Threats and Surprises behind IPv6 Extension Headers

Luuk Hendriks*, Petr Velan†, Ricardo de O. Schmidt*, Pieter-Tjerk de Boer*, and Aiko Pras*
*Design and Analysis of Communication Systems
University of Twente, the Netherlands
Email: {luuk.hendriks,r.schmidt,p.t.deboer,a.pras}@utwente.nl
†CESNET, a.l.e
Zikova 4, 160 00 Prague 6, Czech Republic
Email: petr.velan@cesnet.cz

***Abstract*—The concept of Extension Headers, newly introduced with IPv6, is elusive and enables new types of threats in the Internet. Simply dropping all traffic containing any Extension Header –a current practice by operators– seemingly is an effective solution, but at the cost of possibly dropping legitimate traffic as well. To determine whether threats indeed occur, and evaluate the actual nature of the traffic, measurement solutions need to be adapted. By implementing these specific parsing capabilities in flow exporters and performing measurements on two different production networks, we show it is feasible to quantify the metrics directly related to these threats, and thus allow for monitoring and detection. Analysing the traffic that is hidden behind Extension Headers, we find mostly benign traffic that directly affects end-user QoE: simply dropping all traffic containing Extension Headers is thus a bad practice with more consequences than operators might be aware of.**

I. Introduction

In the current day Internet, we know that for every security measure, there is a multitude of people–with malicious intents– trying to break or evade these measures. This is not a new phenomenon, and security officers try to configure their (flow) monitoring systems in ways such that these attacks become visible. With the increasing adoption and deployment of IPv6 in the Internet however, there are many new possibilities for attackers in terms of misuse, allowing for *e.g.* Advanced Persistence Threats (APTs) and Denial of Service (DoS) attacks. Relying on security approaches from the IPv4 era is not sufficient. In order to stay one step ahead of attackers, security officers and operators should be aware which of the new features in IPv6 are exploitable, and to what extent their monitoring tools are able to detect traffic related to these types of misuse.

One essential difference in IPv6 is the concept of Extension Headers (EHs). Because of their position in the IPv6 header, namely in between the IP header and the upper layer header (*e.g.* TCP or UDP), network devices need to perform extra steps when determining what kind of traffic is actually inside a packet. Tailored solutions to traverse and parse these headers are not always available, be it for financial, technical or other reasons. Simply dropping all the packets containing EHs is an applied approach [1], but means all legitimate traffic with EHs is discarded as well.

Measurement solutions need to be, just like security solutions and other network devices, adapted to support the newly introduced aspects of the IPv6 protocol. Similar to *e.g.* firewalls, a measurement tool needs to traverse and parse possible EHs in order to report statistics on the actual upper layer payload. In the case of flow-based measurements, one will see the protocol number of the first EH in a packet, but have no clue about the remaining contents of that packet. With adapted measurement tools at hand, one can analyze network traffic containing EHs and determine the actual –possibly surprising– nature of traffic hidden behind these EHs, and see what is really being forwarded over the network.

Focussing on EHs, several threats are described in the literature and proven feasible in lab setups, which we make visible in our measurements (§ III): evading Access Control Lists (ACLs) by injecting an EH, *e.g.* sending SSH traffic with *Hop-by-Hop Options*, a possible first step from an APT; causing a DoS or again evading middle-boxes by sending long chains of EHs; causing a DoS by sending artificially large EHs, aiming for devices with limited memory for processing these EHs.

In this paper, we ask ourselves how one can determine whether traffic containing EHs should be forwarded or dropped. We show how flow-based measurements can be adapted to include information on *hidden traffic*, *i.e.* traffic behind one or more EHs.

Contributions: In this work, we qualify and quantify the traffic characteristics that are hidden by EHs, based on measurements in two different types of production networks, namely CESNET, the Czech National Research and Educational Network (NREN), and UTNET, a campus network including residences. We show that by enhancing flow exporters, both legitimate –but overlooked– network traffic, and possibly malicious traffic is made visible: up to 0.7% of IPv6 flows contained hidden information behind one EH. Furthermore, we show that longer chains and large headers do occur, but are exceptional. Our analysis on fragmentation characteristics provides insights on possible improvements for network operators, some directly influencing the Quality of Experience (QoE) of end-users, especially in the case of large DNS responses.

II. Background and Related Work

A. Extension Headers in IPv6

Extension Headers are optional headers between the IPv6 header and the higher layer header. The function of some of these headers have equivalents in IPv4, although in IPv4 the information was stored in fields in the IP header itself. An example of this is the Fragmentation header. No Extension Header is mandatory, and thus an IPv6 packet without any Extension Header (Fig. 1a) is perfectly valid. Multiple Extension Headers can be included by means of header chaining based on the Next Header field, as shown in Fig. 1b: the IP header points to the first Extension Header, which in turn points to the next Extension Header. The final Extension Header points to the actual higher layer header, in this case TCP. As the number and order of Extension Headers cannot be known a priori, devices processing packets need to perform checks on every packet. These extra checks do not only come with a performance penalty during operation, but also add complexity in the design of these devices.

B. Functionality

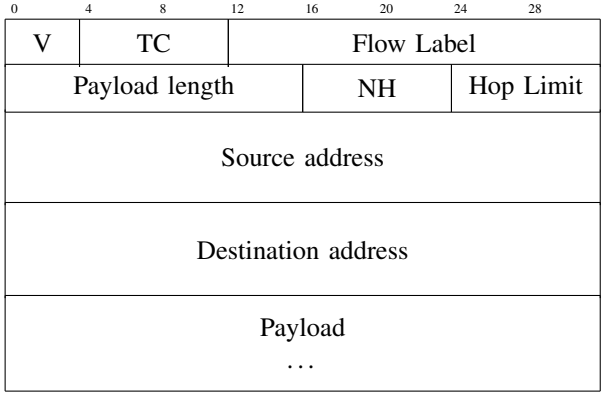
When the IPv6 standard (RFC 2460 [2]) came to be, some of the described Extension Headers either fulfilled a direct requirement, while others were intended for (future) flexibility of the protocol. Table I shows all headers defined in the RFC, and the protocols marked 'EH' by IANA in [3].

