black hat REGIONAL SUMMIT SAO PAULO 2014

ERNW providing security.

Evasion of High-End IDPS Devices at the IPv6 Era

Antonios Atlasis

secfu.net
aatlasis@secfu.net

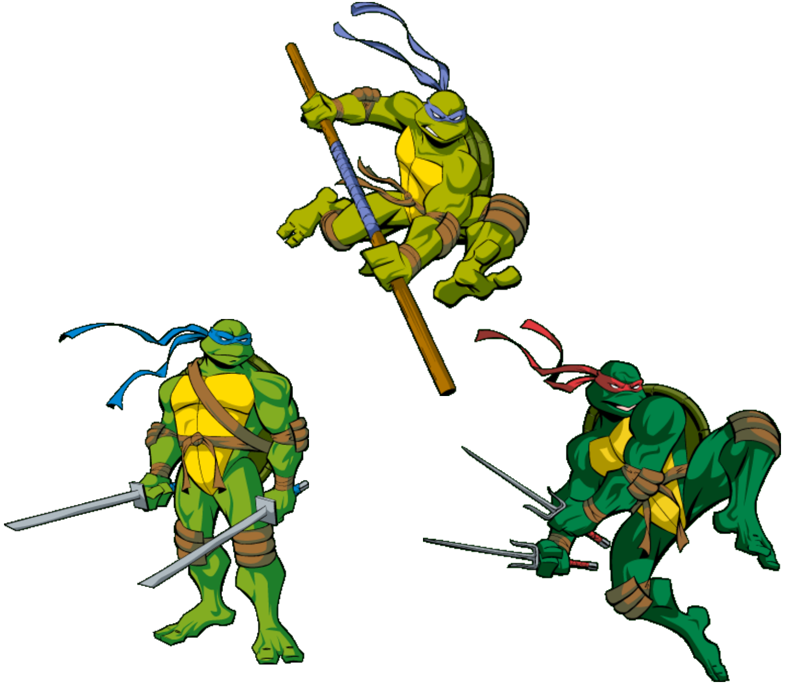
Enno Rey

ERNW GmbH
erey@ernw.de

Rafael Schaefer

ERNW GmbH
rschaefer@ernw.de

Who We Are

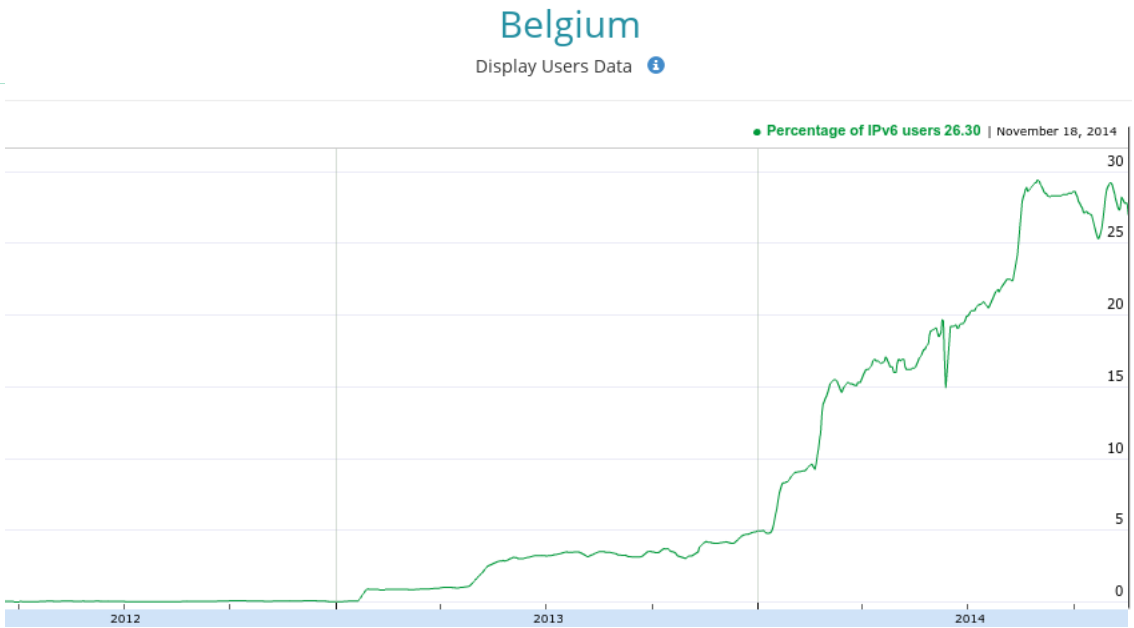


- Enno Rey
  - Old school network security guy. Back in 2001 founder of ERNW & still proudly running the team.
- Antonios Atlasis
  - IT Security enthusiast.
  - Researching security issues for fun.
- Rafael Schaefer
  - ERNW student
  - Young researcher

Outline of the Presentation

- Introduction
  - IPv6 is here
  - What IPv6 brings with it:
    The Extension Headers
- Problem Statement. Describe the Mess
- Tested IDPS devices:
  - Suricata
  - Tipping Point
  - Sourcefire
  - Snort
- Mitigation & Conclusions

IPv6 is Real



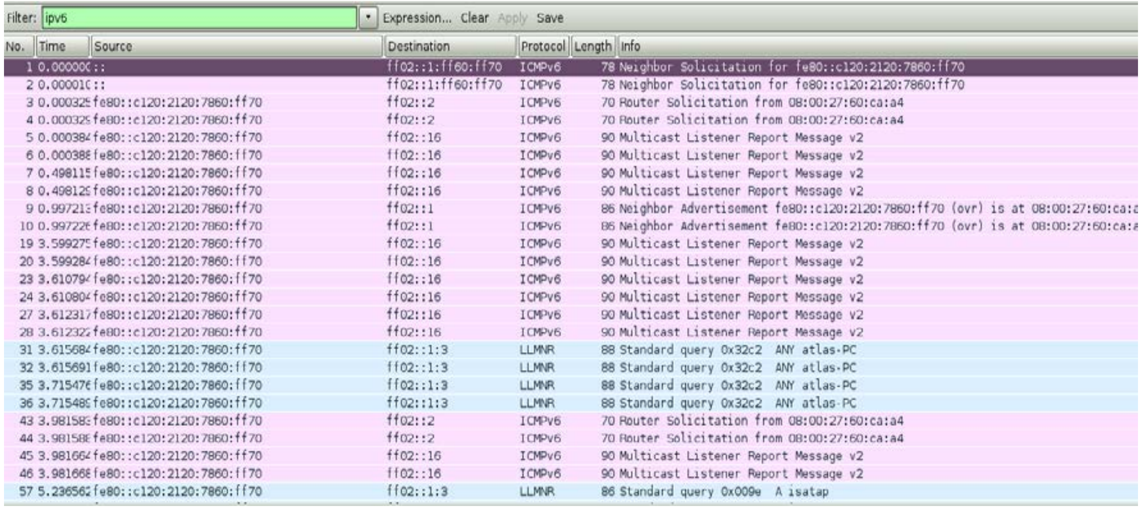
Belgium

Display Users Data

- The trend in other European countries is similar.

