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

ERNW providing security.

Who We Are

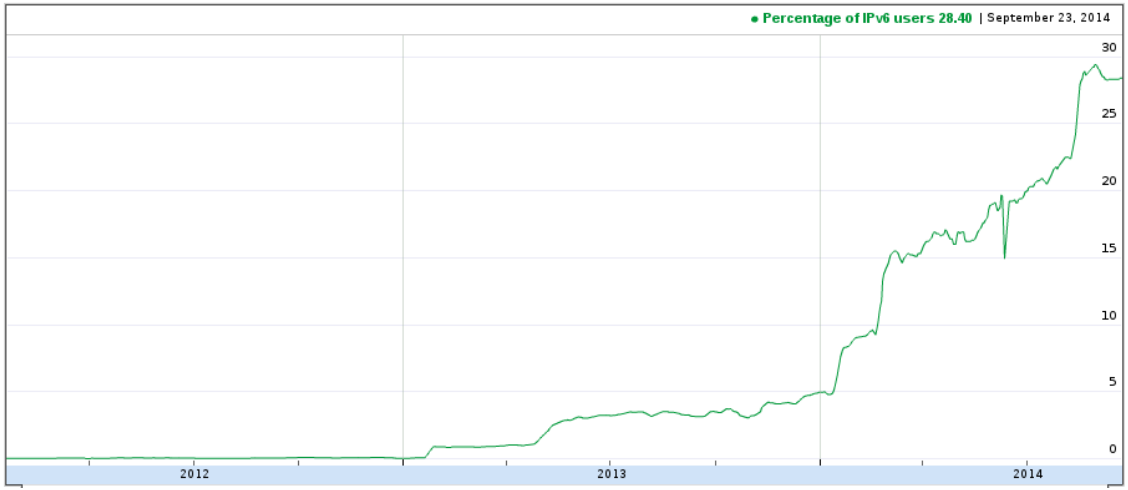
- Enno Rey
  - Old school network security guy. Back in 2001 founder of ERNW & still proudly running the team.
- Antonios Atlasis
  - IT Security enthusiast.
  - Researching security issues for fun.
- Rafael Schaefer
  - ERNW student
  - Young researcher

