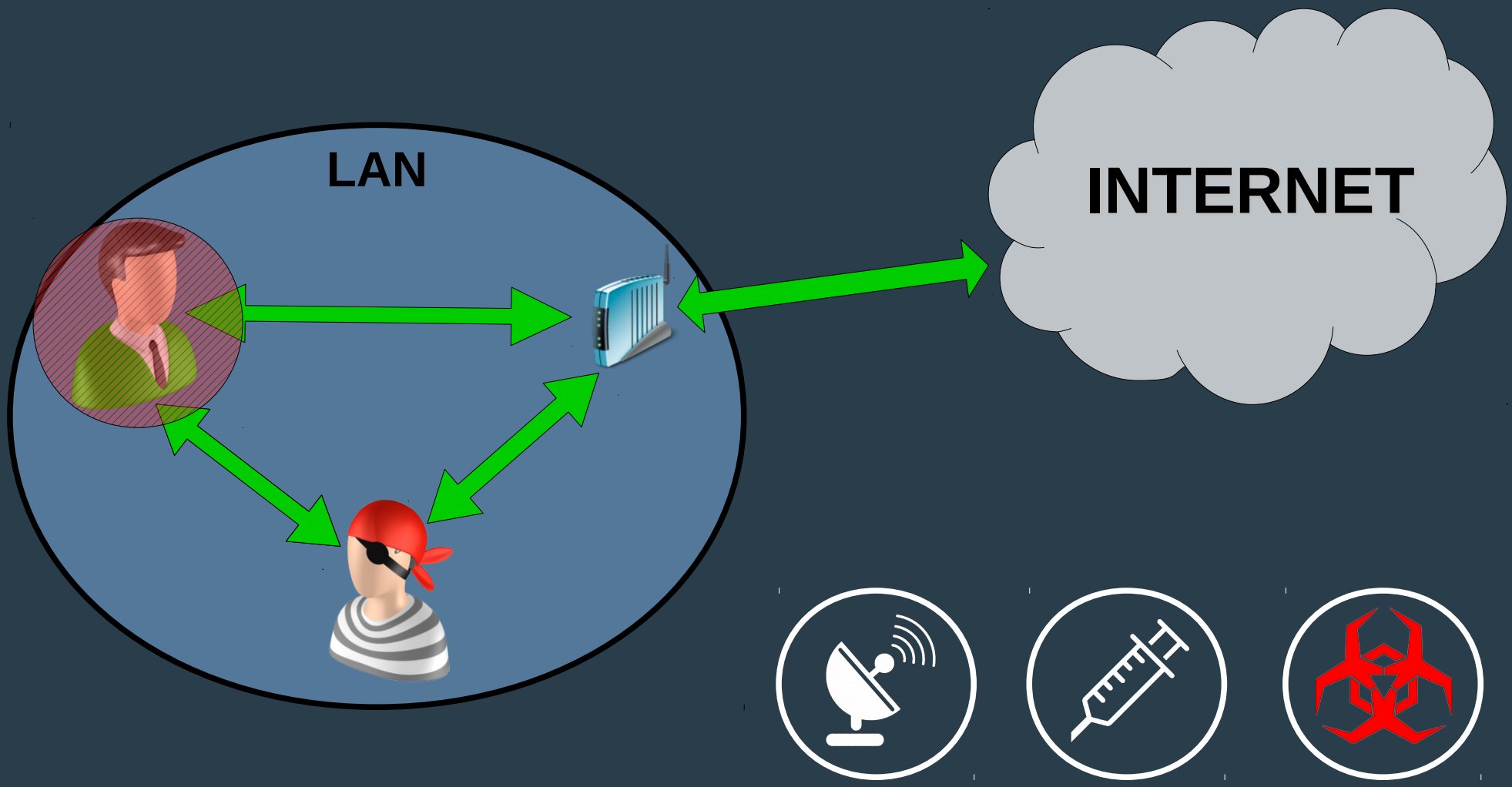
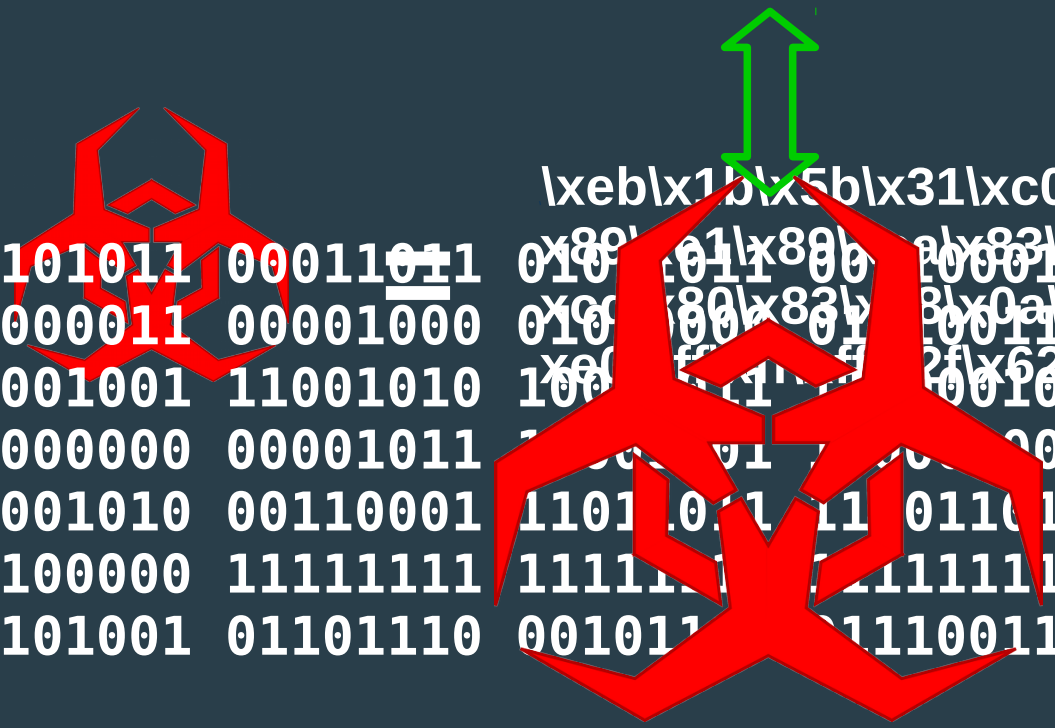
**INTERCEPT**
**INJECT**
**INFECT**