But I don’t Use it in my Environment

- 1) Default Behaviour of Windows 7 Service Pack 1
- 2) Without IPv6 Router in the environment
- 3) These are just a small portion :)



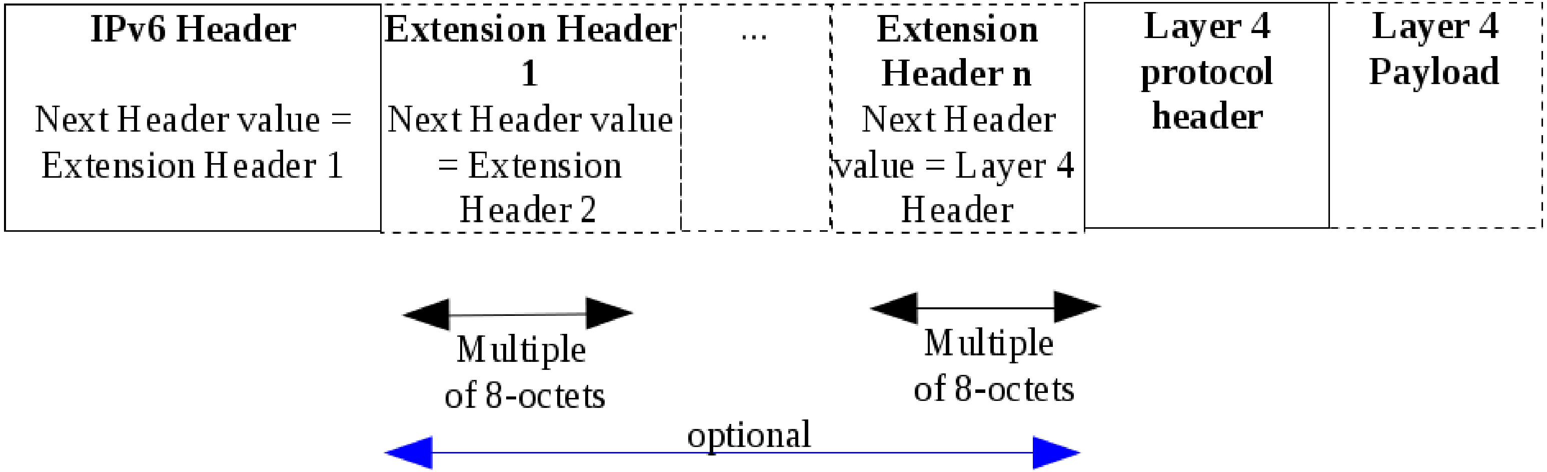
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	::	ff02::1:ff60:ff70	ICMPv6	78	Neighbor Solicitation for fe80::c120:2120:7860:ff70
2	0.000010	::	ff02::1:ff60:ff70	ICMPv6	78	Neighbor Solicitation for fe80::c120:2120:7860:ff70
3	0.000325	fe80::c120:2120:7860:ff70	ff02::2	ICMPv6	70	Router Solicitation from 08:00:27:60:ca:a4
4	0.000325	fe80::c120:2120:7860:ff70	ff02::2	ICMPv6	70	Router Solicitation from 08:00:27:60:ca:a4
5	0.000384	fe80::c120:2120:7860:ff70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
6	0.000388	fe80::c120:2120:7860:ff70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
7	0.498115	fe80::c120:2120:7860:ff70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
8	0.498129	fe80::c120:2120:7860:ff70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
9	0.997213	fe80::c120:2120:7860:ff70	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::c120:2120:7860:ff70 (ovr) is at 08:00:27:60:ca:a
10	0.997226	fe80::c120:2120:7860:ff70	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::c120:2120:7860:ff70 (ovr) is at 08:00:27:60:ca:a
19	3.599275	fe80::c120:2120:7860:ff70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
20	3.599284	fe80::c120:2120:7860:ff70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
23	3.610794	fe80::c120:2120:7860:ff70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
24	3.610804	fe80::c120:2120:7860:ff70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
27	3.612317	fe80::c120:2120:7860:ff70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
28	3.612322	fe80::c120:2120:7860:ff70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
31	3.615684	fe80::c120:2120:7860:ff70	ff02::1:3	LLMNR	88	Standard query 0x32c2 ANY atlas-PC
32	3.615691	fe80::c120:2120:7860:ff70	ff02::1:3	LLMNR	88	Standard query 0x32c2 ANY atlas-PC
35	3.715476	fe80::c120:2120:7860:ff70	ff02::1:3	LLMNR	88	Standard query 0x32c2 ANY atlas-PC
36	3.715489	fe80::c120:2120:7860:ff70	ff02::1:3	LLMNR	88	Standard query 0x32c2 ANY atlas-PC
43	3.981583	fe80::c120:2120:7860:ff70	ff02::2	ICMPv6	70	Router Solicitation from 08:00:27:60:ca:a4
44	3.981588	fe80::c120:2120:7860:ff70	ff02::2	ICMPv6	70	Router Solicitation from 08:00:27:60:ca:a4
45	3.981664	fe80::c120:2120:7860:ff70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
46	3.981668	fe80::c120:2120:7860:ff70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
57	5.236562	fe80::c120:2120:7860:ff70	ff02::1:3	LLMNR	86	Standard query 0x009e A isatap

Still, what is the big deal?

- Just an IPv4 replacement with huge address space, correct?
- Many things have changed, for good (??).
- IPv6 Extension Headers probably being the most devastating!

What an IPv6 Datagrams Looks Like…

What an IPv6 Datagrams Looks Like…



IPv6 Header

Next Header value = Extension Header 1

Extension Header 1

Next Header value = Extension Header 2

…

Extension Header n

Next Header value = Layer 4 Header

Layer 4 protocol header

Layer 4 Payload

Multiple of 8-octets

Multiple of 8-octets

optional

- This is the root of 3 significant problems…

The IPv6 Extension Headers

- Currently defined:
  - Hop-by-Hop Options [RFC2460]
  - Routing [RFC2460]
  - Fragment [RFC2460]
  - Destination Options [RFC2460]
  - Authentication [RFC4302]
  - Encapsulating Security Payload [RFC4303]
  - MIPv6, [RFC6275] (Mobility Support in IPv6)
  - HIP, [RFC5201] (Host Identity Protocol)
  - shim6, [RFC5533] (Level 3 Multihoming Shim Protocol for IPv6)

- There is a RECOMMENDED order.
- All (but the Destination Options header) SHOULD occur at most once.
- How a device should react **if NOT** ?

