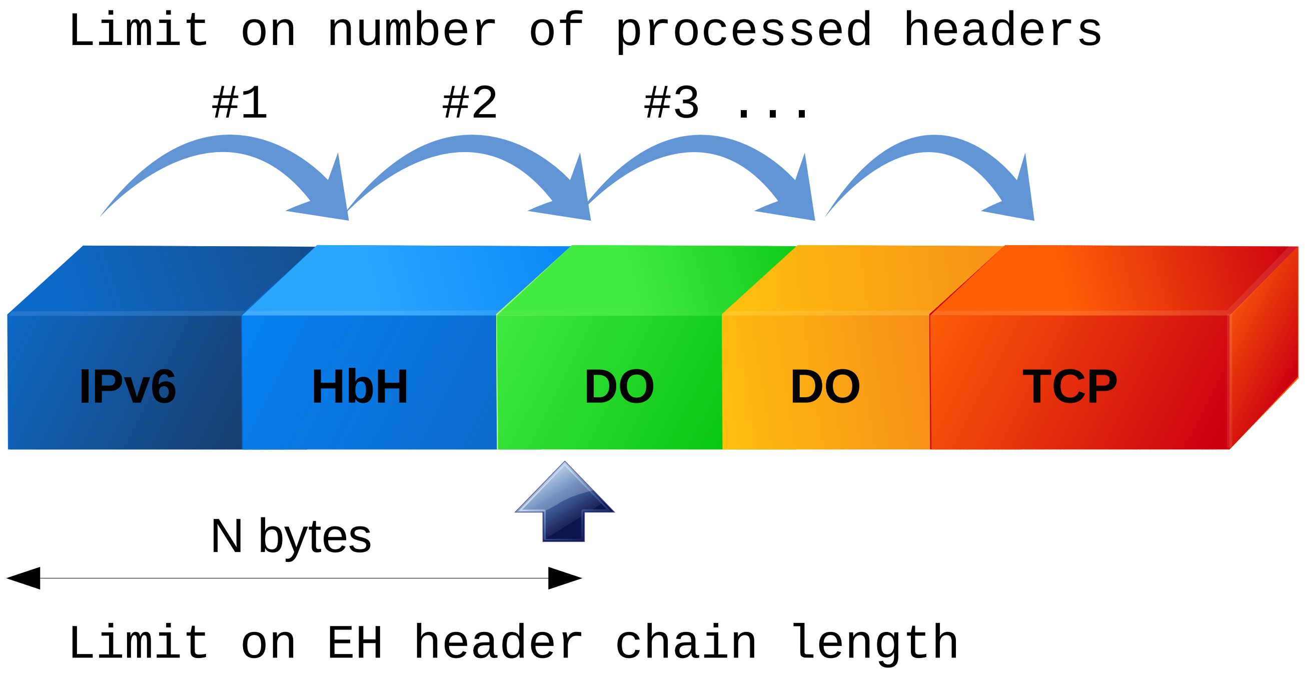


IPv6 EHs & Operational Reality

Requirement to process EH header chains

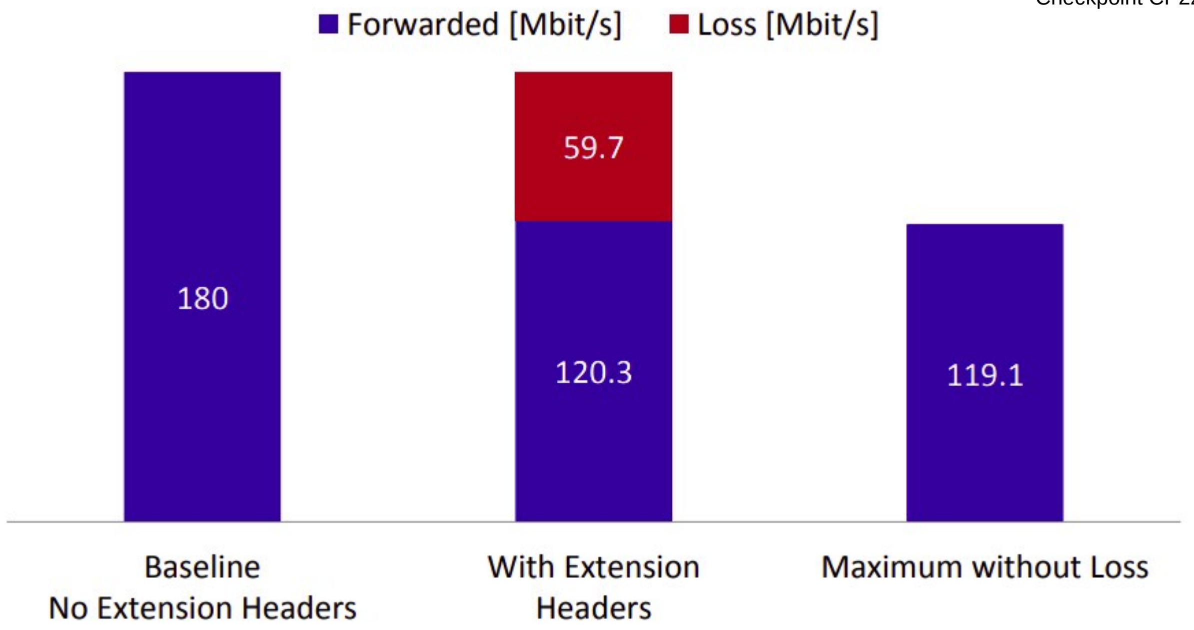
- Some middle-boxes need to obtain layer-4 information
- Requirement to process layer-4 information:
  - Enforcing infrastructure ACLs
  - DDoS Management and Customer Requests for Filtering
  - ECMP and Hash-based Load-Sharing
  - Firewalling
  - IDP/IPS