# Overview



# Shellcode



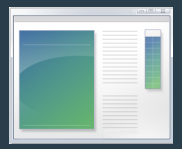
```
\xeb\x1b\x5b\x31\xc0\x89\x43\x08\x50\x53\
x89\x01\x89\xa[illegible]\x83\xc2\x04\x83\xc0\x0b\
xcd\x80\x83\x[illegible]\x0a\x31\xdb\xcd\x80\x08\
xe8[illegible]2f\x62\x69\x6e\x2f\x73\x68

11101011 00011011 010[illegible]011 00[illegible]10001 11000000 10001001
01000011 00001000 010[illegible]000 01[illegible]0011 10001001 11100001
10001001 11001010 100[illegible]011 [illegible]0010 00000100 10000011
11000000 00001011 1[illegible]01 [illegible]00 10000011 11101000
00001010 00110001 1101[illegible]11 [illegible]1 10000000 11101000
11100000 11111111 1111[illegible]1111 00101111 01100010
01101001 01101110 001011[illegible]1110011 01101000
```

# Common Attack

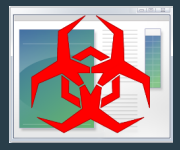
## Victim

## Cracker

www.trustme.ru

# Why it fail?

| Victim | Cracker |
| --- | --- |
|   |   |

# How does AV work?

```
11101011 00011011 01011011 00110001 11000000 10001001 01000011
00001000 01010000 01010011 10001001 11100001 10001001 11001010
10000011 11000010 00000100 10000011 11000000 00001011 11001101
10000000 10000011 11101000 00001010 00110001 11011011 11001101
10000000 11101000 11100000 11111111 11111111 11111111 00101111
01100010 01101001 01101110 00101111 01110011 01101000
```