Outline of the Presentation

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

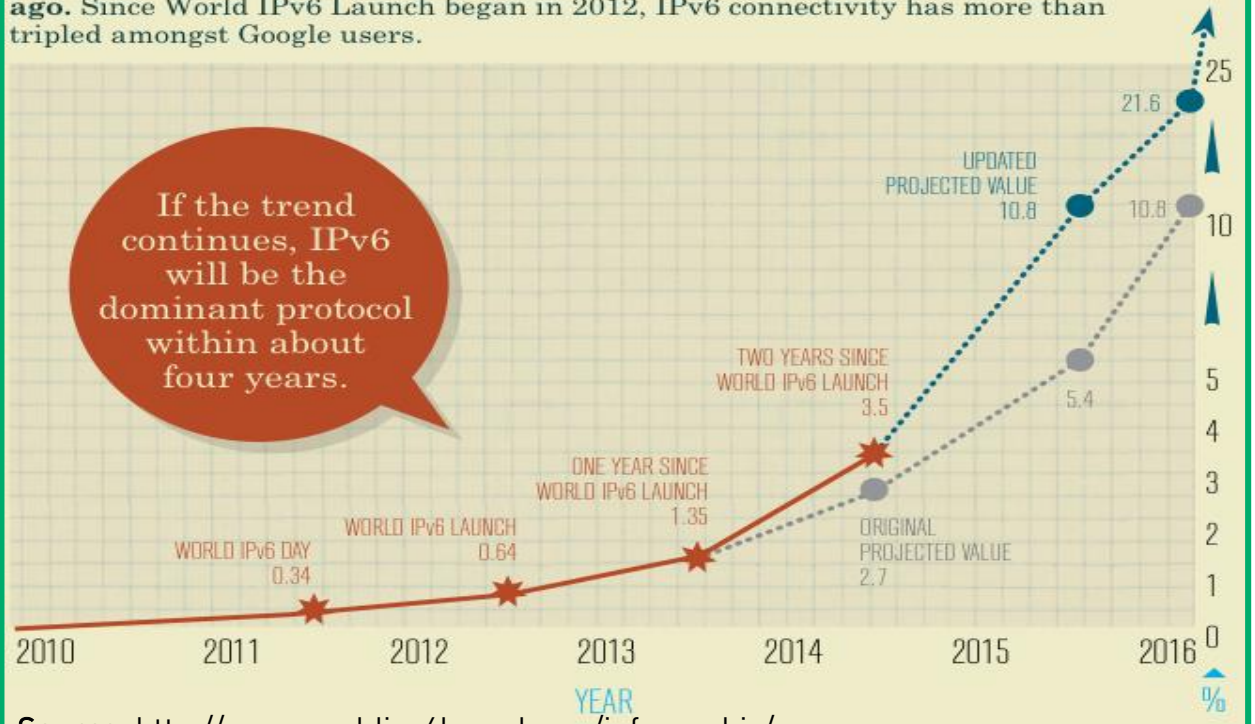
IPv4 Depletion

- The situation in other regions (including Europe) is similar (if not worse).



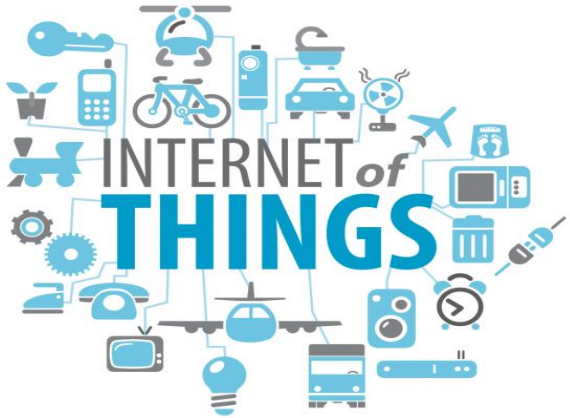
IPv6 MOMENTUM

IPv6 usage is increasing more rapidly than predicted just two years ago. Since World IPv6 Launch began in 2012, IPv6 connectivity has more than tripled amongst Google users.



Source: http://www.worldipv6launch.org/infographic/

- To make matters more urgent…



is coming.

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 :)

Filter: ipv6 Expression... Clear Apply Save

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	0.000329	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 has changed, for good (??)
- The IPv6 Extension Headers probably the most devastating!

What an IPv6 Datagrams Looks Like...