TABLE I: Extension Headers defined in RFC 2460 and IANA assignments

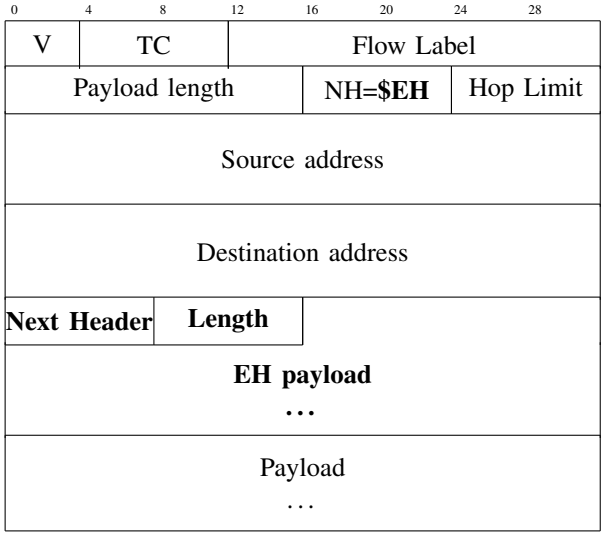
Decimal	Protocol	RFC	IANA
0	Hop-by-Hop Options	✓	✓
43	Routing	✓	✓
44	Fragment	✓	✓
50	Encapsulating Security Payload	✓	✓
51	Authentication	✓	✓
60	Destination Options	✓	✓
135	Mobility Header		✓
139	Host Identity Protocol		✓
140	Shim6		✓
253	Experiments/testing purposes		✓
254	Experiments/testing purposes		✓

The Hop-by-Hop Options and Destination Options are headers in forms of Type-Length-Value (TLV) fields. These headers represent options that should be processed at every forwarding hop or only at the destination, respectively. The highest order three bits determine how a node should act if a packet with a header unknown to that node is observed, and whether the data of that header may be changed en-route. Other than the form and the meaning of the three bits, there are no further definitions in the standard for these Option headers.

The Routing header is used by the source node to specify one or more intermediate nodes en-route to the final destination of the packet. RFC 2460 only describes one type of this header, Type 0, which is deprecated now because of security issues [4]. Other defined Types of this header are Type 1 (unused, originates from the DARPA project Nimrod) and Type 2, which is used in Mobile IPv6.[1] The Fragmentation Header replaces the function of the Identification, Flags and Fragment Offset in the IPv4 header. Finally, the Authentication Header and Encapsulating Payload Header fulfil the functions of IPSEC's AH and ESP, in similar fashion to how it is used in IPv4.



(a) Without Extension Headers



(b) With one Extension Header: *$EH* is the protocol number of the Extension Header between the IPv6 header and the upper layer protocol. The *Next Header* field in the Extension Header describes the protocol number of the upper layer protocol.

Fig. 1: IPv6 Header layouts [2]

As the standard has been around for roughly two decades, deprecation of a certain feature or part does not mean it does not occur in the Internet anymore. Different types of devices with varying implementations form a heterogeneous reality vastly different from the latest version of the standard. But even in that latest version of the standard, multiple types of misuse are possible.

[1] https://tools.ietf.org/html/rfc3775#section-6.4

C. Misuses and caveats

Due to their dynamic nature, correctly implementing EH handling is challenging. Their presence, number and length(s) will vary per packet. Not only network stacks and (hardware) forwarding mechanisms are subject to this challenge: firewalls and ACLs possibly require additional configuration to cover situations where EHs are used.

An example of such **middle-box evasion** is presented in [5]: configuring a firewall to "`block ssh; accept all;`" requires the firewall to traverse the EH-chain and find out the actual upper layer protocol. Only then can it determine whether the transport protocol is TCP, destined for port 22, and thus drop the packet.

Long header chains have implications [6] in scenarios where *e.g.* stateless firewalls need information up to the upper layer protocol: when the packet is fragmented, and due to the long header chain the first fragment does not contain all that needed information, the firewall can possibly not act on that packet appropriately.

Similar to the long header chains, the **length of the EHs** can trigger unexpected behavior. Furthermore, where limited memory for EH-processing is expected in forwarding devices, sending artificially large EHs can form a DoS attempt.

Aside of these ways of intentional misuse, there are several caveats (or possibly surprises) when EHs come into play. One of these is clearly related to the aforementioned threats: by choosing to drop all packets with EHs, one might drop a surprisingly large share of actually benign traffic. In case of *e.g.* fragmentation (handled by an EH in IPv6) large, fragmented answers from servers might never reach a client.

When performing (flow) measurements and aggregating on the protocol number without traversing the EH-chain, not only will the actual type of traffic be hidden: the characteristics of the flows will be vastly different as well. For example, when aggregating fragmented (EH 44) traffic, without using the actual upper layer ports to group the packets on, multiple distinct flows will be aggregated into a big, single flow record. When detection algorithms are implemented on finding big flows, this will result in false positives. At the same time, looking for many small flows, *e.g.* in brute-force dictionary attacks, fails as well.

Attempts at clarifying or even deprecating (parts of) standards might improve the situation in the future. However, old implementations of network stacks and security appliances will be active for years, including faulty, exploitable implementations.

D. Flow-based measurements / IPFIX

Flow-based measurements are based on aggregation: packets are grouped based on a certain set of fields (*e.g.* source and destination IP addresses, transport layer source and destination port, and protocol), and statistics like number of packets and number of bytes are accounted. Packet payload is typically lost. This aggregation allows for reasoning on a higher conceptual level, as well as scalable solutions where processing a large number of packets is not feasible.

