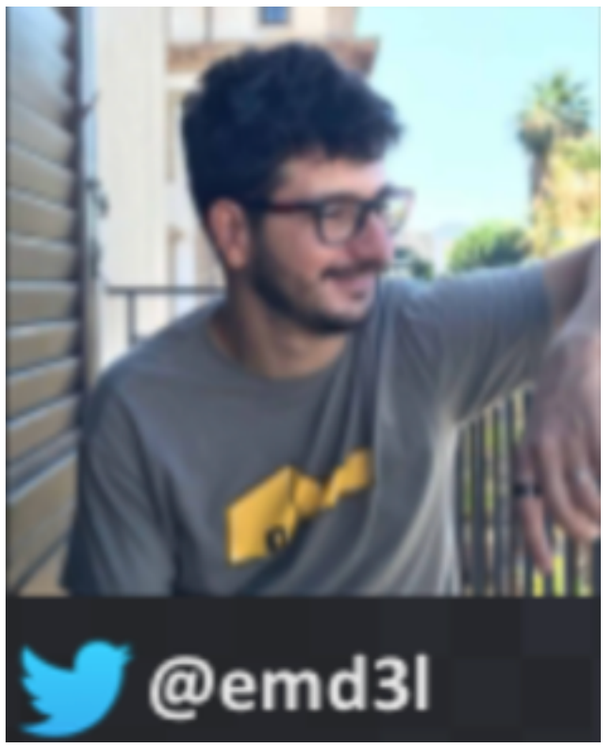
BINARY ANALYSIS NOTES

Mariano Graziano

Malware Research Team - Cisco Talos

M0LECON 2019
Turin, Italy - 30/11/2019

whoami



- Technical Leader at Cisco Talos
- PhD in System Security (Eurecom)
- Alma mater: Politecnico di Torino
- Binary/Malware analysis, Memory forensics, Automation

ROPMEMU

PyREBox

GhIDA

bass

AUTOMATED SIGNATURE SYNTHESIZER

FIRST

OUTLINE

- Binary Analysis
- Linux Threat Landscape
- ELF

BINARY ANALYSIS

- How a binary is generated?

BINARY ANALYSIS

- How a binary is generated?
  - Compilation (from source code to machine code)

BINARY ANALYSIS

- How a binary is generated?
  - Compilation (from source code to machine code)
    - Preprocessing/compilation/assembling/linking
    - Statically linked binaries
- Interpreted programs and JIT compilation —> Scripts to executables (e.g. PyInstaller)

BINARY ANALYSIS

Binary analysis is the art of understanding compiled programs

BINARY ANALYSIS

- Binary analysis is the art of understanding compiled programs

- From machine code to assembly —> Disassembler

DISASSEMBLER

```
emdel@ubuntu:~$ echo -ne "\x83\xc4\x04\x5b" | ndisasm - -b32
00000000  83C404              add esp,byte +0x4
00000003  5B                  pop ebx
emdel@ubuntu:~$ echo -ne "\x04\x5b" | ndisasm - -b32
00000000  045B                add al,0x5b
emdel@ubuntu:~$
```

BINARY ANALYSIS

- Binary analysis is the art of understanding compiled programs
- From machine code to assembly
- Understand from the machine code what the binary does and its properties/behavior

BINARY ANALYSIS

- How binary analysis is conducted?

BINARY ANALYSIS

- How binary analysis is conducted?
  - Static Analysis

BINARY ANALYSIS

- How binary analysis is conducted?
  - Static Analysis
    - Strings/symbols/API calls
    - disassembler

BINARY ANALYSIS

- How binary analysis is conducted?
  - Static Analysis

cost

BINARY ANALYSIS

- How binary analysis is conducted?
  - Dynamic analysis

BINARY ANALYSIS

- How binary analysis is conducted?
- Dynamic analysis:
  - Debugging/Instrumented environment
  - Interaction with the OS

BINARY ANALYSIS

- How binary analysis is conducted?
  - Dynamic analysis

BINARY ANALYSIS

- Why binary analysis is useful?

BINARY ANALYSIS

- Why binary analysis is useful?
  - Reverse engineering activities
  - Malware analysis/Exploitation
  - Detect plagiarism
  - Interoperability
  - Modify and understand applications (closed source)

BINARY ANALYSIS

- Why binary analysis is hard?