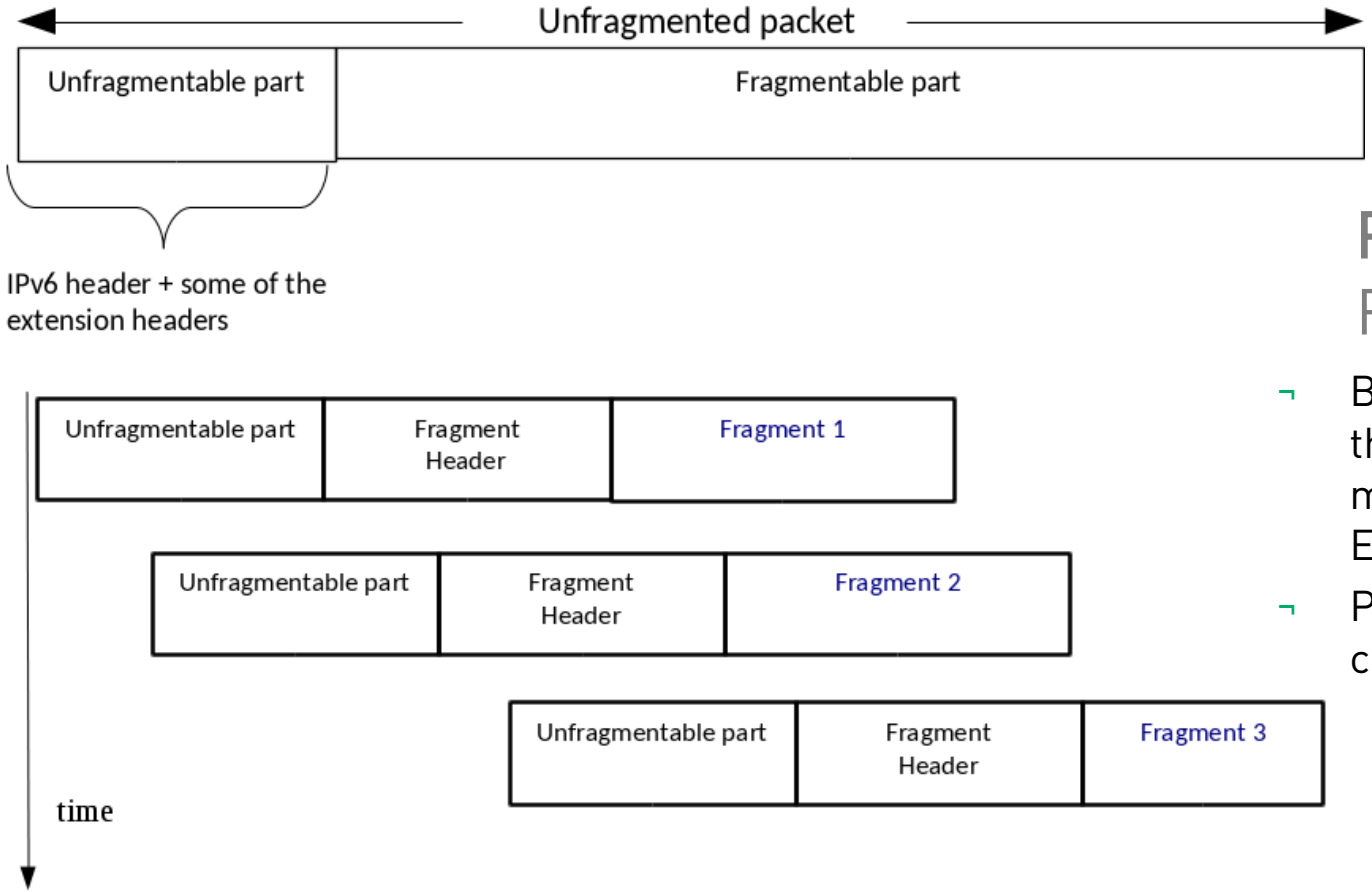
The IPv6 Extension Headers (RFC 2460)

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

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?

Fragmentable part

IPv6 Destination Options header Next Header Value = 6	TCP header + payload ...

Fragment 1

IPv6 header Next Header Value = 43	IPv6 Routing Extension header Next Header Value = 44	IPv6 Fragment Extension header Next Header Value	(part 1 out of 2 of the fragmentable part)

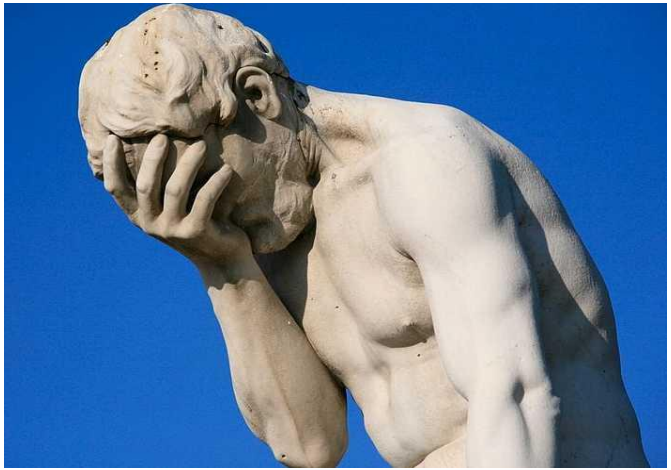
Fragment 2

IPv6 header Next Header Value = 43	IPv6 Routing Extension header Next Header Value = 44	IPv6 Fragment Extension header Next Header Value	(part 2 out of 2 of the fragmentable part)

Did You Notice?