Transmission & Processing of IPv6 Ext. Hdrs

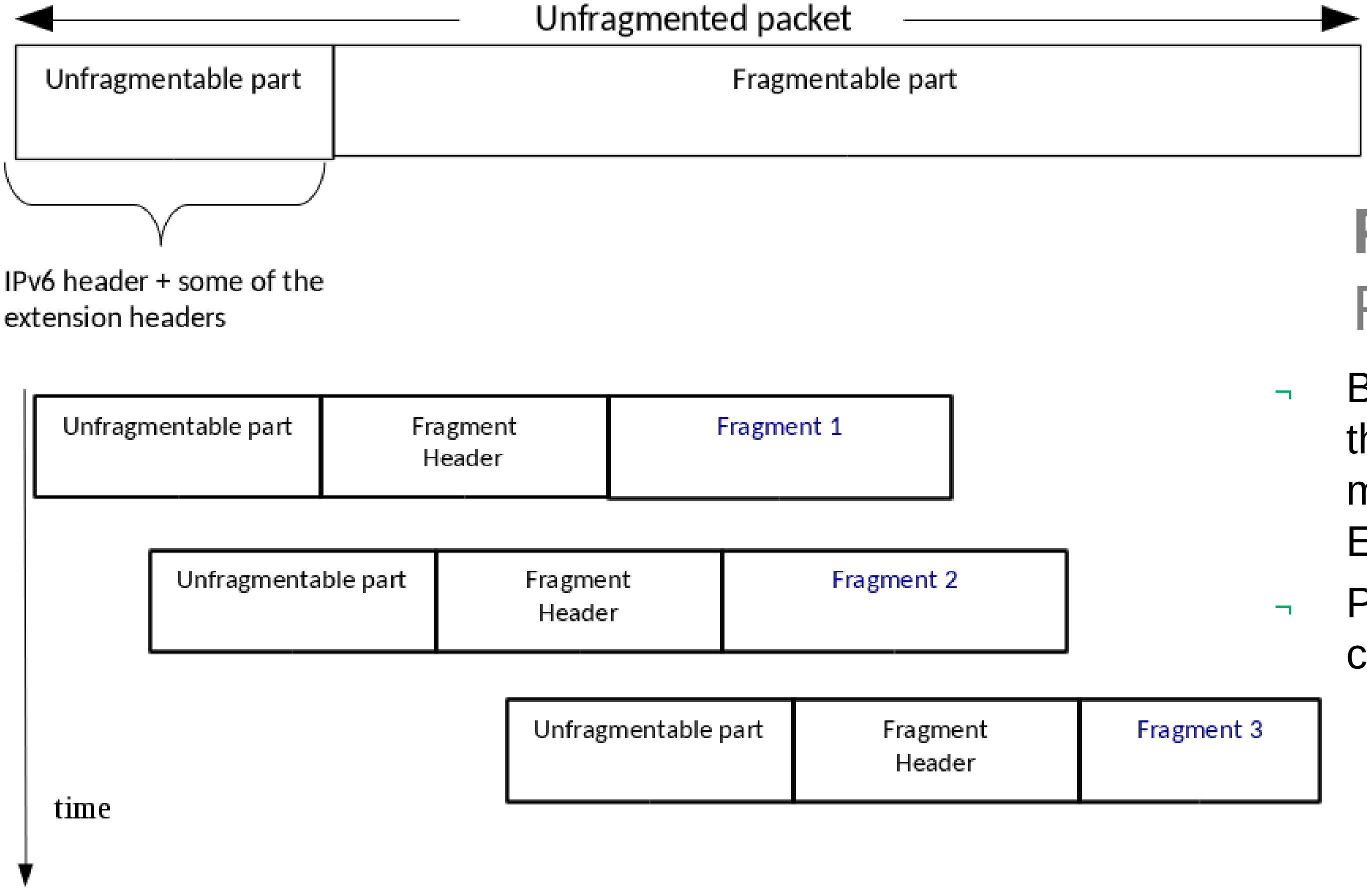
- RFC 7045. Any forwarding node along an IPv6 packet's path:
  - should forward the packet regardless of any extension headers that are present.
  - MUST recognize and deal appropriately with all standard IPv6 extension header types.
  - SHOULD NOT discard packets containing unrecognised extension headers.

Problem 1: Too Many Things to Vary

- Variable types
- Variable sizes
- Variable order
- Variable number of occurrences of each one.
- Variable fields

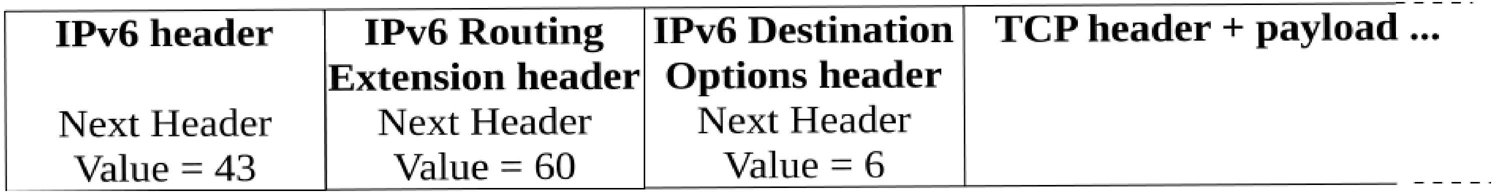
IPv6 = f(v,w,x,y,z,)

Problem 2: Fragmentation



- Both the *Fragmentable* and the *Unfragmentable* parts may contain any IPv6 Extension headers.
- Problem 1 becomes more complicated.

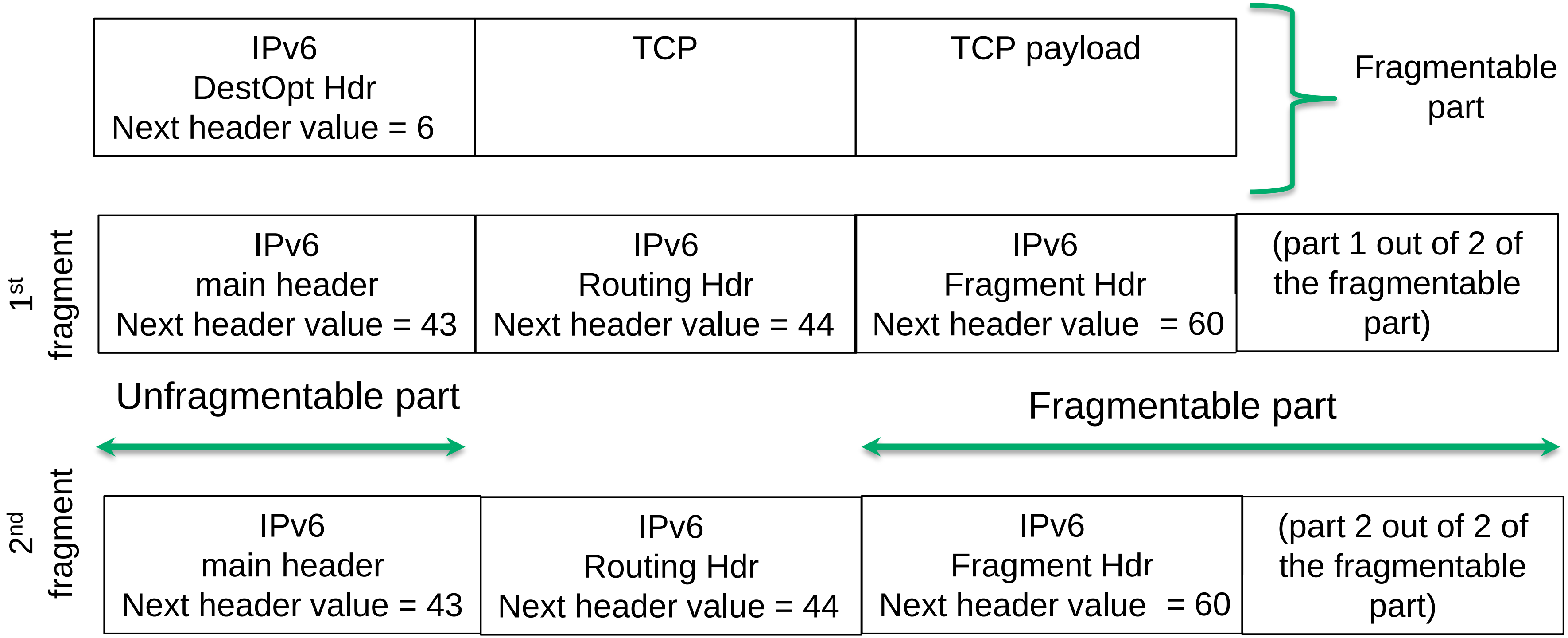
Problem 3: How IPv6 Extension Headers are Chained?