The process of aggregation happens either on a networking forwarding device, *e.g.* a router, or at a dedicated flow exporter which processes a mirror of the network traffic (in forms of packets). The router or the flow exporter then sends out (*exports*) the generated flow records to a *collector*, where analysis takes place. Multiple exporters can export to a single collector, enabling for easy analysis of multiple vantage points.

Two well-known standards for these flow measurements are *NetFlow* (originally by Cisco, often available on forwarding devices) and the IETF's standardization effort *IPFIX*. An important feature in IPFIX is its extensibility, which allows exporting of new so-called Information Elements (IEs), a concept we leverage in this work: while the IANA assigned list [7] of IEs is extensive, it does not cover all the metrics we are interested in.

An essential aspect of flow-based measurements is how the *flow cache* in the exporter is handled: when implementing new IEs, one needs to decide whether packets should be grouped on that IE, possibly creating more distinct flow records than prior to introducing the new IE.

For a comprehensive overview of all parts and processes in flow-based measurements refer to [8] by Rick Hofstede *et al.*, or see [9] by Brian Trammell and Elisa Boschi for an IPFIX-specific introduction.

E. Related Work

To the best of our knowledge, no large-scale passive measurements on IPv6 Extension Headers have been performed in recent years. Active measurements efforts by Fernando Gont, Jen Linkova *et al.* are documented in an IETF Informational document [1], showing that not only fragmentation headers but EHs in general are often dropped in transit networks. The Internet-Draft [10] by Fernando Gont *et al.* focusses on operational implications regarding EH handling.

In [11], Martin Elich *et al.* evaluate traffic encapsulated in IPv6 tunneling mechanisms, also using IPFIX and implementing custom Information Elements. A comprehensive overview of threats introduced with IPv6 is given by Johanna Ullrich *et al.* in [12].

III. Measurement setup

We performed passive measurements on multiple links, to observe which and how Extension Headers are actually used on the Internet. In two different production networks, one or more links were measured using dedicated flow probes, exporting IPFIX records containing our additional Information Elements. Only IPv6 flows were considered, for a time period of roughly a month. Details on these networks and the exporting process are described in the following sections.

A. Networks / Vantage points

1) CESNET: The NREN of the Czech Republic. Dedicated flow probes are deployed on 8 different links, metering unsampled, exporting to a single collecting machine. These are the external links, so any traffic going in or out of CESNET is measured by one of the 8 probes. No specific filtering is

active on the links. The collection period was December 1 - December 28, 2016.

2) UTNET: The campus network of the University of Twente. A dedicated flow probe is deployed monitoring the uplink of the network, unsampled. This uplink connects office buildings, lecture halls, as well as student residences. No specific filtering is active on this uplink, and the collection period spanned the same four weeks as at CESNET. While a campus network is naturally different from a consumer access network, the students and employees living on-campus use this same network as if it were a commercial Internet Service Provider (ISP).

B. Extraction of properties

We implemented a plugin for the dedicated flow probes to traverse the EHs and extract the properties listed in Table II.

TABLE II: Overview of essential EH-related properties

Property	Type	Size	in key
No. of EHs	integer	8 bits	✓
Total size of EHs	integer	16 bits	
Order of EHs	string	255 chars	✓
Upper layer protocol	integer	8 bits	✓
Upper layer source port	integer	16 bits	✓
Upper layer destination port	integer	16 bits	✓
Upper layer ICMP Type & Code	integer	16 bits	✓

NB: The IANA list in [7] contains Information Elements that could be used, but to make a clear distinction of our own implemented fields, we created new fields. Some of these IANA-assigned fields have shortcomings, for example the IE *ipv6ExtensionHeaders* (ElementId 64) lists all observed EHs but does not tell anything about the order. The normal fields for transport layer information could and should be reused, were this implemented as a production feature.

In order to populate the newly defined Information Elements in the IPFIX records, every packet passing through the metering process is checked for certain fields. This happens in addition to the already existing export behavior, *i.e.* the usual Information Elements are still exported. To obtain information about the EHs, the (possible chain of) Next Headers must be followed, until a header is observed that is not defined as an EH. While performing this traversal, the following actions are performed:

1) Increase EH count (first entry in Table II)
2) Add size of EH in bytes to sum total (second entry)
3) Append EH protocol number to list (third entry)

Upon observing the first non-EH (thus a protocol number not listed in Table I), all information about the EHs has been obtained. The non-EH protocol number tells us what the actual upper layer protocol is, and is exported as such. Based on that protocol number, the payload can be parsed to extract transport layer port numbers or ICMP type and code.

C. Adapting flow cache keys

The set of fields aggregated on in the flow exporter naturally determines which fields are visible in the flow records leaving the exporter. The flow cache, containing the statistics of flows, uses this set of fields as a *flow key* mapping to the statistics (*i.e.* packet and byte counters). Therefore, for every flow that we want to distinguish, this set needs to be unique. In case of the hidden traffic that we want to expose, new fields are introduced that can and have to be used in the flow key, thus the aggregation. For our newly introduced IEs, the last column in Table II marks whether the property is indeed included in the flow key. In case of TCP and UDP on the actual upper layer, we add the protocol number, the source port and the destination port to the flow key. Note that without traversing and parsing the EH chain, these three fields are not available: two fragmented flows between a pair of hosts would show up as a single flow record, containing the sum of packets and bytes of both flows. Similarly for ICMP, the type and code are used in the flow key. Lastly, the number and order of EHs are used in the flow key as well: if one of these things changes 'within a flow', we do not want it to go unnoticed, ergo export separate records.