BINARY ANALYSIS

- Why binary analysis is hard?
  - Semantic gap

OUTLINE

- Binary Analysis
- **Linux Threat Landscape**
- ELF

DESKTOP

OS	Share
Windows	86,66
OSX	11,03
Linux	1,66
Chrome OS	0,41
Unknown	0,24

https://netmarketshare.com/operating-system-market-share.aspx?options=%7B%22filter%22%3A%7B%22%24and%22%3A%5B%7B%22deviceType%22%3A%7B%22%24in%22%3A%5B%22Desktop%2Flaptop%22%5D%7D%7D%5D%7D%2C%22dateLabel%22%3A%22Custom%22%2C%22attributes%22%3A%22share%22%2C%22group%22%3A%22platform%22%2C%22sort%22%3A%7B%22share%22%3A-1%7D%2C%22id%22%3A%22platformsDesktop%22%2C%22dateInterval%22%3A%22Monthly%22%2C%22dateStart%22%3A%222019-09%22%2C%22dateEnd%22%3A%222019-10%22%2C%22segments%22%3A%22-1000%22%7D

DESKTOP

OS	Share
Windows	86,66
OSX	11,03
Linux	1,66
Chrome OS	0,41
Unknown	0,24

https://netmarketshare.com/operating-system-market-share.aspx?options=%7B%22filter%22%3A%7B%22%24and%22%3A%5B%7B%22deviceType%22%3A%7B%22%24in%22%3A%5B%22Desktop%2Flaptop%22%5D%7D%7D%5D%7D%2C%22dateLabel%22%3A%22Custom%22%2C%22attributes%22%3A%22share%22%2C%22group%22%3A%22platform%22%2C%22sort%22%3A%7B%22share%22%3A-1%7D%2C%22id%22%3A%22platformsDesktop%22%2C%22dateInterval%22%3A%22Monthly%22%2C%22dateStart%22%3A%222019-09%22%2C%22dateEnd%22%3A%222019-10%22%2C%22segments%22%3A%22-1000%22%7D

MOBILE

OS	Share
Android	69,34
iOS	30,3
Unknown	0,25
Series 40	0,04
Windows Phone	0,03
Linux	0,02

https://netmarketshare.com/operating-system-market-share.aspx?options=%7B%22filter%22%3A%7B%22%24and%22%3A%5B%7B%22deviceType%22%3A%7B%22%24in%22%3A%5B%22Mobile%22%5D%7D%7D%5D%7D%2C%22dateLabel%22%3A%22Custom%22%2C%22attributes%22%3A%22share%22%2C%22group%22%3A%22platform%22%2C%22sort%22%3A%7B%22share%22%3A-1%7D%2C%22id%22%3A%22platformsDesktop%22%2C%22dateInterval%22%3A%22Monthly%22%2C%22dateStart%22%3A%222019-09%22%2C%22dateEnd%22%3A%222019-10%22%2C%22segments%22%3A%22-1000%22%7D

MOBILE

OS	Share
Android	69,34
iOS	30,3
Unknown	0,25
Series 40	0,04
Windows Phone	0,03
Linux	0,02

https://netmarketshare.com/operating-system-market-share.aspx?options=%7B%22filter%22%3A%7B%22%24and%22%3A%5B%7B%22deviceType%22%3A%7B%22%24in%22%3A%5B%22Mobile%22%5D%7D%7D%5D%7D%2C%22dateLabel%22%3A%22Custom%22%2C%22attributes%22%3A%22share%22%2C%22group%22%3A%22platform%22%2C%22sort%22%3A%7B%22share%22%3A-1%7D%2C%22id%22%3A%22platformsDesktop%22%2C%22dateInterval%22%3A%22Monthly%22%2C%22dateStart%22%3A%222019-09%22%2C%22dateEnd%22%3A%222019-10%22%2C%22segments%22%3A%22-1000%22%7D

WEB

OS	Share
Unix	70,8
Windows	29.2

https://w3techs.com/technologies/overview/operating_system

WEB

OS	Share
Unix	70,8
Windows	29.2

https://w3techs.com/technologies/overview/operating_system

MALWARE?

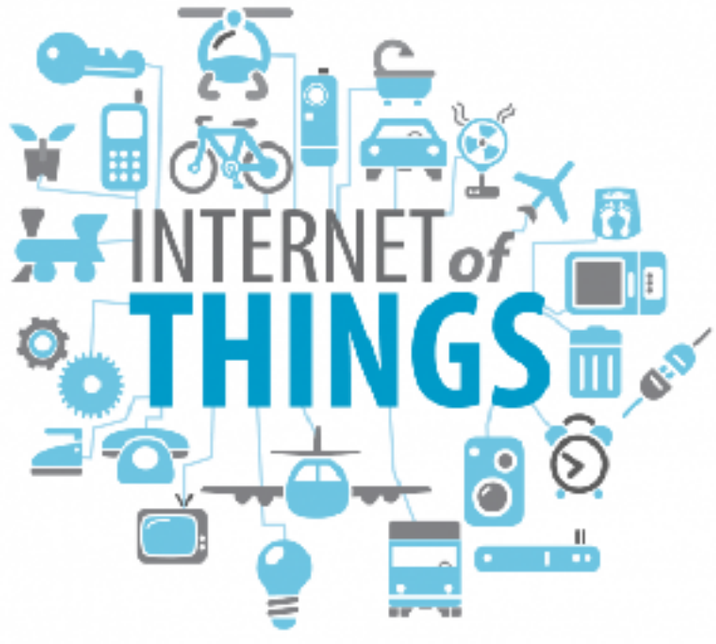
[1] http://www.tom-yam.or.jp/2238/ref/secur.pdf