IPv6 header	IPv6 Routing Extension header	IPv6 Destination Options header	TCP header + payload ...
Next Header Value = 43	Next Header Value = 60	Next Header Value = 6	

- Next header fields:
  - Contained in IPv6 headers, identify the type of header immediately following the current one.
  - They use the same values as the IPv4 Protocol field.

Why IPv6 Header Chaining is a Problem?



IPv6
DestOpt Hdr
Next header value = 6

TCP

TCP payload

Fragmentable
part

1st
fragment

IPv6
main header
Next header value = 43

IPv6
Routing Hdr
Next header value = 44

IPv6
Fragment Hdr
Next header value = 60

(part 1 out of 2 of
the fragmentable
part)

Unfragmentable part

Fragmentable part

2nd
fragment

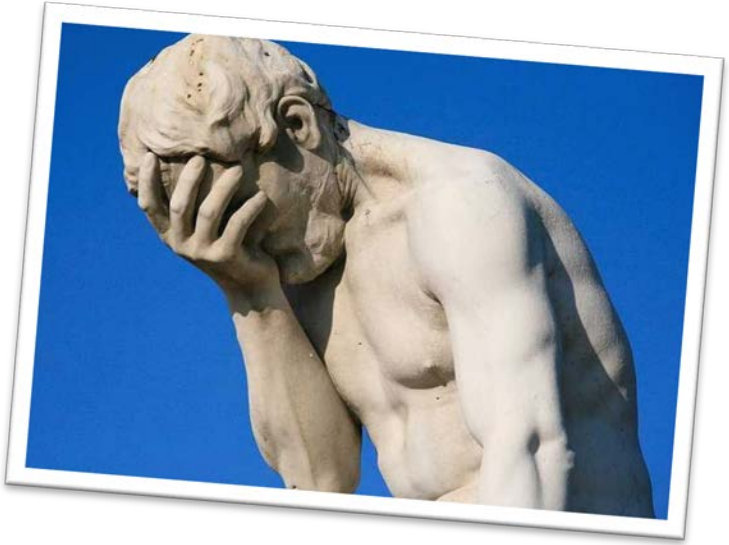
IPv6
main header
Next header value = 43

IPv6
Routing Hdr
Next header value = 44

IPv6
Fragment Hdr
Next header value = 60

(part 2 out of 2 of
the fragmentable
part)

To sum up the Mess in IPv6



- Vary:
  - The types of the IPv6 Extension headers
  - The order of the IPv6 Extension headers
  - The number of their occurrences.
  - Their size.
  - Their fields.
  - The Next Header values of the IPv6 Fragment Extension headers in each fragment.
  - Fragmentation (where to split the datagram)
- And combine them.

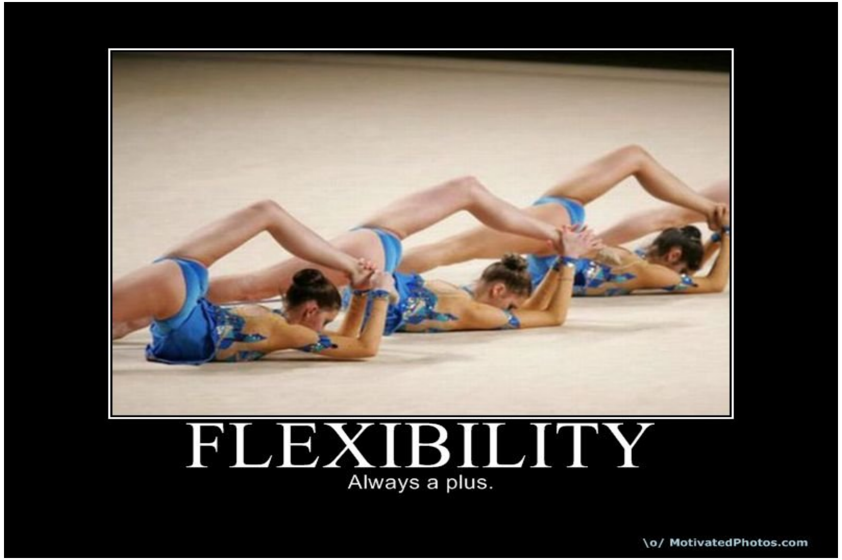
Did You Notice?

- When designing/writing IPv6 protocols & parsers they didn‘t pay too much attention to #LANGSEC.

- Please visit www.langsec.org.

We May Have a Fundamental Problem Here…

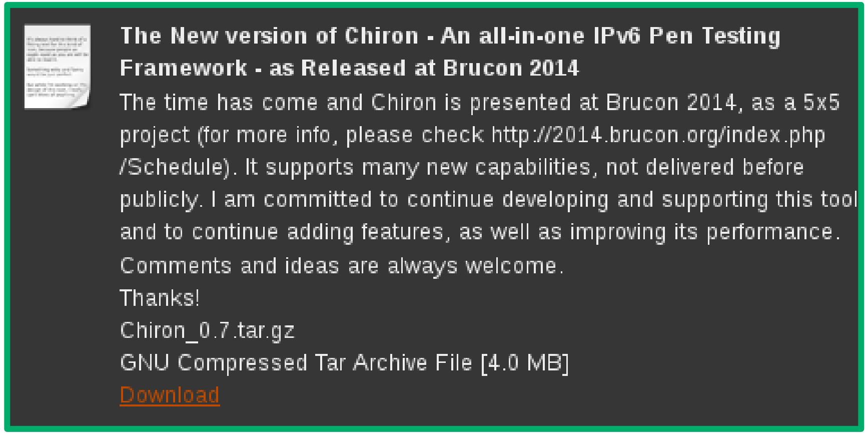
- There is too much flexibility and freedom…
- Which is usually inverse proportional to security :-)
- And it can potentially lead to a complete *cha0s*…



So, What Can Possibly Go Wrong?

- Detection Signatures, e.g. used by IDPS rules, etc. are based on blacklisting traffic.
- What if we confuse their parsers by abusing IPv6 Extension headers in an unusual / unexpected way?

All this is not just a theory



The New version of Chiron - An all-in-one IPv6 Pen Testing Framework - as Released at Brucon 2014

The time has come and Chiron is presented at Brucon 2014, as a 5x5 project (for more info, please check http://2014.brucon.org/index.php/Schedule). It supports many new capabilities, not delivered before publicly. I am committed to continue developing and supporting this tool and to continue adding features, as well as improving its performance.

Comments and ideas are always welcome.

Thanks!