**Definizioni**

| Virus | Signature |
|---|---|
| XYZ | 11000000 00001011 11001101 10000000 10000011 |

# Crypter

Passw: 'a'
ASCII = 01100001

11101011 00011011 01011011 00110001 11000000 10001001 01000011 00001000

+

a

01100001 01100001 01100001 01100001 01100001 01100001 01100001 01100001

=

10001010 01111010 00111010 01010000 10100001 11101000 00100010 01101001

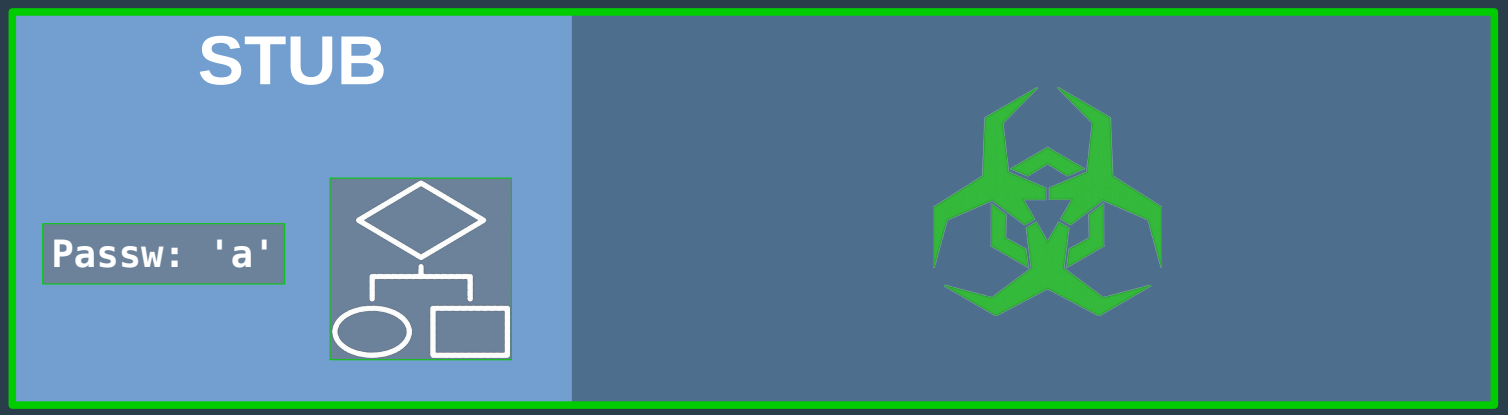


gianluca.gabrielli@autistici.org