D. Ethical considerations

While our measurements require IP addresses to aggregate packets to flows, we do not need the IP addresses themselves. Thus, systematic and deterministic anonymization of the addresses in the export process on the different vantage points does not interfere with our analysis, while preserving privacy of users on these networks.

IV. Results and Discussion

A. Share of traffic containing EHs

Firstly, we look at what share of traffic contains one or more EHs. An overview of the results for both networks is given in Table III. For CESNET, we found 0.7% of IPv6 flows to contain one or two EHs. The share for UTNET is smaller, at 0.1%. Packet count and byte count wise, the shares are smaller than for the number of flows on CESNET (0.2% and 0.3%, respectively), while on UTNET these numbers are equivalent.

Note that in case of fragmented traffic, these flow counts are derived *after reassembly*. As L4 port information lacks from non-first fragments, our flow exporters export first-fragments and non-first-fragments as separate flow records. Thus, the numbers in the overview tables are corrected for that by merging these separate flow records and counting them as a single flow.

Overviews of all the observed protocols over IPv6, which can be obtained without any additional intelligence on flow exporters, are listed in Table IV. This table shows which protocols the aforementioned 0.7% and 0.1% are comprised of: focussing on EHs in that table, we find mainly Fragmentation Headers and, in the case of CESNET, also Hop-by-Hop Options.

B. Chains of multiple EHs

More details on the EH chains longer than 1 are provided in Table V. The clear majority of flows, packets and bytes are accounted for by ICMP6 containing Hop-by-Hop Options

TABLE III: Measurement overview: Observed numbers of EHs

Dataset	No. of EHs	Flows	Packets	Bytes	Notes
CESNET	0	2.5G (99.3%)	86.8G (99.8%)	81.0Ti (99.7%)	NREN/transitional network;
	1	17.0M (0.7%)	197.4M (0.2%)	214.4Gi (0.3%)	8 vantage points;
	2	654 (0.0%)	72.1K (0.0%)	48.3Mi (0.0%)	unsampled
UTNET	0	2.2G (99.9%)	158.5G (99.9%)	140.6Ti (99.9%)	Campus network;
	1	2.0M (0.1%)	169.1M (0.1%)	148.6Gi (0.1%)	1 vantage point;
	2	58 (0.0%)	5.4K (0.0%)	3.7Mi (0.0%)	unsampled

TABLE IV: CESNET/UTNET: Flows/packets/bytes per protocol

CESNET				**UTNET**			
Protocol	Flows	Packets	Bytes	Protocol	Flows	Packets	Bytes
UDP	1.1G (45.6%)	13.2G (15.2%)	9.2Ti (11.4%)	TCP	1.5G (67.0%)	111.3G (70.2%)	101.5Ti (72.1%)
TCP	738.0M (29.7%)	70.5G (81.1%)	71.4Ti (87.9%)	UDP	554.7M (25.4%)	46.7G (29.4%)	39.0Ti (27.7%)
ICMP6	591.4M (23.8%)	2.8G (3.2%)	279.9Gi (0.3%)	ICMP6	163.7M (7.5%)	427.4M (0.3%)	36.2Gi (0.0%)
IPv6-Frag	10.1M (0.4%)	187.1M (0.2%)	213.7Gi (0.3%)	IPv6-Frag	1.8M (0.1%)	4.7M (0.0%)	4.3Gi (0.0%)
HOPOPT	7.0M (0.3%)	10.4M (0.0%)	912.2Mi (0.0%)	PIM	375.7K (0.0%)	376.4K (0.0%)	34.5Mi (0.0%)
IPv6-NoNxt	3.2M (0.1%)	4.9M (0.0%)	186.1Mi (0.0%)	HOPOPT	154.1K (0.0%)	171.6K (0.0%)	17.0Mi (0.0%)
PIM	309.6K (0.0%)	1.5M (0.0%)	198.5Mi (0.0%)	IPv6	101.0K (0.0%)	20.0M (0.0%)	3.1Gi (0.0%)
IPv4	16.6K (0.0%)	270.1M (0.3%)	110.8Gi (0.1%)	ESP	68.6K (0.0%)	164.2M (0.1%)	144.2Gi (0.1%)
OSPFIGP	4.3K (0.0%)	116.4K (0.0%)	8.4Mi (0.0%)	IPv6-Opts	8.9K (0.0%)	8.9K (0.0%)	976.0Ki (0.0%)
ESP	2.1K (0.0%)	265.6K (0.0%)	65.0Mi (0.0%)	Reserved	20 (0.0%)	20 (0.0%)	2.4Ki (0.0%)
Other	713 (0.0%)	793 (0.0%)	245.2Ki (0.0%)	Other	6 (0.0%)	6 (0.0%)	472 (0.0%)

TABLE V: Longer EH chains detailed

EHs	Upper proto	Flows	Packets	Bytes
CESNET:				
HOPOPT, IPv6-Frag	IPv6-ICMP	501	25.7K	16.6Mi
IPv6-Frag, ESP	ESP	144	46.4K	31.7Mi
IPv6-Frag, IPv6-Frag	TCP	3	21	1.6Ki
254, HIP	XTP	1	1	1.4Ki
IPv6-Route, AH	151	1	1	1.4Ki
HOPOPT, AH	ARIS	1	1	996
AH, Shim6	192	1	1	299
AH, ESP	ESP	1	1	176
253, IPv6-Route	PUP	1	1	158
UTNET:				
IPv6-Frag, ESP	ESP	58	5.4K	3.7Mi

(proto 0) followed by a Fragmentation Header (proto 44). The other combinations of headers are only observed once. Note that protocol numbers 253 and 254, used for experimentation and testing, are marked as an IPv6 Extension Header in [3], but these protocol numbers can be used without adhering to actual extension header wire formats. Interpreting these headers as if they are extension headers might lead to bogus information, which might have happened for the two flows listed in the table.