Chiron_0.7.tar.gz

GNU Compressed Tar Archive File [4.0 MB]

Download

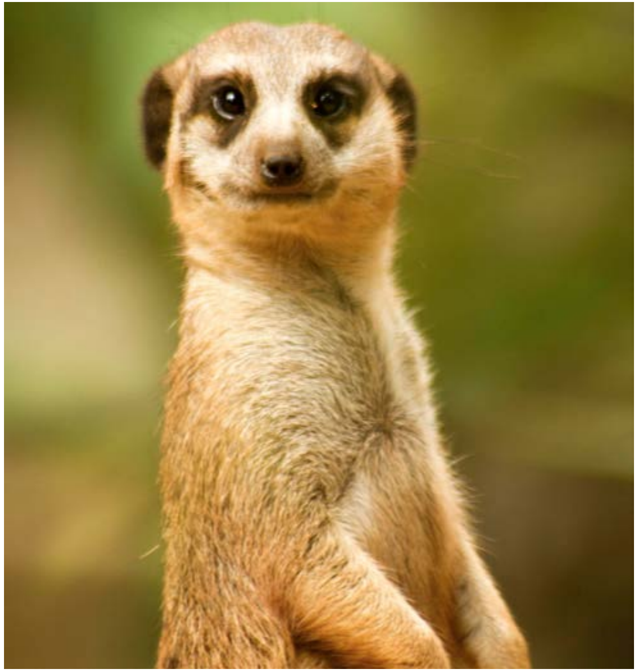
- You can reproduce all the results that we shall demonstrate using *Chiron*

- It can be downloaded from:
  http://www.secfu.net/tools-scripts/

Our Tests at a Glance

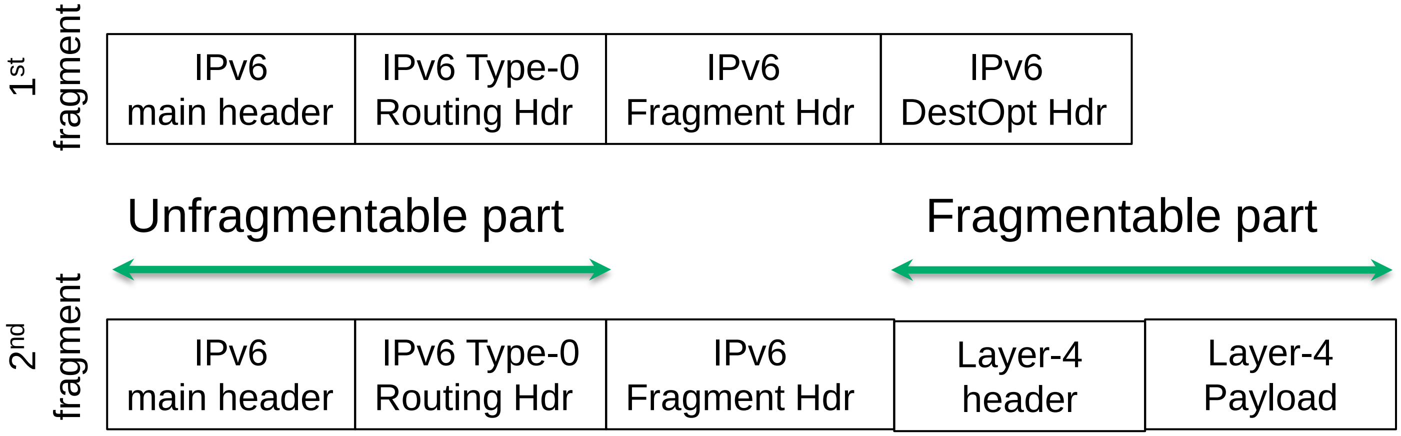
- Four (4) IDPS (two open-source, two high-end commercial ones).
- At least twelve (12) different evasion techniques, in total.
- All of them 0-days at the time of the finding.
- All of them were reported (disclosed responsibly).
- Most of them were patched, either promptly or not that promptly ☺.
- Some guys were too busy though, so two of the products still suffer from 0-days IPv6 evasion techniques.

Evading Suricata



- Versions 2.0.1, 2.0.2 and 2.0.3 were evaded one by one by using various techniques
- All of them can be found in the white paper and can be reproduced by using *Chiron*.
- We will demonstrate the latest one.

Evading Suricata 2.0.3

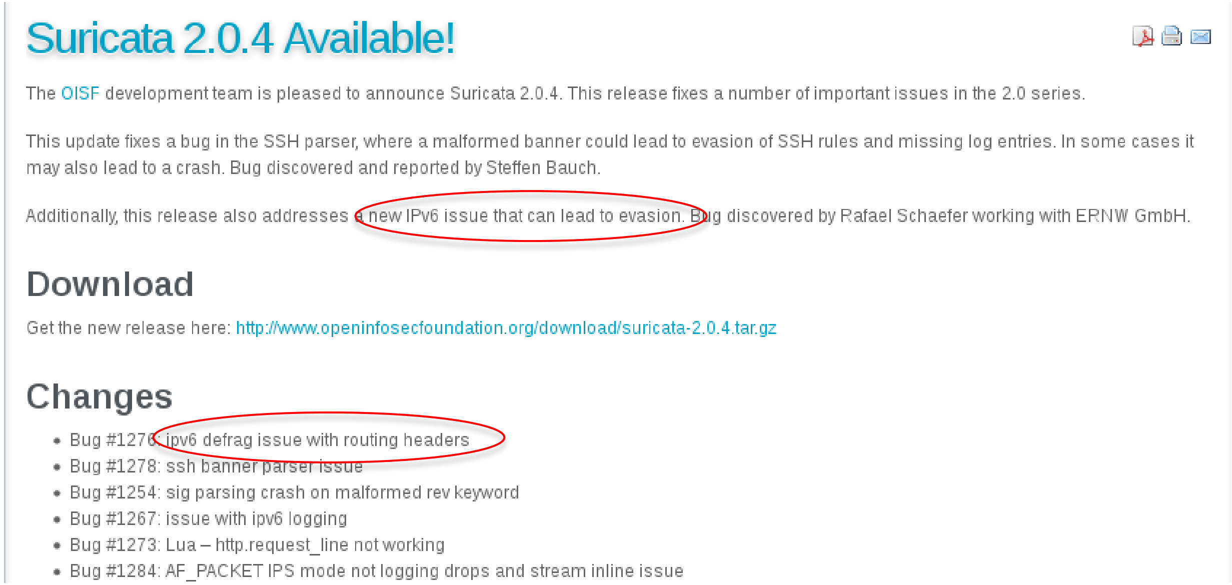


Note: Other combinations of Extension Headers can also work (your …homework)

Time for Action

- Demo against Suricata 2.0.3

Suricata Developers in Each Reported Case Reacted really Fast



Suricata 2.0.4 Available!

The OISF development team is pleased to announce Suricata 2.0.4. This release fixes a number of important issues in the 2.0 series.

This update fixes a bug in the SSH parser, where a malformed banner could lead to evasion of SSH rules and missing log entries. In some cases it may also lead to a crash. Bug discovered and reported by Steffen Bauch.

Additionally, this release also addresses a new IPv6 issue that can lead to evasion. Bug discovered by Rafael Schaefer working with ERNW GmbH.