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

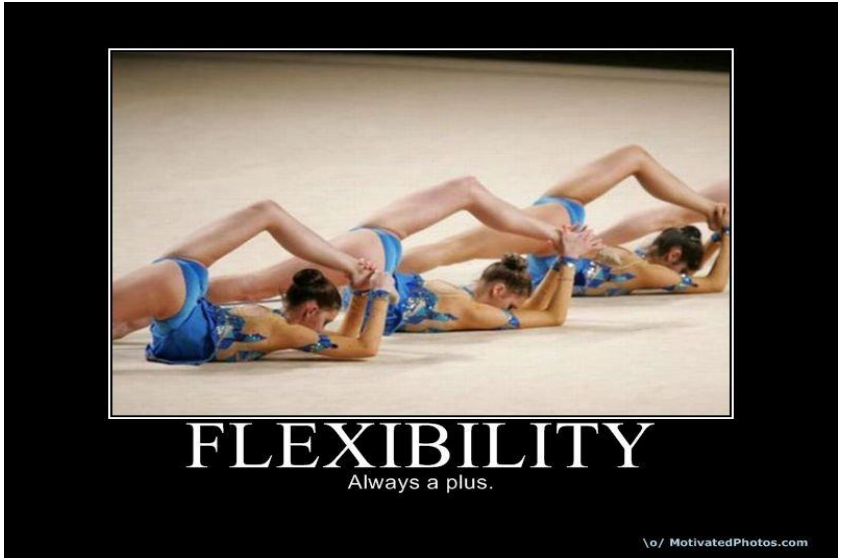
The Mess in IPv6



- 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.

We May Have a Fundamental Problem Here…

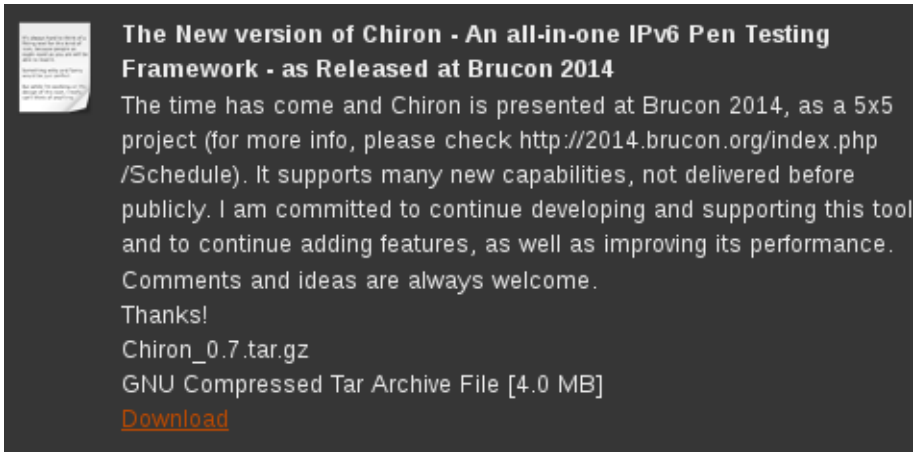
- There is too much flexibility and freedom…
- Which is usually inverse proportional to security :-)
- And it can potentially lead to a complete cha0s…



So, What Can Possibly Go Wrong With Them?

- 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 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!