REALITY

Shellshock

文件名 .扩展名	大小(类型)	修改时间	点击量↓
linux2.6	1.78 MB	2014-8-4 下午 09:01:32	2667
linux-mips	1.10 MB	2014-8-4 下午 09:01:34	2575
windou	380.53 KB	2014-8-30 下午 10:13:52	2350
Fdowsd	447.55 KB	2014-8-30 下午 10:13:52	2338
dd-wrt	1.10 MB	2014-8-4 下午 09:01:34	2336
dows2.4	402.17 KB	2014-9-1 下午 05:16:06	2267
uhdyp	402.28 KB	2014-9-12 上午 10:44:06	2091
linux-arm	977.99 KB	2014-8-4 下午 09:01:33	1819

The POC of how many China ELF DDoSers was downloaded since August 2014 in Multi Platform/Architecture - PoC by @MalwareMustDie!



INFECTIONS

- Exploiting known vulnerabilities:
  - Apache struts/ElasticSearch/Redis etc
  - Shellshock
  - CMS vulnerabilities (Wordpress, Joomla etc)
- Low hanging fruits:
  - Telnet and SSH bruteforcing

MALWARE

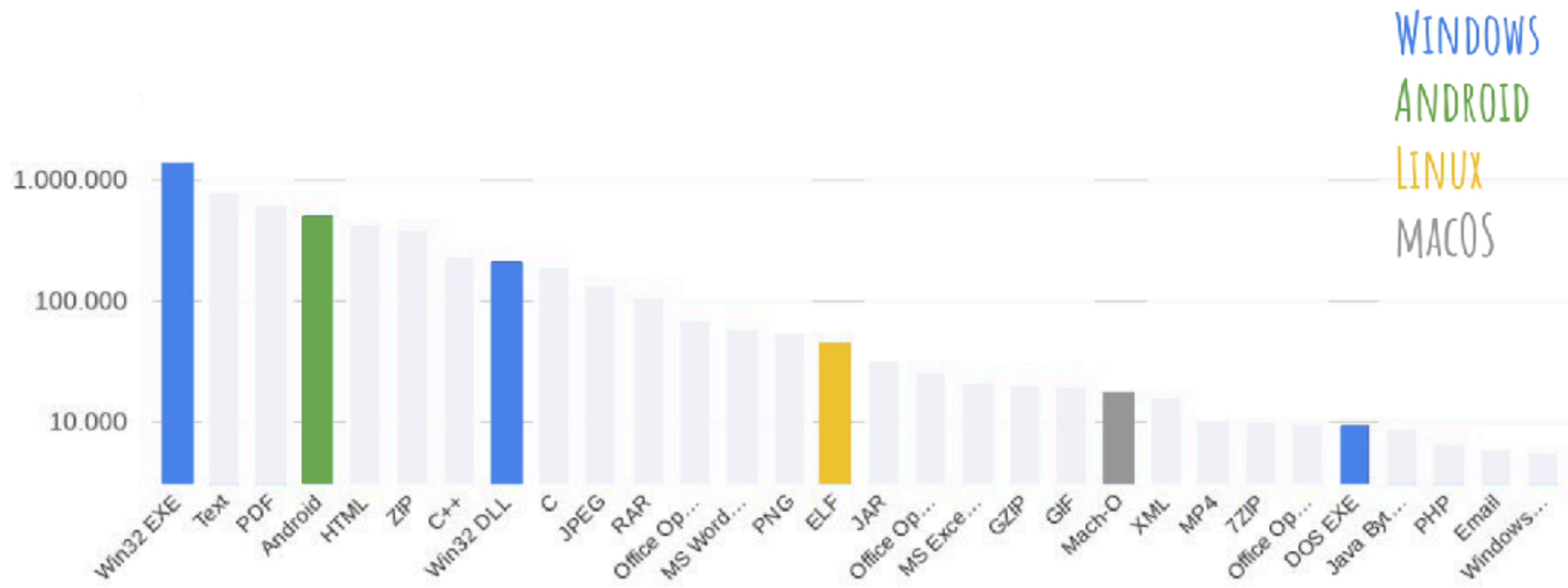
- Xor.DDoS — rootkit component
- ChinaZ — via shellshock
- Hand of Thief — Banker
- Mayhem
- Mirai
- VPNFilter — multistage
- HiddenWasp
- …