C. Actual, hidden upper layer protocols

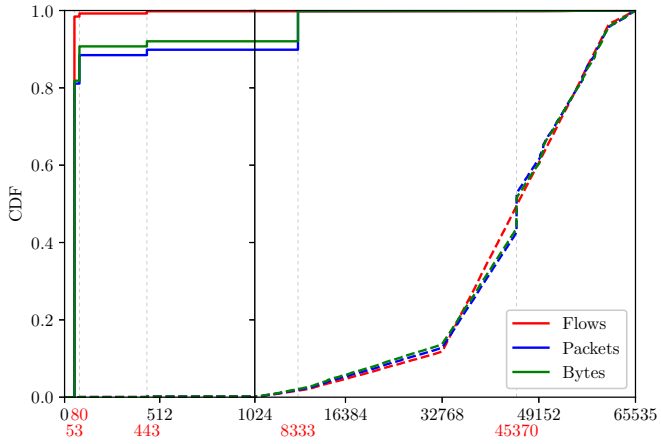
Aggregating the first EH and the actual upper layer protocol, we find UDP preceded by Fragmentation Headers to form the lionshare of the traffic on both networks, albeit only in terms of flows. For CESNET, as detailed in Table VI, we find the fragmented UDP to cover 58.0% of flows, but over 99% of transfered bytes. On UTNET (Table VI) on the other hand, 87.3% of flows is accounted for by fragmented UDP, while it is less than 3% of transfered bytes. IPSEC ESP is responsible for 97.1% of bytes on UTNET, but negligible for all flow, packet and byte counts on CESNET. Due to its encrypted nature, ESP does not allow for further analysis within scope of this research.

A significant share of the flows on CESNET is comprised of ICMP6 preceded by Hop-by-Hop Options. At 40.9% that is a fivefold of what is observed at UTNET. This shows different (types of) networks can vastly differ in terms of EHs being transfered, just like they differ with 'normal' traffic. As ICMP6 has a different–often more important–role in IPv6 compared to IPv4, simply dropping all traffic containing EHs would result in loss of possibly essential ICMP information. In § IV-F, we analyze the actual types and codes of this hidden ICMP traffic in more detail.

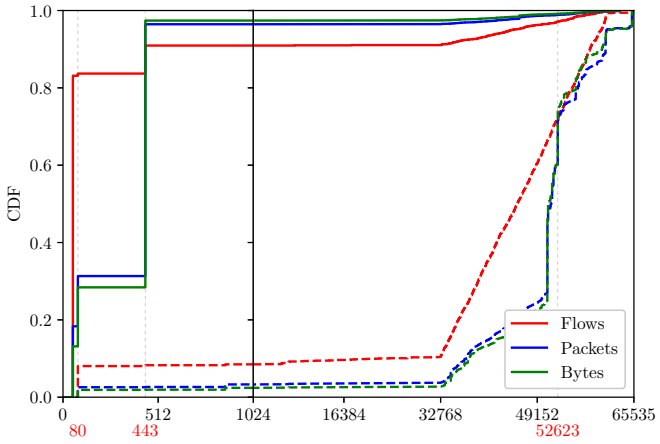
D. Breakdown of hidden TCP and UDP traffic

Extension headers hide, among other, TCP and UDP traffic that is directly related to end-user QoE. Our exporters extracted information from the actual upper layer protocols, *e.g.* source and destination ports for UDP and TCP, which are otherwise not available for analysis. In this section we present the distributions of those ports in terms of flow, packet and byte counts, in order to draw conclusions regarding the actual nature of the hidden traffic. These distributions are visualized in Fig. 2a and 2c for CESNET, and Fig. 2b and 2d for UTNET. Note that for TCP and UDP, the observed EH is with negligible exception always the Fragmentation Header. Thus, the following analysis is mainly addressing fragmented traffic, which likely explains some of the found phenomena.

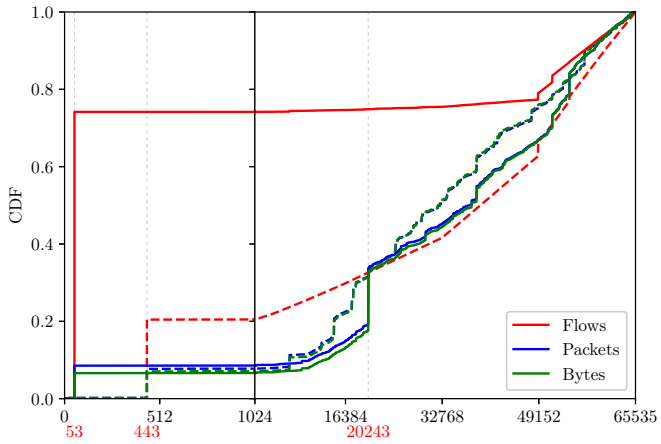
1) TCP traffic: Analysis of the TCP source port distribution (Fig. 2a and 2b) observed in the traffic shows 90% of traffic originates from ports below 1024, hinting at server traffic.