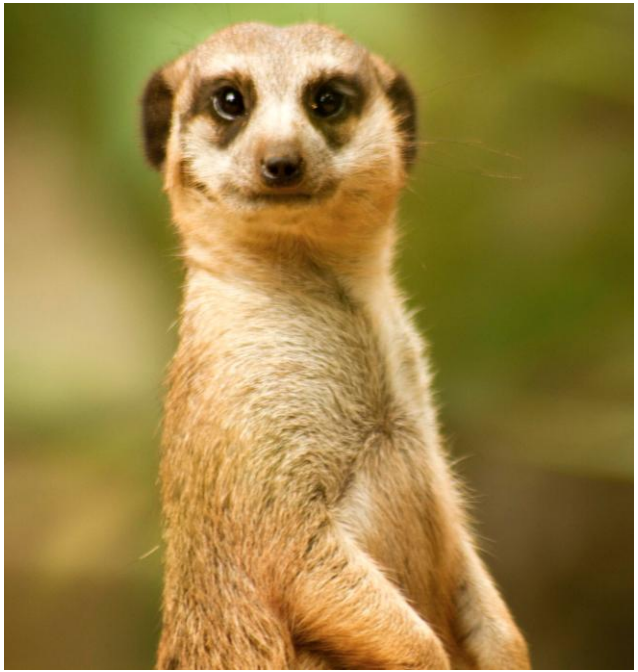
Chiron_0.7.tar.gz

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/

Evading Suricata



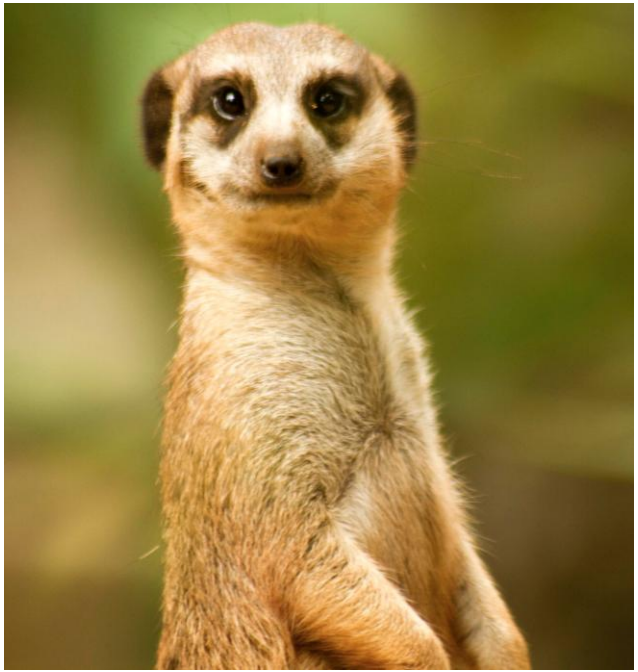
- Versions 2.0.1, 2.0.2 and 2.0.3 were evaded one by one by using various means.
- 1st case, version 2.0.1 :
  - An IPv6 Destination Option header is used a part of the fragmentable piece of the IPv6 datagram.
  - The IPv6 Destinations Option header is padded with six (6) octets of bytes (at least).
  - The IPv6 datagram is fragmented in at least 7 fragments, which are sent mis-ordered.

No.	Time	Source	Destination	Protocol	Length	Info
3	0.046598	fdf3:f0c0:2567:7fe4:800:27ff:fe00:0	fdf3:f0c0:2567:7fe4:a00:27ff:fe74:ddaa	IPv6	66	IPv6 fragment (nxt=TCP (6) off=9 id=0x1e72b80c)
4	0.166711	fdf3:f0c0:2567:7fe4:800:27ff:fe00:0	fdf3:f0c0:2567:7fe4:a00:27ff:fe74:ddaa	IPv6	70	IPv6 fragment (nxt=TCP (6) off=1 id=0x1e72b80c)
5	0.286228	fdf3:f0c0:2567:7fe4:800:27ff:fe00:0	fdf3:f0c0:2567:7fe4:a00:27ff:fe74:ddaa	IPv6	70	IPv6 fragment (nxt=TCP (6) off=2 id=0x1e72b80c)
6	0.404625	fdf3:f0c0:2567:7fe4:800:27ff:fe00:0	fdf3:f0c0:2567:7fe4:a00:27ff:fe74:ddaa	IPv6	70	IPv6 fragment (nxt=TCP (6) off=3 id=0x1e72b80c)
7	0.530602	fdf3:f0c0:2567:7fe4:800:27ff:fe00:0	fdf3:f0c0:2567:7fe4:a00:27ff:fe74:ddaa	IPv6	70	IPv6 fragment (nxt=TCP (6) off=4 id=0x1e72b80c)
8	0.650375	fdf3:f0c0:2567:7fe4:800:27ff:fe00:0	fdf3:f0c0:2567:7fe4:a00:27ff:fe74:ddaa	IPv6	70	IPv6 fragment (nxt=TCP (6) off=5 id=0x1e72b80c)
9	0.766897	fdf3:f0c0:2567:7fe4:800:27ff:fe00:0	fdf3:f0c0:2567:7fe4:a00:27ff:fe74:ddaa	IPv6	70	IPv6 fragment (nxt=TCP (6) off=6 id=0x1e72b80c)
10	0.887468	fdf3:f0c0:2567:7fe4:800:27ff:fe00:0	fdf3:f0c0:2567:7fe4:a00:27ff:fe74:ddaa	IPv6	70	IPv6 fragment (nxt=TCP (6) off=7 id=0x1e72b80c)
11	1.004217	fdf3:f0c0:2567:7fe4:800:27ff:fe00:0	fdf3:f0c0:2567:7fe4:a00:27ff:fe74:ddaa	IPv6	70	IPv6 fragment (nxt=TCP (6) off=8 id=0x1e72b80c)
12	1.125647	fdf3:f0c0:2567:7fe4:800:27ff:fe00:0	fdf3:f0c0:2567:7fe4:a00:27ff:fe74:ddaa	TCP	70	[TCP segment of a reassembled PDU]

```
▷ 0110 .... = Version: 6
▷ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00000000
  .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
  Payload length: 16
  Next header: IPv6 fragment (44)
  Hop limit: 64
  Source: fdf3:f0c0:2567:7fe4:800:27ff:fe00:0 (fdf3:f0c0:2567:7fe4:800:27ff:fe00:0)
  Destination: fdf3:f0c0:2567:7fe4:a00:27ff:fe74:ddaa (fdf3:f0c0:2567:7fe4:a00:27ff:fe74:ddaa)
  [Destination SA MAC: CadmusCo_74:dd:aa (08:00:27:74:dd:aa)]
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
▷ Fragmentation Header
▷ [10 IPv6 Fragments (76 bytes): #12(8), #4(8), #5(8), #6(8), #7(8), #8(8), #9(8), #10(8), #11(8), #3(4)]
▷ Destination Option
▷ Transmission Control Protocol, Src Port: 64340 (64340), Dst Port: http (80), Seq: 0, Len: 8
```