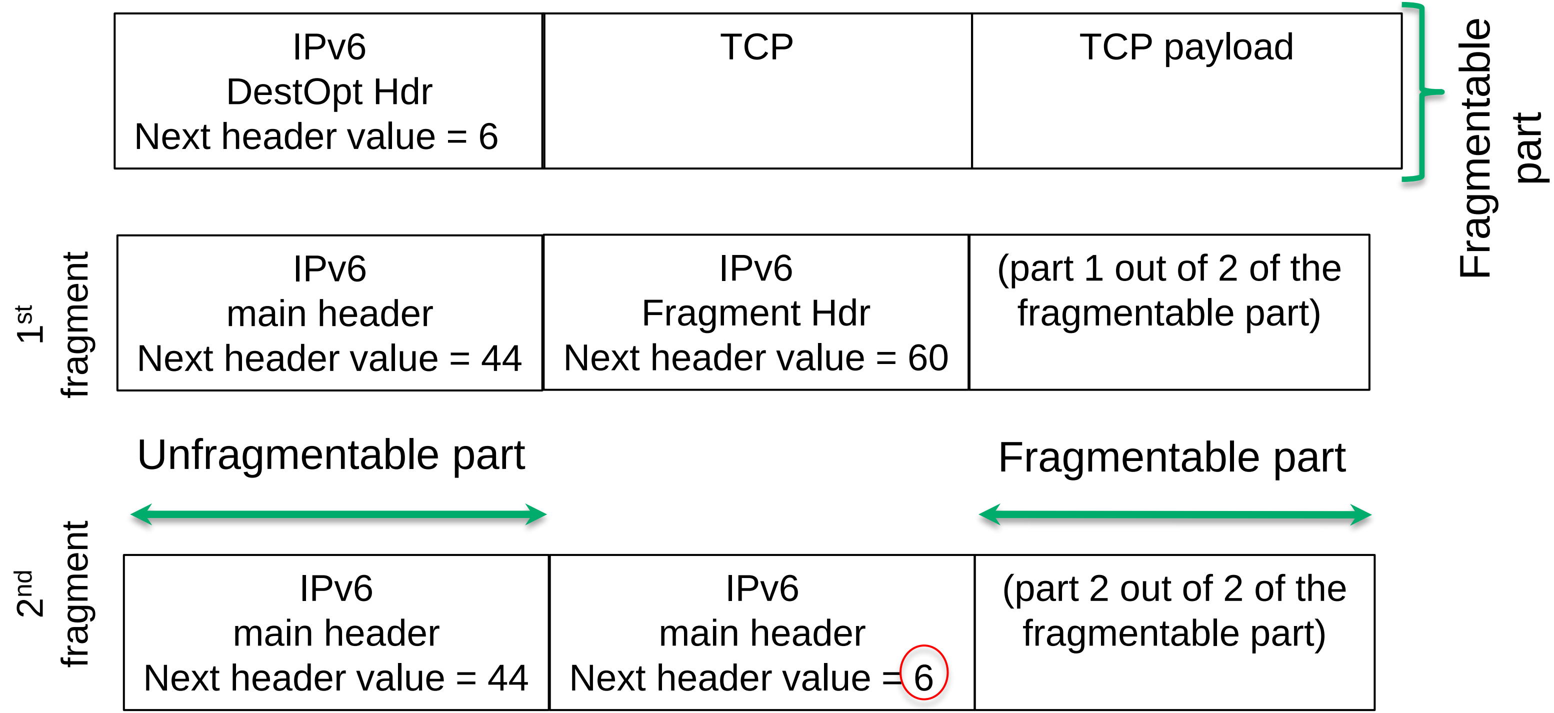
Download

Get the new release here: http://www.openinfosecfoundation.org/download/suricata-2.0.4.tar.gz

Changes

- Bug #1276: ipv6 defrag issue with routing headers
- Bug #1278: ssh banner parser issue
- Bug #1254: sig parsing crash on malformed rev keyword
- Bug #1267: issue with ipv6 logging
- Bug #1273: Lua – http.request_line not working
- Bug #1284: AF_PACKET IPS mode not logging drops and stream inline issue

Evading TippingPoint, “the Old Way” (March 2014)



IPv6
DestOpt Hdr
Next header value = 6

TCP

TCP payload

Fragmentable part

1st fragment

IPv6
main header
Next header value = 44

IPv6
Fragment Hdr
Next header value = 60

(part 1 out of 2 of the fragmentable part)

Unfragmentable part

Fragmentable part

2nd fragment

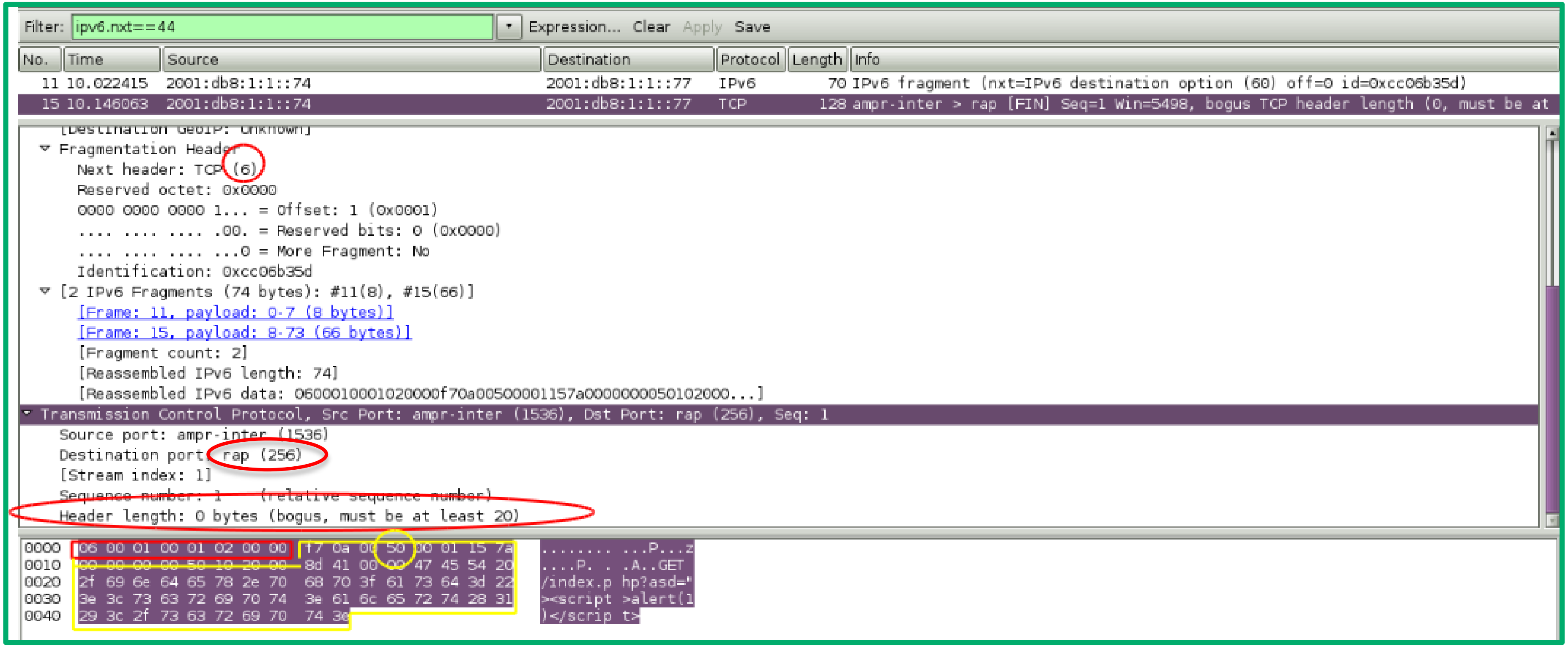
IPv6
main header
Next header value = 44

IPv6
main header
Next header value = 6

(part 2 out of 2 of the fragmentable part)

<u>Note</u>: Layer-4 header can be in the 1st fragment and the attack still works

Evading TippingPoint, “The Old Way”