MALWARE

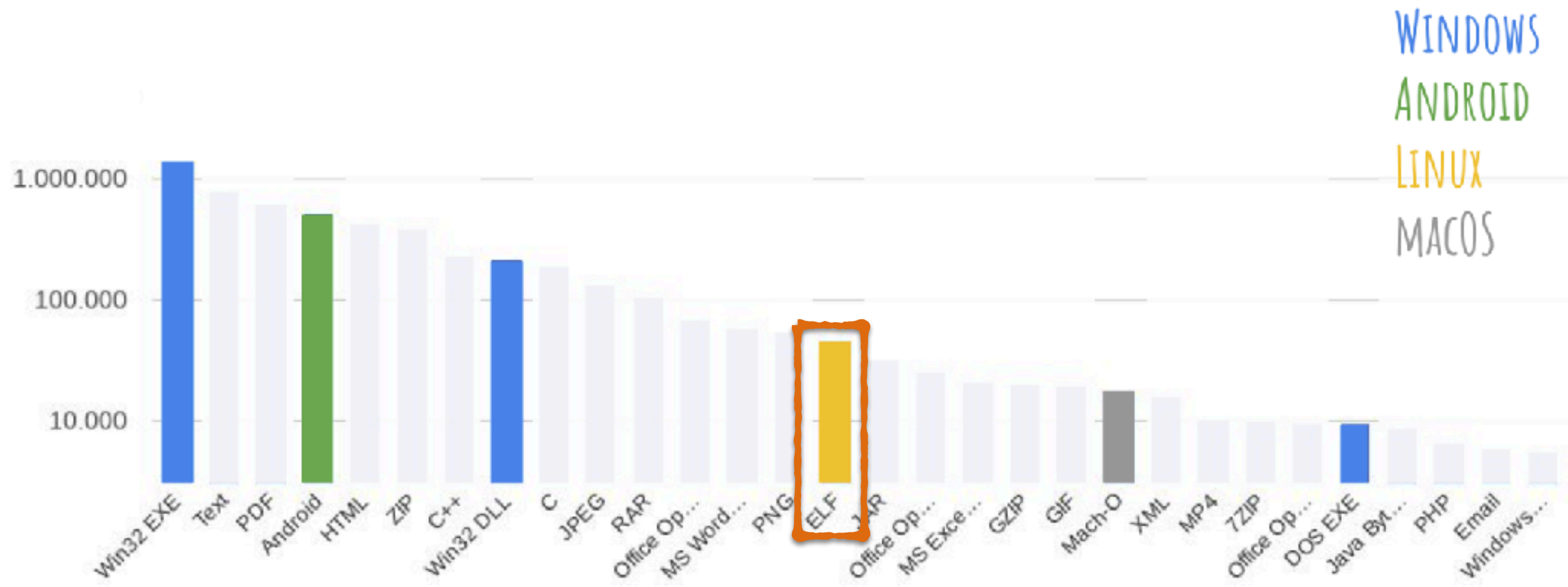
- Xor.DDoS — rootkit component
- ChinaZ — via shellshock
- Hand of Thief — Banker
- Mayhem
- Mirai
- VPNFilter — multistage
- HiddenWasp
- …