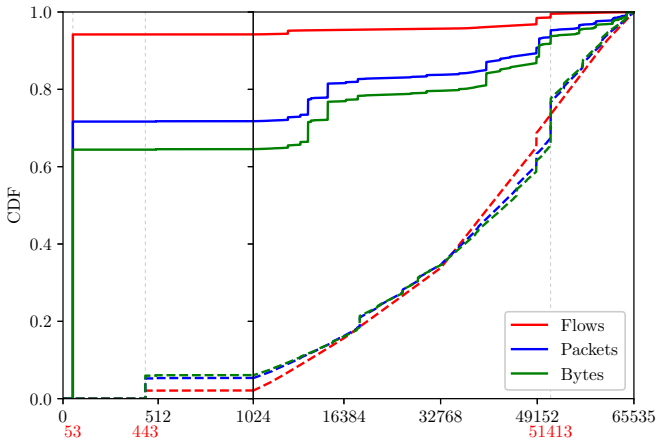
(a) TCP CESNET



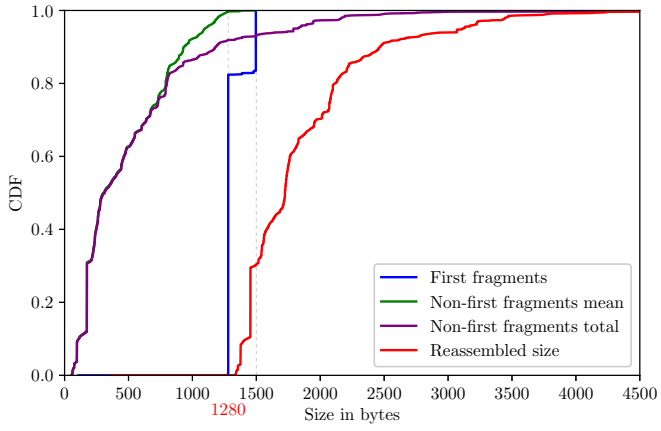
(b) TCP UTNET



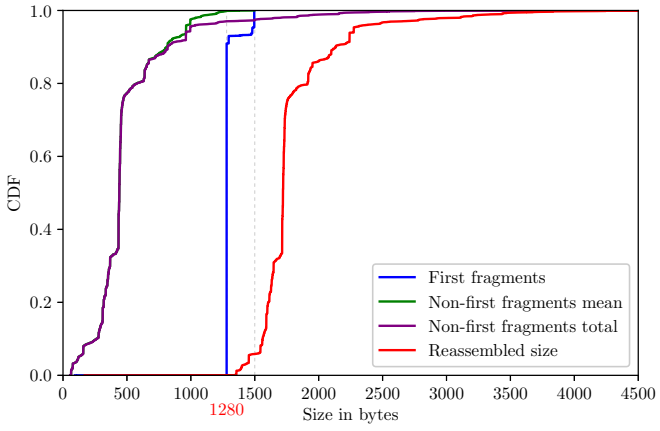
(c) UDP CESNET



(d) UDP UTNET



(e) Fragment sizes CESNET



(f) Fragment sizes UTNET

Fig. 2: Transport layer port distribution of hidden traffic, and fragmentation characteristics.
CESNET plots on the left, UTNET plots on the right.
NB: Horizontal axes are non-linear. In the port plots, **dashed** lines represent **d**estination ports; **solid** lines for **s**ource ports.

TABLE VI: CESNET/UTNET: Extension Headers and the actual upper layer

CESNET					UTNET				
EHs	upper	Flows	Packets	Bytes	EHs	upper	Flows	Packets	Bytes
44;	UDP	9.9M (58.0%)	186.4M (94.3%)	213.1Gi (99.3%)	44;	UDP	1.7M (87.3%)	4.7M (2.8%)	4.3Gi (2.9%)
0;	ICMP6	7.0M (40.9%)	10.4M (5.2%)	895.6Mi (0.4%)	0;	ICMP6	154.1K (7.7%)	171.6K (0.1%)	17.0Mi (0.0%)
44;	ICMP6	117.0K (0.7%)	273.5K (0.1%)	198.5Mi (0.1%)	50;	ESP	68.6K (3.4%)	164.2M (97.1%)	144.2Gi (97.1%)
44;	TCP	72.3K (0.4%)	399.5K (0.2%)	378.5Mi (0.2%)	44;	ICMP6	20.0K (1.0%)	42.9K (0.0%)	29.0Mi (0.0%)
50;	ESP	2.1K (0.0%)	265.6K (0.1%)	65.0Mi (0.0%)	60;	IPv6	8.9K (0.4%)	8.9K (0.0%)	976.0Ki (0.0%)
0;44;	ICMP6	501 (0.0%)	25.7K (0.0%)	16.6Mi (0.0%)	44;	TCP	1.8K (0.1%)	16.2K (0.0%)	16.4Mi (0.0%)
44;50;	ESP	144 (0.0%)	46.4K (0.0%)	31.7Mi (0.0%)	44;50;	ESP	58.0 (0.0%)	5.4K (0.0%)	3.7Mi (0.0%)
254;	176.0	4 (0.0%)	8 (0.0%)	1.5Ki (0.0%)	253;	217.0	1 (0.0%)	1 (0.0%)	72 (0.0%)
44;44;	TCP	3 (0.0%)	21 (0.0%)	1.6Ki (0.0%)	253;	218.0	1 (0.0%)	1 (0.0%)	72 (0.0%)
Other	Other	219 (0.0%)	219 (0.0%)	120.1Ki (0.0%)	Other	Other	0 (0.0%)	0 (0.0%)	0 (0.0%)

This is the case for both CESNET and UTNET, and can be explained by the nature of small requests resulting in large responses (thus fragmentation) that is often seen in networked services.

The plots feature a non-linear x-axis to include more detail on the first 1024 ports. In both networks, the first (and largest) share of traffic is attributed to source port 53, likely large DNS responses containing DNSSEC signatures [13]. Furthermore, both networks show HTTP(S) traffic from ports 80 and 443 (as annotated in the plots). On CESNET another source port is noticeable, namely 8333[2], which is likely related to BitCoin network traffic. The remaining shares of traffic is divided over higher ports (most evident in Fig. 2b) where the ephemeral port ranges[3] are notable, hinting at client side initiated connections.

Looking at the distribution of TCP destination ports, we learn that 10% of traffic (in terms of flows) is directed at ports below 1024 in UTNET (Fig. 2b), mostly at port 80. In CESNET however, there is no sign of significant amounts of server-oriented traffic: the distribution in CESNET shows again the ephemeral port ranges, without any major jumps. In UTNET we observed noticeable jumps in the lower 50000-range, between port 50000 up to 52623. This might indicate use of specific (types of) software, *e.g.* certain BitTorrent clients.