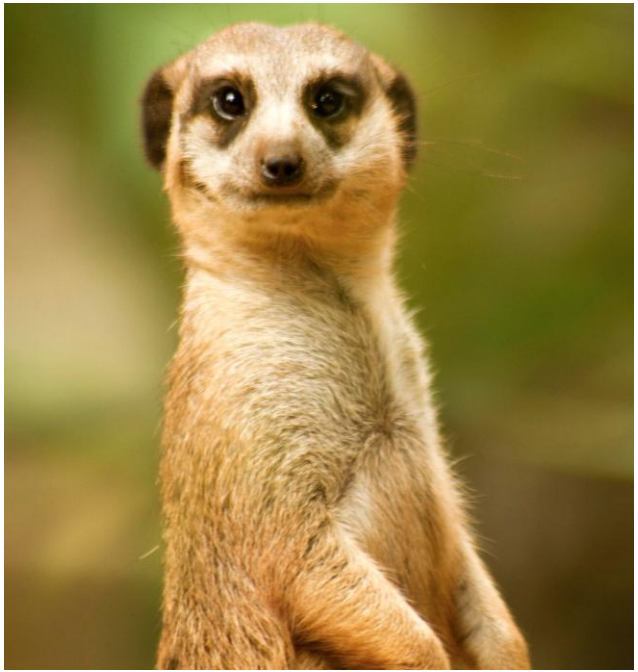
```
0000  08 00 27 74 dd aa 0a 00  27 00 00 00 86 dd 60 00   ..'t.... '.....`.
0010  00 00 00 10 2c 40 fd f3  f0 c0 25 67 7f e4 08 00   ....,@.. %g....
```

Evading Suricata



- 2nd case, version 2.0.2:
  - A) Fragmentation and Extension headers in both the fragmentable and the unfragmentable part.
  - B) Abusing the delay between two consecutive fragments (when preciseness does matter)
  - C) Using unknown/not supported extension headers

Evading Suricata



- 3rd case, version 2.0.3:
  - Scenario A from version 2.0.2 but using a, IPv6 Routign header instead of a Destination Options header.
  - Several variations can also be used (please, see the paper for more info).

Suricata Developers in each case reacted really fast

Evading TippingPoint, "The Old Way" (Mar 2014)

IPv6 Destination Options header Next Header Value = 6	**TCP header + payload ...**

IPv6 header Next Header Value = 43	**IPv6 Fragment Extension header** Next Header Value = 60	(part 1 out of 2 of the fragmentable part)

IPv6 header Next Header Value = 43	**IPv6 Fragment Extension header** Next Header Value = 6	(part 2 out of 2 of the fragmentable part)

TippingPoint®

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 Again TP After Patching

Fragment 1:

IPv6 main header nh=44	IPv6 Fragment Extension Header nh=60	IPv6 Destination Options Ext. Hdr nh=6	TCP Header	TCP payload Part 1/2