- When unable, they may drop the corresponding packet
  - Packet Forwarding Engine Constraints

Packet Forwarding Engine Constraints


Limit on number of processed headers
#1
#2
#3 ...
IPv6
HbH
DO
DO
TCP
N bytes
Limit on EH header chain length

Other processing constraints

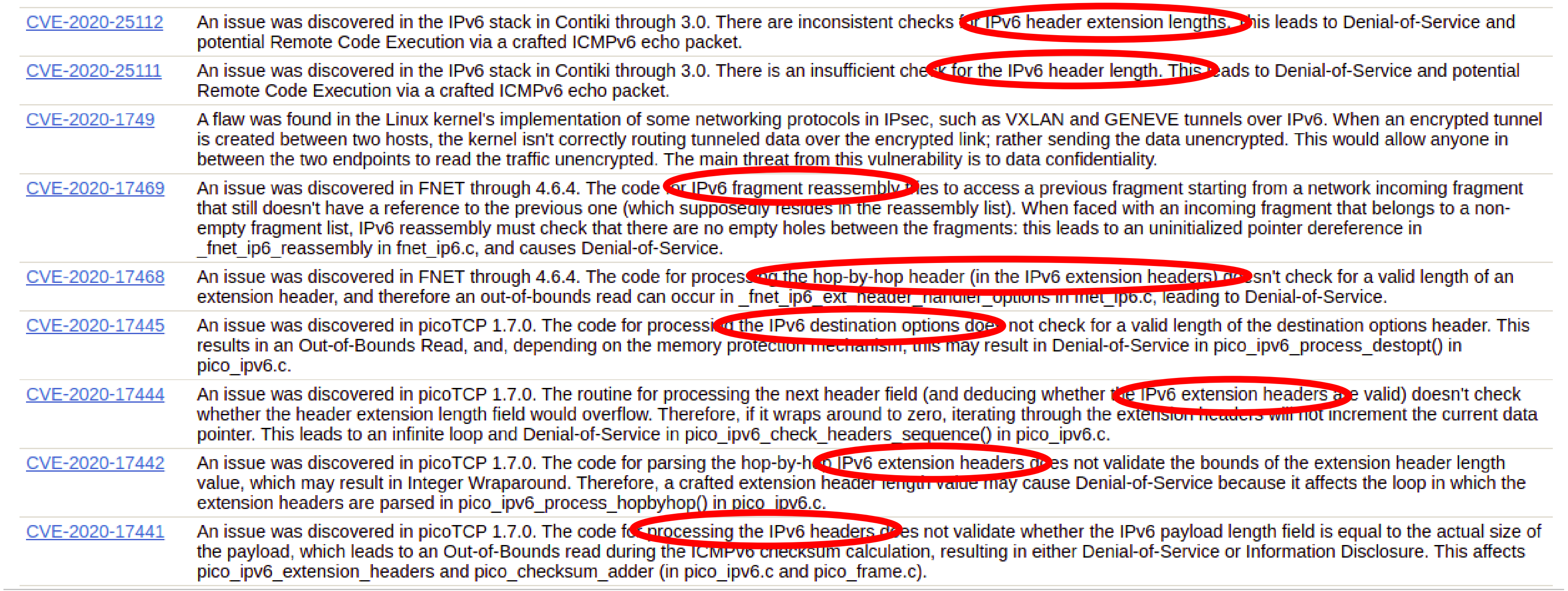
Checkpoint CP2210



Zack, E., "IPv6 Security Assessment and Benchmarking". IPv6 Hackers #1, July 2013.