Comparing the distributions of source and destination ports for TCP, we conclude that responses from servers are often fragmented, while the initial connection was not. With aggressive EH filtering on network edges, this means that *e.g.* webservers or nameservers do receive and handle incoming requests, while their outgoing responses might never leave the network.

2) UDP traffic: For UDP traffic originating from ports below 1024, the difference between the distribution of flows, and the distribution of packet count and byte count is more significant than for TCP, in both networks (Fig. 2c, 2d). In both networks, most *flows* originate from source port 53 (DNS, again likely with DNSSEC signatures). Small jumps are visible in both networks, port 20243 in CESNET being the most significant but only in terms of packets and bytes. This means a small number of large flows is responsible for this jump.

[2]https://bitcoin.org/en/full-node#network-configuration

[3]Ephemeral port ranges: IANA: 49152 – 65535 (used by recent version of MS Windows and FreeBSD); Linux: 32768 – 61000

The destination plots show an ostensible jump at port 443, most significant on CESNET (20% of flows). UDP traffic on port 443 is most likely QUIC, though one would expect this to be traffic originating from 443 (*e.g.* YouTube streaming) rather than destination port 443. Other possible explanations are uploading large files over QUIC (again, YouTube videos), because there is a jump for bytes and packets as well. Besides QUIC, other protocols could be explicitly configured to use UDP/443, *e.g.* OpenVPN. On UTNET, we find a jump (for packets and bytes) at destination port 51413, which is used by the popular BitTorrent client Transmission. Similar to CESNET, we see a jump at port 443.

E. Fragmentation characteristics

Looking into how the previously described traffic is fragmented, we find that at least 90% of the traffic is rightfully fragmented, *i.e.* has a total size of at least 1500 bytes after reassembly. In Fig. 2e and 2f, the distribution of sizes of first fragments and non-first fragments are plotted. Clearly, most first fragments are 1280 bytes in size, hinting at either a default value in fragmentation procedures in network stacks, or network administrators that prefer safely configured forwarding devices and chose the minimum IPv6 payload size for their MTUs.

The non-first fragments are plotted with the mean packet size within their flow, and the total size (of all non-first fragments combined, within their flow). These distributions only differ in the upper 20% and 10% for CESNET and UTNET, respectively, meaning that 80% and 90% of fragmented packets consist of only two fragments: one *first fragment*, and one *non-first fragment*. Combining all the fragments results in the *Total size* plot, which shows us the aforementioned 90% of reassembled packets to be larger than 1500 bytes in size.

F. Breakdown of hidden ICMP6 traffic

Mostly behind Hop-by-Hop Options, ICMP6 is the second-most observed hidden upper layer protocol in both networks. In CESNET, the share of ICMP6 flows behind HBH-options is 40.9% (Table VI). Additionally, 501 flows with HBH-options also included a Fragmentation Header, followed by the actual ICMP6 payload. In UTNET, the lower 7.7% still forms a significant part of the hidden traffic in terms of flows, however it only accounts for 0.1% of packets.

TABLE VII: CESNET: hidden ICMP6 types and codes

EHs	Type	Code	Description	Packets
CESNET:				
0;	131	0	Multicast Listener Report	8.0M
0;	143	0	Version 2 Multicast Listener Report	2.1M
44;	129	0	Echo Reply	154.7K
0;	130	0	Multicast Listener Query	125.0K
0;	135	0	Neighbor Solicitation	108.8K
0;	4	1	Parameter Problem	24.4K
0;44;	3	1	Time Exceeded	12.9K
0;	1	4	Destination Unreachable	4.8K
44;	128	0	Echo Request	3.8K
44;	3	0	Time Exceeded	33.0
0;	132	0	Multicast Listener Done	29.0
44;	1	0	Destination Unreachable	4.0
0;	128	0	Echo Request	1.0
43;	161	13	161	1.0
UTNET:				
0;	130	0	Multicast Listener Query	86.1K
0;	143	0	Version 2 Multicast Listener Report	85.6K
44;	129	0	Echo Reply	21.5K
44;	128	0	Echo Request	928.0
44;	3	0	Time Exceeded	36.0

We analyzed the ICMP types and codes per the (one or multiple) EHs. The overviews of these numbers are given in Table VII. Due to the nature of ICMP, *i.e.* facilitating control rather than transport, we only show packet counts in the tables.

In both networks, most ICMP packets are multicast-related, all preceded by HBH-options. Fragmented ICMP consists of Echo Replies (*pongs*) mostly, and in UTNET accounts for 11.1% of all ICMP with EHs.

G. Analysis of EH misuse

As described in § II, there is a plethora of known, possible misuses based on EHs. While it is not in all cases possible to classify traffic as benign or malicious based on the collected flow data, we did observe several cases that are at least suspicious.

1) Abnormally large Extension Headers: On CESNET, we measured 179 flows to contain EHs with a size larger than 256 bytes: of these, 86 were larger than 1280 bytes, and 74 even exceeded 1460 bytes. Adding the 40 bytes of the IPv6 header itself, those packets would fill 1500 bytes without counting upper layer payload (if any). Comparing the size of the packets to the number of bytes specified in the Length field of the EHs, 158 of these flows show a difference of more than 56 bytes (*i.e.* the IPv6 header size of 40 bytes, plus two times 8 bytes for two EHs): the largest difference is over 2000 bytes.