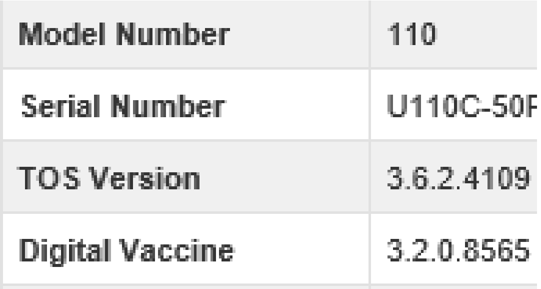
```
No. Time       Source             Destination        Protocol Length Info
 11 10.022415  2001:db8:1:1::74   2001:db8:1:1::77   IPv6         70 IPv6 fragment (nxt=IPv6 destination option (60) off=0 id=0xcc06b35d)
 15 10.146063  2001:db8:1:1::74   2001:db8:1:1::77   TCP         128 ampr-inter > rap [FIN] Seq=1 Win=5498, bogus TCP header length (0, must be at
```

```
    [Destination GeoIP: Unknown]
  ▽ Fragmentation Header
      Next header: TCP (6)
      Reserved octet: 0x0000
      0000 0000 0000 1... = Offset: 1 (0x0001)
      .... .... .... .00. = Reserved bits: 0 (0x0000)
      .... .... .... ...0 = More Fragment: No
      Identification: 0xcc06b35d
  ▽ [2 IPv6 Fragments (74 bytes): #11(8), #15(66)]
      [Frame: 11, payload: 0-7 (8 bytes)]
      [Frame: 15, payload: 8-73 (66 bytes)]
      [Fragment count: 2]
      [Reassembled IPv6 length: 74]
      [Reassembled IPv6 data: 0600010001020000f70a00500001157a0000000050102000...]
▽ Transmission Control Protocol, Src Port: ampr-inter (1536), Dst Port: rap (256), Seq: 1
    Source port: ampr-inter (1536)
    Destination port: rap (256)
    [Stream index: 1]
    Sequence number: 1    (relative sequence number)
    Header length: 0 bytes (bogus, must be at least 20)
```

```
0000  06 00 01 00 01 02 00 00  f7 0a 00 50 00 01 15 7a   ........ ...P...z
0010  00 00 00 00 50 10 20 00  8d 41 00 00 47 45 54 20   ....P. . .A..GET 
0020  2f 69 6e 64 65 78 2e 70  68 70 3f 61 73 64 3d 22   /index.p hp?asd="
0030  3e 3c 73 63 72 69 70 74  3e 61 6c 65 72 74 28 31   ><script >alert(1
0040  29 3c 2f 73 63 72 69 70  74 3e                     )</scrip t>
```

That First One Was Patched…

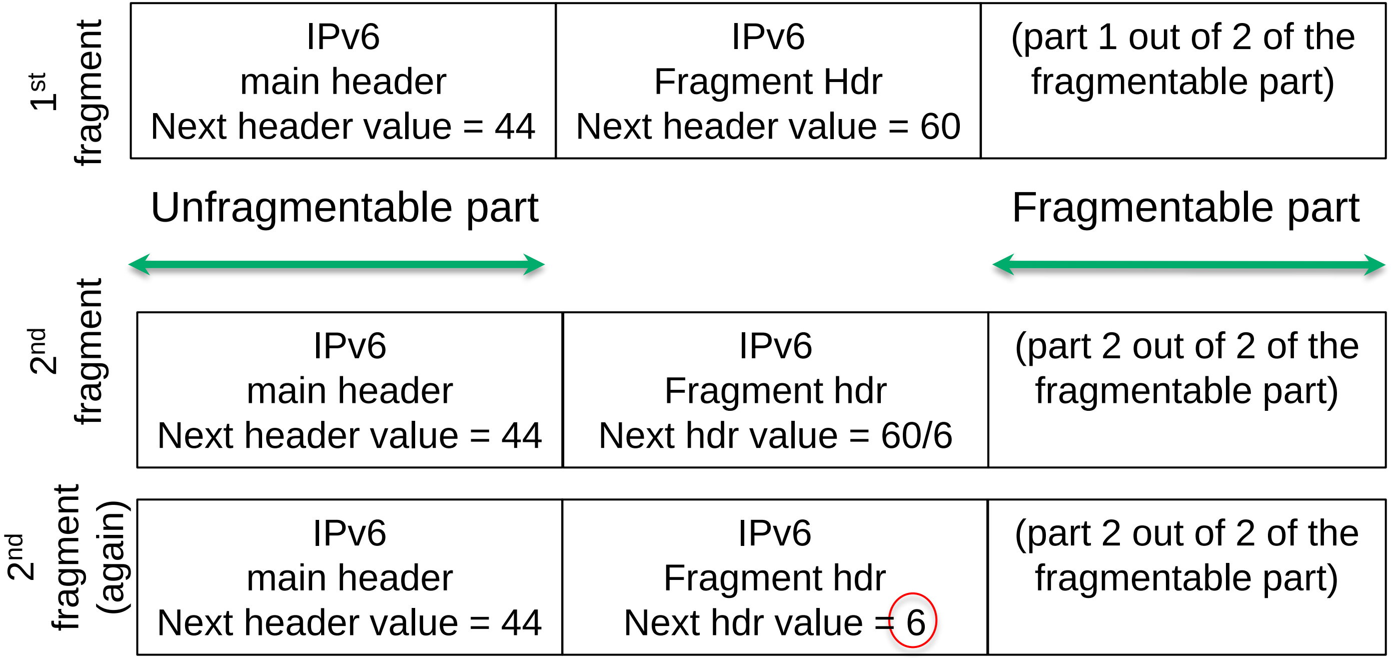
But Again We Had a New One ;-)



Model Number	110
Serial Number	U110C-50F
TOS Version	3.6.2.4109
Digital Vaccine	3.2.0.8565

- Configured to:
  - Operate inline at Layer 2.
  - Block any HTTP traffic.
  - Additional XSS rules (to test attacks at the payload too).

Evading TippingPoint, after First Patching



1st fragment	IPv6 main header Next header value = 44	IPv6 Fragment Hdr Next header value = 60	(part 1 out of 2 of the fragmentable part)
	Unfragmentable part		Fragmentable part
2nd fragment	IPv6 main header Next header value = 44	IPv6 Fragment hdr Next hdr value = 60/6	(part 2 out of 2 of the fragmentable part)
2nd fragment (again)	IPv6 main header Next header value = 44	IPv6 Fragment hdr Next hdr value = 6	(part 2 out of 2 of the fragmentable part)

Note: Layer-4 header can be in the 1st fragment and the attack still works

Snort / Sourcefire

- Quite similar situations, as expected.
- Still, the commercial device suffers from a 0-day evasion technique that the latest open-source version does not!

The Chronicle of the Communication

- We first contacted the Snort devs on 17th of June.
  - "Please, send us the pcap files"
  - We did; no news since then…
- Reported a Sourcefire issue on Sep 14, and Sep 25, etc., including pcap files.
  - A kind of "don't waste my time" approach.
  - "Please, contact the customer support…"

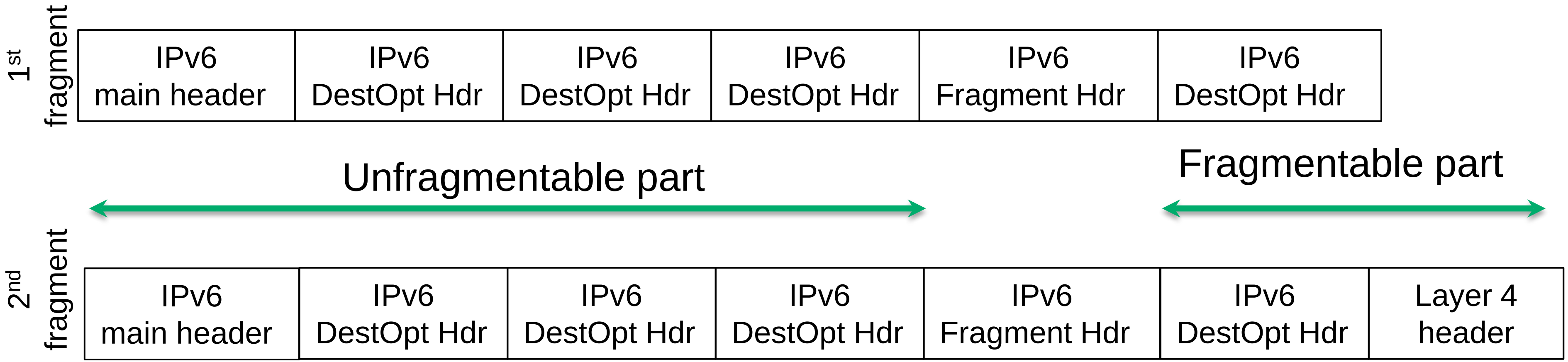
Fair enough!