Many families and categories

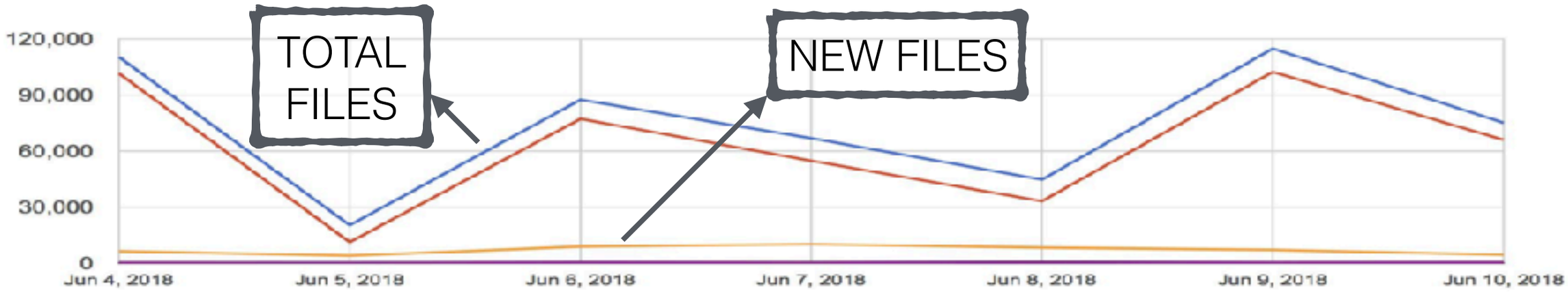
CURRENT SITUATION



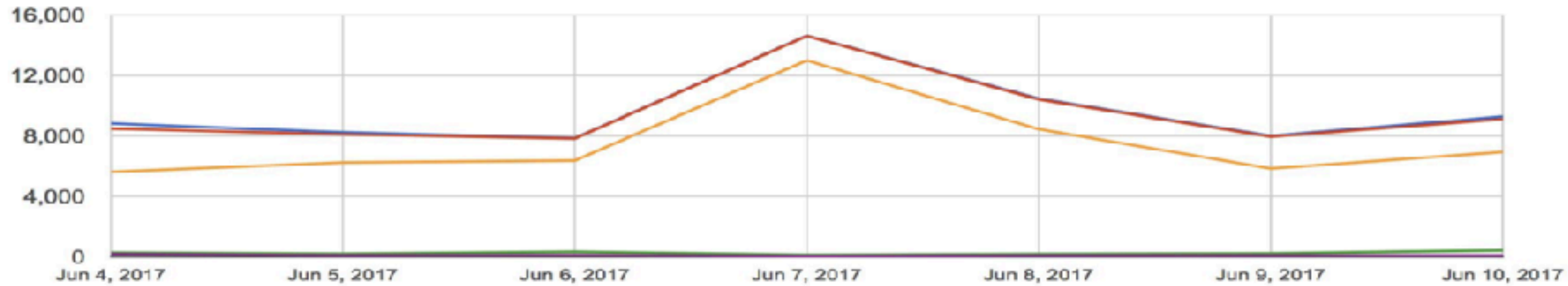
CURRENT SITUATION



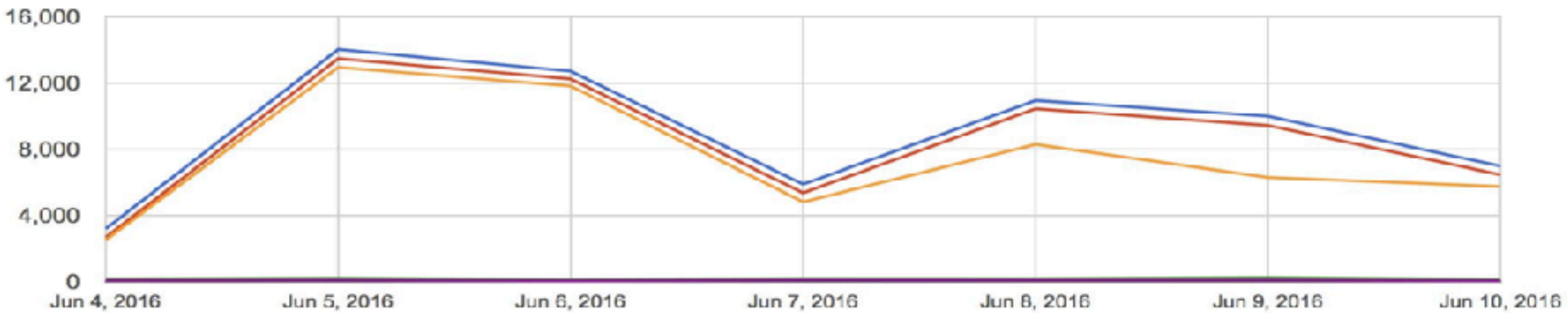
ELF SITUATION



2018

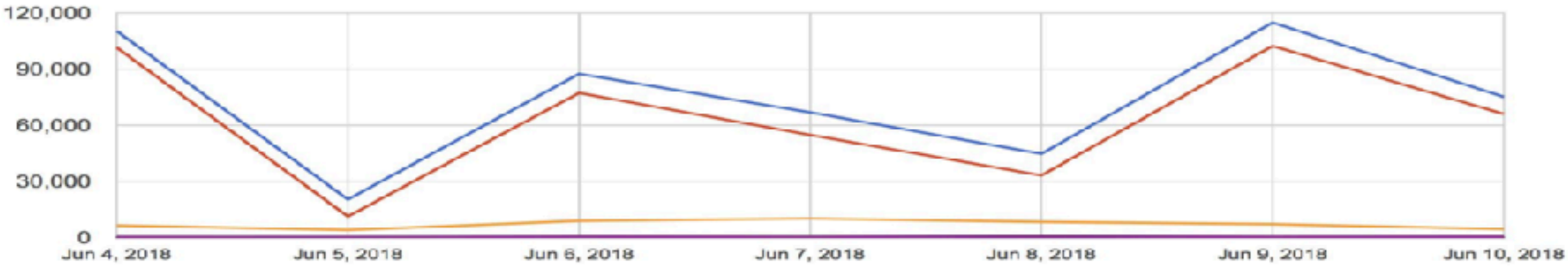


2017



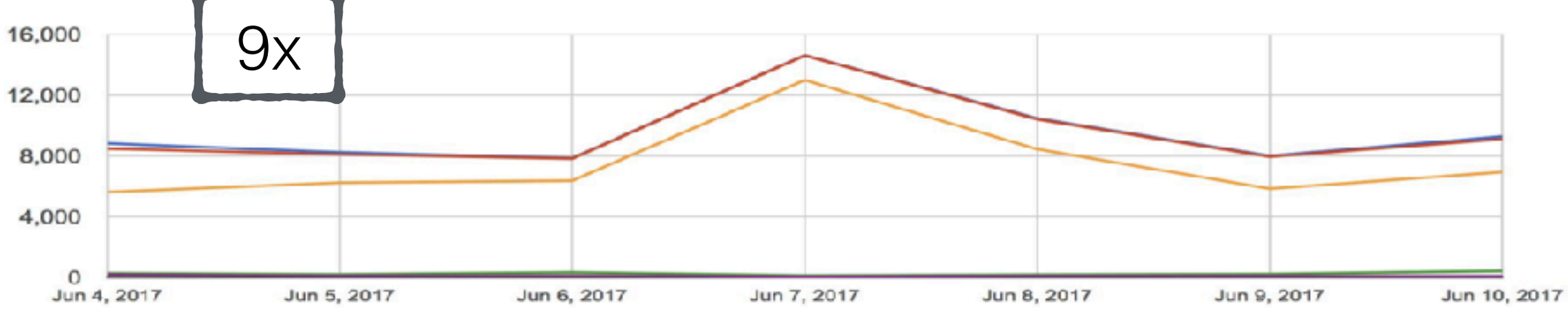
2016

ELF SITUATION

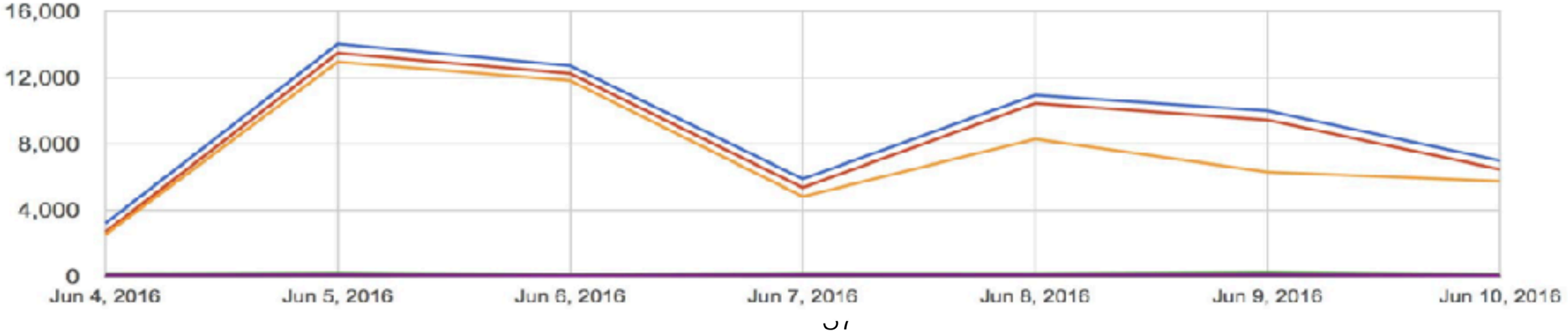


2018

9x

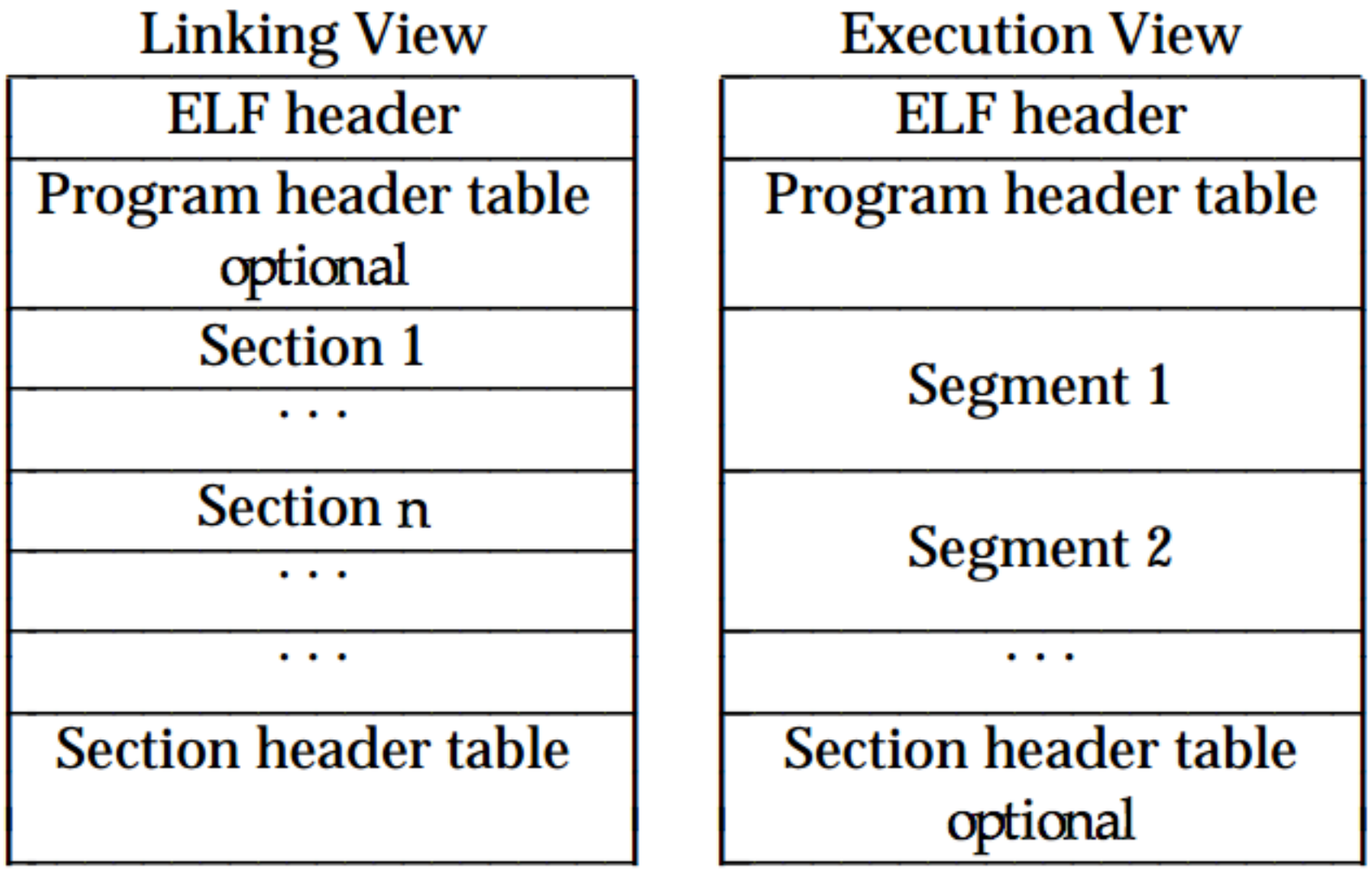


2017



2016

ELF


Linking View
ELF header
Program header table optional
Section 1
. . .
Section n
. . .
. . .
Section header table
Execution View
ELF header
Program header table
Segment 1
Segment 2
. . .
Section header table optional

ELF HEADER

```
typedef struct {