IPv6 EHs & Security

IPv6 EHs & Security Issues

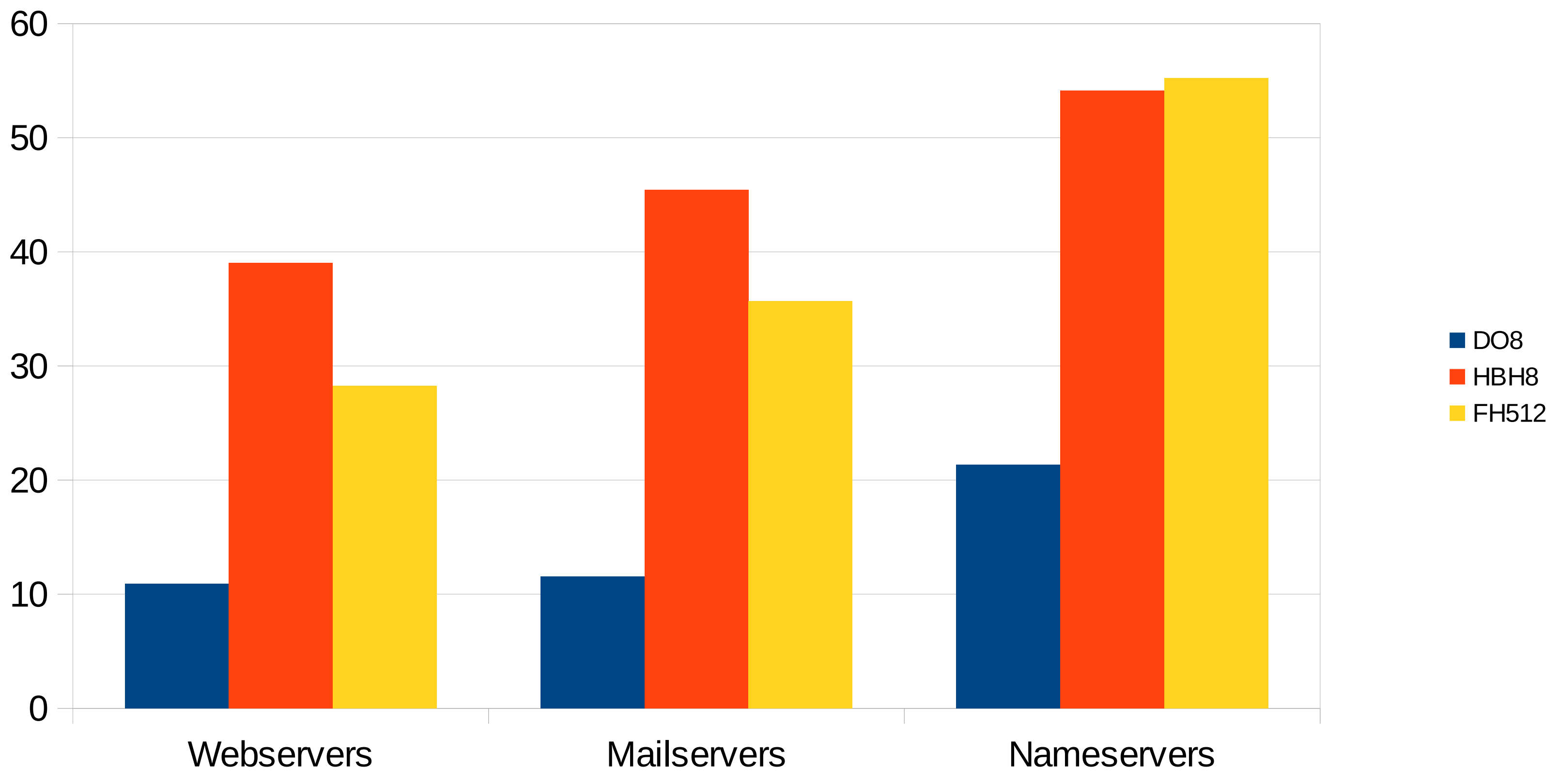


CVE-2020-25112	An issue was discovered in the IPv6 stack in Contiki through 3.0. There are inconsistent checks for IPv6 header extension lengths. This leads to Denial-of-Service and potential Remote Code Execution via a crafted ICMPv6 echo packet.
CVE-2020-25111	An issue was discovered in the IPv6 stack in Contiki through 3.0. There is an insufficient check for the IPv6 header length. This leads to Denial-of-Service and potential Remote Code Execution via a crafted ICMPv6 echo packet.
CVE-2020-1749	A flaw was found in the Linux kernel's implementation of some networking protocols in IPsec, such as VXLAN and GENEVE tunnels over IPv6. When an encrypted tunnel is created between two hosts, the kernel isn't correctly routing tunneled data over the encrypted link; rather sending the data unencrypted. This would allow anyone in between the two endpoints to read the traffic unencrypted. The main threat from this vulnerability is to data confidentiality.
CVE-2020-17469	An issue was discovered in FNET through 4.6.4. The code for IPv6 fragment reassembly tries to access a previous fragment starting from a network incoming fragment that still doesn't have a reference to the previous one (which supposedly resides in the reassembly list). When faced with an incoming fragment that belongs to a non-empty fragment list, IPv6 reassembly must check that there are no empty holes between the fragments: this leads to an uninitialized pointer dereference in _fnet_ip6_reassembly in fnet_ip6.c, and causes Denial-of-Service.
CVE-2020-17468	An issue was discovered in FNET through 4.6.4. The code for processing the hop-by-hop header (in the IPv6 extension headers) doesn't check for a valid length of an extension header, and therefore an out-of-bounds read can occur in _fnet_ip6_ext_header_handler_options in fnet_ip6.c, leading to Denial-of-Service.
CVE-2020-17445	An issue was discovered in picoTCP 1.7.0. The code for processing the IPv6 destination options does not check for a valid length of the destination options header. This results in an Out-of-Bounds Read, and, depending on the memory protection mechanism, this may result in Denial-of-Service in pico_ipv6_process_destopt() in pico_ipv6.c.