- We did a full disclosure @BlackHat-EU and @HackLu!
- Demo time against SourceFire.

Evading Sourcefire

- Sourcefire, Model 3D7020 (81) Version 5.2.0.3 (Build 48).
- Preproc decoder rules were enabled:
  - GID 116 family and specifically, SID 458 (IPV6_BAD_FRAG_PKT), 272 and 273 are enabled.

Evading Sourcefire



1st fragment

IPv6 main header | IPv6 DestOpt Hdr | IPv6 DestOpt Hdr | IPv6 DestOpt Hdr | IPv6 Fragment Hdr | IPv6 DestOpt Hdr

Unfragmentable part

Fragmentable part

2nd fragment

IPv6 main header | IPv6 DestOpt Hdr | IPv6 DestOpt Hdr | IPv6 DestOpt Hdr | IPv6 Fragment Hdr | IPv6 DestOpt Hdr | Layer 4 header

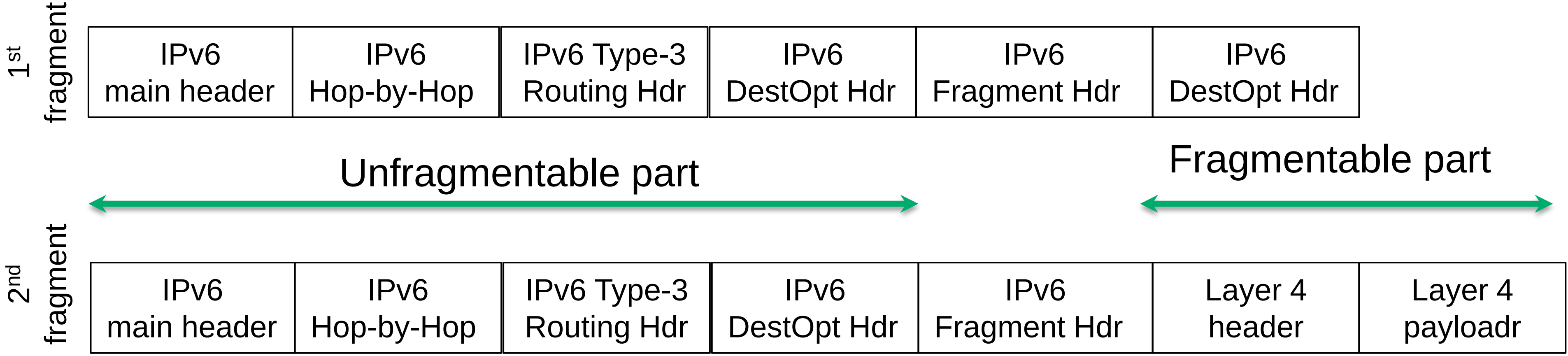
Note: Next header values for Fragment Extension headers: The correct ones (60)

Evading Snort

- Latest Snort version 2.9.6.2
- Preprocessor decoder rules are enabled:
  - GID 116 family and specifically, SID 458 (IPV6_BAD_FRAG_PKT), 272 and 273 are enabled.

DEMO

Evading Snort



1st fragment

IPv6 main header | IPv6 Hop-by-Hop | IPv6 Type-3 Routing Hdr | IPv6 DestOpt Hdr | IPv6 Fragment Hdr | IPv6 DestOpt Hdr

Unfragmentable part

Fragmentable part

2nd fragment

IPv6 main header | IPv6 Hop-by-Hop | IPv6 Type-3 Routing Hdr | IPv6 DestOpt Hdr | IPv6 Fragment Hdr | Layer 4 header | Layer 4 payloadr

Note: Next header values for Fragment Extension headers: the correct ones (60)

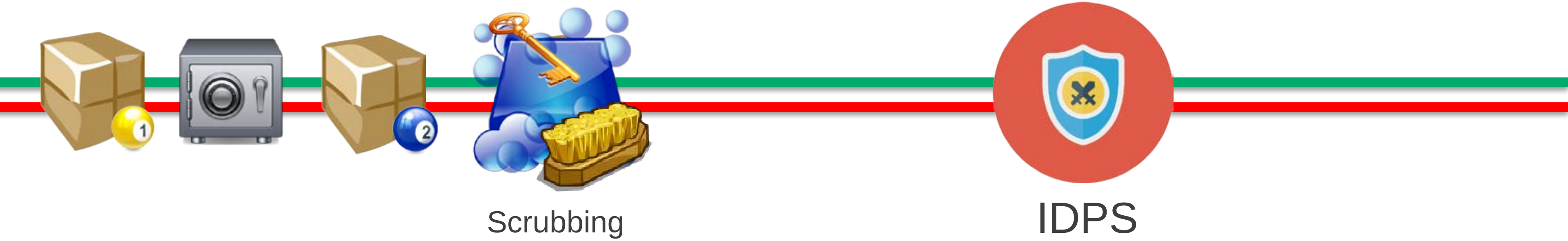
“Culture” Mitigations

- RFCs should strictly define the exact legitimate usage.
  - “Loose” specifications result in ambiguities and so they introduce potential attack vectors.
  - Functionality and flexibility are definitely good things, but security is non-negotiable.
- Make fully-compliant IPv6 products and test them thoroughly.

Technical Mitigations

- Implementation of RFC 7112.
  - An intermediate system (e.g., router or firewall) that receives an IPv6 First Fragment that does not include the entire IPv6 Header Chain MAY discard that packet.
  - Still, not a panacea…
- For the time being:
  - Configure your devices to drop IPv6 extension headers not used in your environment. OR
  - At least sanitize traffic before the IDPS.

In Case You still Want to Use an IDPS …



- you MUST (header-wise) scrub the traffic before entering the IDPS.

The Most Important “Take Away”

- These are just some of the IPv6 “grey areas”. Other may also exist.
  - Hint: MLD comes to mind…

- IPv6 security awareness.
  - Test it and use it, in your lab.
  - You will have to do it, sooner or later, anyway…

There's never enough time…

THANK YOU…

...for yours!

Tool & Slides:

https://www.insinuator.net

http://www.secfu.net/tools-scripts/

Questions?

- For more info check our white paper
- You can reach us at:✉
  - aatlasis@secfu.net, www.secfu.net
  - erey@ernw.de, www.insinuator.net
  - rschaefer@ernw.de
- Follow us at: 🐦
  - @AntoniosAtlasis
  - @Enno_Insinuator

There are few things to know about TROOPERS:

March, 16-20 2015
Heidelberg, Germany
Make the world a safer place.

REGISTRATION OPEN: www.troopers.de