        unsigned char   e_ident[EI_NIDENT];
        Elf32_Half      e_type;
        Elf32_Half      e_machine;
        Elf32_Word      e_version;
        Elf32_Addr      e_entry;
        Elf32_Off       e_phoff;
        Elf32_Off       e_shoff;
        Elf32_Word      e_flags;
        Elf32_Half      e_ehsize;
        Elf32_Half      e_phentsize;
        Elf32_Half      e_phnum;
        Elf32_Half      e_shentsize;
        Elf32_Half      e_shnum;
        Elf32_Half      e_shstrndx;
} Elf32_Ehdr;
```

e_ident

Name	Value	Purpose
EI_MAG0	0	File identification
EI_MAG1	1	File identification
EI_MAG2	2	File identification
EI_MAG3	3	File identification
EI_CLASS	4	File class
EI_DATA	5	Data encoding
EI_VERSION	6	File version
EI_PAD	7	Start of padding bytes
EI_NIDENT	16	Size of e_ident[]

Name	Value	Position
ELFMAG0	0x7f	e_ident[EI_MAG0]
ELFMAG1	'E'	e_ident[EI_MAG1]
ELFMAG2	'L'	e_ident[EI_MAG2]
ELFMAG3	'F'	e_ident[EI_MAG3]

e_machine

Name	Value	Meaning
EM_NONE	0	No machine
EM_M32	1	AT&T WE 32100
EM_SPARC	2	SPARC
EM_386	3	Intel 80386
EM_68K	4	Motorola 68000
EM_88K	5	Motorola 88000
EM_860	7	Intel 80860
EM_MIPS	8	MIPS RS3000

SEGMENTS

- Execution view — How to create a process image
- A segment can contain zero or more sections

p_type

Name	Value
PT_NULL	0
PT_LOAD	1
PT_DYNAMIC	2
PT_INTERP	3
PT_NOTE	4
PT_SHLIB	5
PT_PHDR	6
PT_LOPROC	0x70000000
PT_HIPROC	0x7fffffff