Fragment 2:

IPv6 main header nh=44	IPv6 Fragment Extension Header nh=6/**60**	TCP payload Part 2/2

Fragment 2 (again):

IPv6 main header nh=44	IPv6 Fragment Extension Header nh=6	TCP payload Part 2/2

Evading Snort

- Version2.9.6.2 GRE (build 77), Registered User's Release Rules, default installation.
- Use 9 times the Destination Options header, even if not fragmented.
  - or 8 Dest Opt and 1 Frag Ext Hdr
  - or, 1 Hop-by-Hop, 1 Routing Header, 1 Dest Opt Header, 1 Fragment Header, 5 Dest Opt headers, etc.
- To handle it:
  - Enable pre-processor *decoder.rules*.
  - A *"[116:456:1] (snort_decoder) WARNING: too many IP6 extension headers "* alert is triggered.

Why This Way of Handling Such Attacks is not the Best

- The "attack" itself (http traffic in our tests) is still NOT detected.
- Quite a few false "alarms" (warnings) are generated by the preproc/decoder.rules.
- From an RFCs perspective, there can be fully legitimate packets that include nine or more IPv6 Extension Headers.
- To make matter worse, the upper-layer can also be an IPv6 main header, which can include its own IPv6 Extension headers, etc.

Evading Sourcefire

- Sourcefire, Model 3D7020 (63) Version 5.2.0.3 (Build 48) is based on Snort 2.9.6 (Build 57)
- After enabling the Preproc decoder Rules and specifically, the GID 116 family and making sure that the rules with SID 458 (IPV6_BAD_FRAG_PKT), 272 and 273 are enabled, Sourcefire can be evaded.
  - a. The unfragmentable part consists of three (3) Destination Option headers
  - b. The fragmentable part consists of two (2) Destination Option headers plus the layer 4 header.
  - c. The aforementioned datagram is splitted in two fragments.

Evading Sourcefire

No.	Time	Source	Destination	Protocol	Length	Info
5	0.064967	2001:db8:1:1::aa	2001:db8:1:1::cc	IPv6	94	IPv6 fragment (nxt=IPv6 destination option (60) off=0 id=0x56eecfa7)[Ma
6	0.190343	2001:db8:1:1::aa	2001:db8:1:1::cc	ICMPv6	102	Echo (ping) request id=0x129c, seq=0, hop limit=64 (reply in 7)
7	0.190407	2001:db8:1:1::cc	2001:db8:1:1::aa	ICMPv6	62	Echo (ping) reply id=0x129c, seq=0, hop limit=128 (request in 6)

```
Source: 2001:db8:1:1::aa (2001:db8:1:1::aa)
Destination: 2001:db8:1:1::cc (2001:db8:1:1::cc)
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
▸ Destination Option
▸ Destination Option
▸ Destination Option
▸ Fragmentation Header
▾ [2 IPv6 Fragments (24 bytes): #5(8), #6(16)]
    [Frame: 5, payload: 0-7 (8 bytes)]
    [Frame: 6, payload: 8-23 (16 bytes)]
    [Fragment count: 2]
    [Reassembled IPv6 length: 24]
    [Reassembled IPv6 data: 3c000100010200003a0001000102000080001035129c0000]
▸ Destination Option
▸ Destination Option
▾ Internet Control Message Protocol v6
  Type: Echo (ping) request (128)
  Code: 0
  Checksum: 0x1035 [correct]
  Identifier: 0x129c
  Sequence: 0
  [Response In: 7]
```

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.
- Vendors should definitely make fully-compliant products and test them thoroughly before claiming IPv6-readiness.
- For the time being: Configure your devices to drop IPv6 Extension headers not used in your environment.

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.
  - Meet the protocol, play with it, test it in your lab and in your environment, study thoroughly potential configurations and finally, use it.
  - You will have to to do it, sooner or later. The earlier you will be familiarised with it, the better.

There's never enough time...

THANK YOU...

...for yours!

Tool & Slides:

https://www.insinuator.net

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

(..soon)

Questions?

- You can reach us at:
  - aatlasis@secfu.net, www.secfu.net
  - erey@ernw.de, www.insinuator.net, www.ernw.de
- Follow us at:
  - @AntoniosAtlasis
  - @Enno_Insinuator