CVE-2020-17444	An issue was discovered in picoTCP 1.7.0. The routine for processing the next header field (and deducing whether the IPv6 extension headers are valid) doesn't check whether the header extension length field would overflow. Therefore, if it wraps around to zero, iterating through the extension headers will not increment the current data pointer. This leads to an infinite loop and Denial-of-Service in pico_ipv6_check_headers_sequence() in pico_ipv6.c.
CVE-2020-17442	An issue was discovered in picoTCP 1.7.0. The code for parsing the hop-by-hop IPv6 extension headers does not validate the bounds of the extension header length value, which may result in Integer Wraparound. Therefore, a crafted extension header length value may cause Denial-of-Service because it affects the loop in which the extension headers are parsed in pico_ipv6_process_hopbyhop() in pico_ipv6.c.
CVE-2020-17441	An issue was discovered in picoTCP 1.7.0. The code for processing the IPv6 headers does not validate whether the IPv6 payload length field is equal to the actual size of the payload, which leads to an Out-of-Bounds read during the ICMPv6 checksum calculation, resulting in either Denial-of-Service or Information Disclosure. This affects pico_ipv6_extension_headers and pico_checksum_adder (in pico_ipv6.c and pico_frame.c).

It's 2021, and we can't properly process an IPv6 header chain!

IPv6 EHs in the Real World

Alexa’s Top-1M domains: Packet Drop rate



Recent IETF Work

Recent IETF work

- RFC 9098:
  - Raises awareness about operational implications of IPv6 EHs.
- RFC 7872:
  - Real world measurements about IPv6 EHs.
- RFC 8200 (IPv6 Standard):
  - Incorporates RFC 5722, RFC 7112, RFC 6946, RFC 7739, and RFC 8021.
  - But whole fragmentation/reassembly section replaced by Errata ID 5945!
- RFC 8900:
  - IP fragmentation considered fragile

Conclusions

Conclusions

- Operational challenges are associated with the very design/nature of IPv6 EHs.
- It’s also a chicken-and-egg problem:
  - Nobody will invest in supporting something that doesn’t have a good use-case.
  - No good use-case until the feature is widely-supported/dependable/reliable.
- For the foreseeable future, their use will be associated to “limited domains”.

Questions?

Thanks!

Fernando Gont

fernando.gont@edgeuno.com

edgeuno.com