DEMO 0x00

READELF

ELF HEADER

/BIN/LS

```
00000000   7f 45 4c 46 02 01 01 00   |.ELF....|
00000008   00 00 00 00 00 00 00 00   |........|
00000010   02 00 3e 00 01 00 00 00   |..>.....|
00000018   c5 48 40 00 00 00 00 00   |.H@.....|
00000020   40 00 00 00 00 00 00 00   |@.......|
00000028   48 c7 01 00 00 00 00 00   |H.......|
00000030   00 00 00 00 40 00 38 00   |....@.8.|
00000038   09 00 40 00 1b 00 1a 00   |..@.....|
```

- e_ident
- e_type
- e_machine
- e_version
- e_entry
- e_phoff
- e_shoff
- e_flags
- e_ehsize
- e_phentsize
- e_phnum
- e_shentsize
- e_shnum
- e_shstrndx

ELF HEADER

/BIN/LS

e_ident

Name	Value	Purpose
EI_MAG0	0	File identification
EI_MAG1	1	File identification
EI_MAG2	2	File identification
EI_MAG3	3	File identification
EI_CLASS	4	File class
EI_DATA	5	Data encoding
EI_VERSION	6	File version
EI_PAD	7	Start of padding bytes
EI_NIDENT	16	Size of e_ident[]

Name	Value	Position
ELFMAG0	0x7f	e_ident[EI_MAG0]
ELFMAG1	'E'	e_ident[EI_MAG1]
ELFMAG2	'L'	e_ident[EI_MAG2]
ELFMAG3	'F'	e_ident[EI_MAG3]

EI_DATA

EI_DATA The sixth byte specifies the data encoding of the processor-specific data in the file. Currently these encodings are supported:

ELFDATANONE Unknown data format.
ELFDATA2LSB Two's complement, little-endian.
ELFDATA2MSB Two's complement, big-endian.

DEMO 0x00

1 BYTE

https://github.com/radareorg/r2con2019/blob/master/talks/elf_crafting/ELF_Crafting_ulexec.pdf

GLIBC INITIALIZATION

- Where is my main()?

GLIBC INITIALIZATION

- ELF entry point points to:
  - _start
  - glibc initialization code

```
  400440:   31 ed                   xor    %ebp,%ebp
  400442:   49 89 d1                mov    %rdx,%r9
  400445:   5e                      pop    %rsi
  400446:   48 89 e2                mov    %rsp,%rdx
  400449:   48 83 e4 f0             and    $0xfffffffffffffff0,%rsp
  40044d:   50                      push   %rax
  40044e:   54                      push   %rsp
  40044f:   49 c7 c0 e0 05 40 00    mov    $0x4005e0,%r8
  400456:   48 c7 c1 70 05 40 00    mov    $0x400570,%rcx
  40045d:   48 c7 c7 4d 05 40 00    mov    $0x40054d,%rdi
  400464:   e8 b7 ff ff ff          callq  400420
<__libc_start_main@plt>
  400469:   f4                      hlt
```

- fini
- init
- main

libc_start_main

- _start —> __libc_start_main(main, init, fini)

DEMO 0x01

CONSTRUCTOR

ANTI ANALYSIS

- Bad guys can complicate our job:
  - Anti analysis techniques
  - Anti debugging techniques
  - Packing

DEMO 0x02

STRIP

DEMO 0x03

ANTIDEBUG TECHNIQUES

DEMO 0x04

New NextCry Ransomware Encrypts Data on NextCloud Linux Servers

By Ionut Ilascu

November 15, 2019 02:08 PM 0

A new ransomware has been found in the wild that is currently undetected by antivirus engines on public scanning platforms. Its name is NextCry due to the extension appended to encrypted files and that it targets clients of the NextCloud file sync and share service.

The malware targets Nextcloud instances and for the time being there is no free decryption tool available for victims.

Zero detection

xact64, a Nextcloud user, posted on the BleepingComputer forum some details about the malware in an attempt to find a way to decrypt personal files.

Although his system was backed up, the synchronization process had started to update files on a laptop with their encrypted version on the server. He took action the moment he saw the files renamed but some of them still got processed by NextCry, otherwise known as Next-Cry.

DEMO 0x04

NEXTCRY

SHA256: 027d5f87ab71044a4bbac469b6a3bf5e02571c4661939699d9050a4300d10230

REMARKS

- Linux malware is a real threat
  - We have to be ready
  - We need more tools
  - We need to know the internals
- IoT complicates the analysis:
  - OS and architecture diversifications
  - Need more background knowledge

THE END

THANK YOU

email: magrazia@cisco.com
twitter: @emd3l