Naturally, extracting upper layer payload information from these packets is not feasible: the (too) large EH Lengths point to an upper layer offset that is outside of the actual packet, hence one can not use it to find actual upper layers. Whether these packets were malformed in transit or purposely constructed by the source, can not be concluded from this data. Like our measurement appliance, forwarding devices and (security-related) middleboxes that parse the EHs will encounter the same problem, and need to make a decision on whether to forward or to drop the traffic.

2) Double fragmentation headers: As shown in Table V, we observed three flows in CESNET that contained multiple Fragmentation Headers. All of the 21 packets in these flows originated from the same IPv6 address, from TCP source port 80 — likely HTTP traffic. All packets were sent to a single IPv6 destination address, though to three different ports (and therefore split into three different flow records). The mean packet size in these flows was 76 bytes. This small size does not justify fragmentation, and containing two fragmentation headers hints at either an evasion attempt, or a network stack in an erroneous state.

V. Conclusions

Dropping all packets that contain Extension Headers is a bad practice. Our measurements show that a significant share of the IPv6 traffic contains one EH, carrying payloads crucial for both operators (in the case of ICMP6) or end-users (*e.g.* fragmented DNS responses). Discarding this traffic leads to unpleasant surprises that are not trivial to troubleshoot.

The share of traffic containing more than one EH however, is very small. For the design of hardware able to handle the dynamic nature of EHs, we therefore recommend to support at least one EH: the exceptional packets containing more EHs can be handled in the *slow-path*, *i.e.* a slower CPU in the network device, without substantial performance loss, while still offering flexibility to drop packets with *e.g.* more than three EHs to prevent the possible Denial of Service attack. Choosing to simply drop packets with more than one EH still impairs end-user experiences, *e.g.* in the case of fragmented IPSEC ESP, and is therefore not recommended.

Threats based on Extension Headers become visible when adapting your flow monitoring to traverse and account for the EH-chain. While no extraordinarily long chains were observed, we did measure suspiciously sized EHs. The presented enhancements in flow measurements enable security officers to filter out suspicious traffic easily, and conduct further analysis.

Measuring hidden traffic, *e.g.* TCP or UDP preceded by EHs, does not only reveal which (end-user) services are used, as aforementioned: it also aids in spotting possible middlebox evasion. Querying flow-data for suspicious traffic towards critical services is easy, since one can filter on the EH count and service ports, *e.g.* TCP destination port 22 for SSH.

Finally, the two measured networks should not be considered representative for the entire Internet: the different results emphasize the need for adequately adapted flow measurement tools. Different types of networks carry a different collection and distribution of Extension Headers, or have their own specific configuration that might result in traffic considered suspicious in other networks. Deploying adapted measurement and monitoring solutions is a necessary first step to inventory what actually is hidden behind the Extension Headers.

Future work: Some of our findings raised questions that we hope to answer in future research. Particularly, we are interested in the cause(s) of the observed fragmented TCP traffic.

Furthermore, enhancing the measurement setup with the option to fully capture packets of suspicious flows (*e.g.* abnormally large Extension Headers) should help us understand the true nature of these packets. Both these works should provide insights for operators, helping them to identify (and distinguish) both misconfigurations and misuse in their networks.

Acknowledgments: This work is partially supported by SURFNet's Research on Networking (RoN) project, and project Reg. No. CZ.02.1.01/0.0/0.0/16_013/0001797 co-funded by the Ministry of Education, Youth and Sports of the Czech Republic and European Regional Development Fund.

REFERENCES

[1] F. Gont, S. Networks, J. Linkova, Google, T. Chown, Jisc, W. Liu, and H. Technologies, "RFC 7872: Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World," 2016.

[2] S. Deering, Cisco, R. Hinden, and Nokia, "RFC 2460: Internet Protocol, Version 6 (IPv6) Specification," 1998.

[3] "Assigned Internet Protocol Numbers," http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml.

[4] J. Abley, Afilias, P. Savola, CSC/FUNET, and G. Neville-Neil, "RFC 5095: Deprecation of Type 0 Routing Headers in IPv6," 2007.

[5] A. Atlasis, "The Impact of Extension Headers on IPv6 Access Control Lists Real Life Use Cases," Heidelberg, Germany, 2016.

[6] F. Gont, V. Manral, and R. Bonica, "RFC 7112: Implications of Oversized IPv6 Header Chains," 2014.

[7] "IP Flow Information Export (IPFIX) Entities," http://www.iana.org/assignments/ipfix/ipfix.xhtml.

[8] R. Hofstede, P. Čeleda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras, "Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2037–2064, 2014.

[9] B. Trammell and E. Boschi, "An introduction to IP flow information export (IPFIX)," *IEEE Communications Magazine*, vol. 49, no. 4, 2011.

[10] F. Gont, N. Hilliard, G. Doering, W. Liu, and W. Kumari, "Operational Implications of IPv6 Packets with Extension Headers," https://tools.ietf.org/id/draft-gont-v6ops-ipv6-ehs-packet-drops-03.txt, 2016.

[11] M. Elich, M. Grégr, and P. Čeleda, "Monitoring of tunneled ipv6 traffic using packet decapsulation and ipfix (short paper)," in *International Workshop on Traffic Monitoring and Analysis*. Springer, 2011, pp. 64–71.

[12] J. Ullrich, K. Krombholz, H. Hobel, A. Dabrowski, and E. R. Weippl, "IPv6 Security: Attacks and Countermeasures in a Nutshell," in *USENIX WOOT*, 2014.

[13] G. Van Den Broek, R. van Rijswijk-Deij, A. Sperotto, and A. Pras, "DNSSEC meets real world: dealing with unreachability caused by fragmentation," *IEEE communications magazine*, vol. 52, no. 4, pp. 154–160, 2014.

[14] O. Tange, "GNU Parallel - The Command-Line Power Tool," *;login: The USENIX Magazine*, vol. 36, no. 1, pp. 42–47, Feb 2011. [Online]. Available: http://www.gnu.org/s/parallel