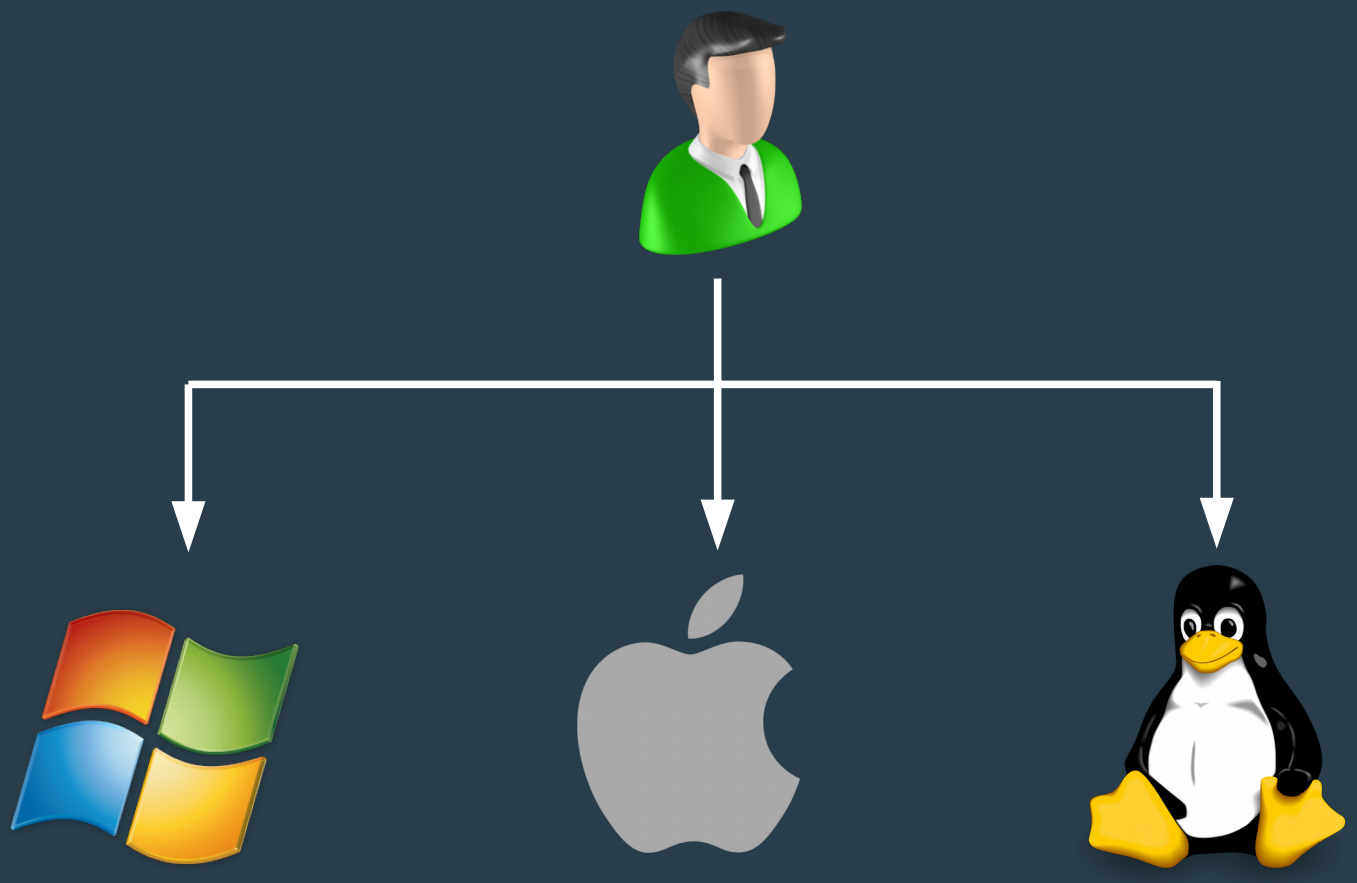
@Gianlucode

# Crypter

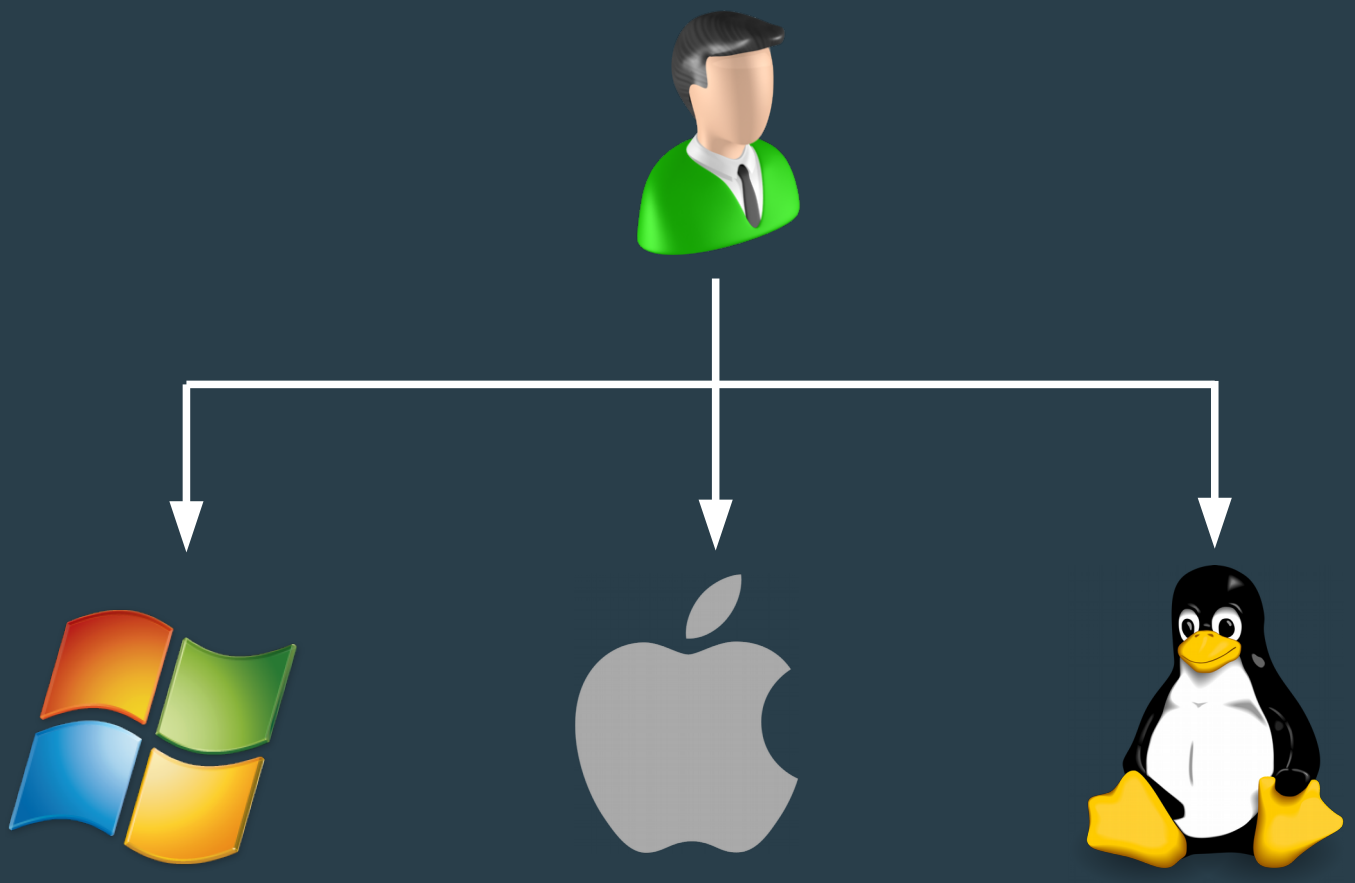
10001010 01111010 00111010 ( 010 0 1 100001 11101000 00100010 01101001



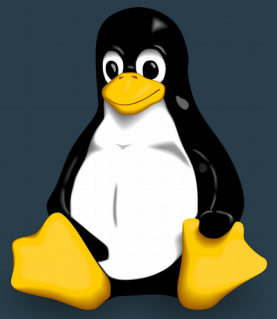

# Operative Systems



# Operative Systems



# Operative Systems

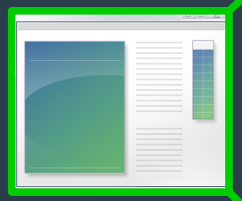


PE

Mach-O

ELF

# Code Caves



```
01100100 10010010 11101100
10010010 10111101 10100101
01001000 00101000 01100100
10010010 11101100 10010010
10111101 10100101 01001000
00101000 01100100 10010010
11101100 10010010 10111101
10100101 01001000 00101000
01100100 10010010 11101100
10010010 10111101 10100101
01001000 00101000 01100100
10010010 11101100 10010010
10111101 10100101 01001000
00101000 01100100 10010010
11101100 10010010 10111101
              .
              .
              .
01001000 00101000 01100100
10010010 11101100 10010010
10111101 10100101 01001000
```

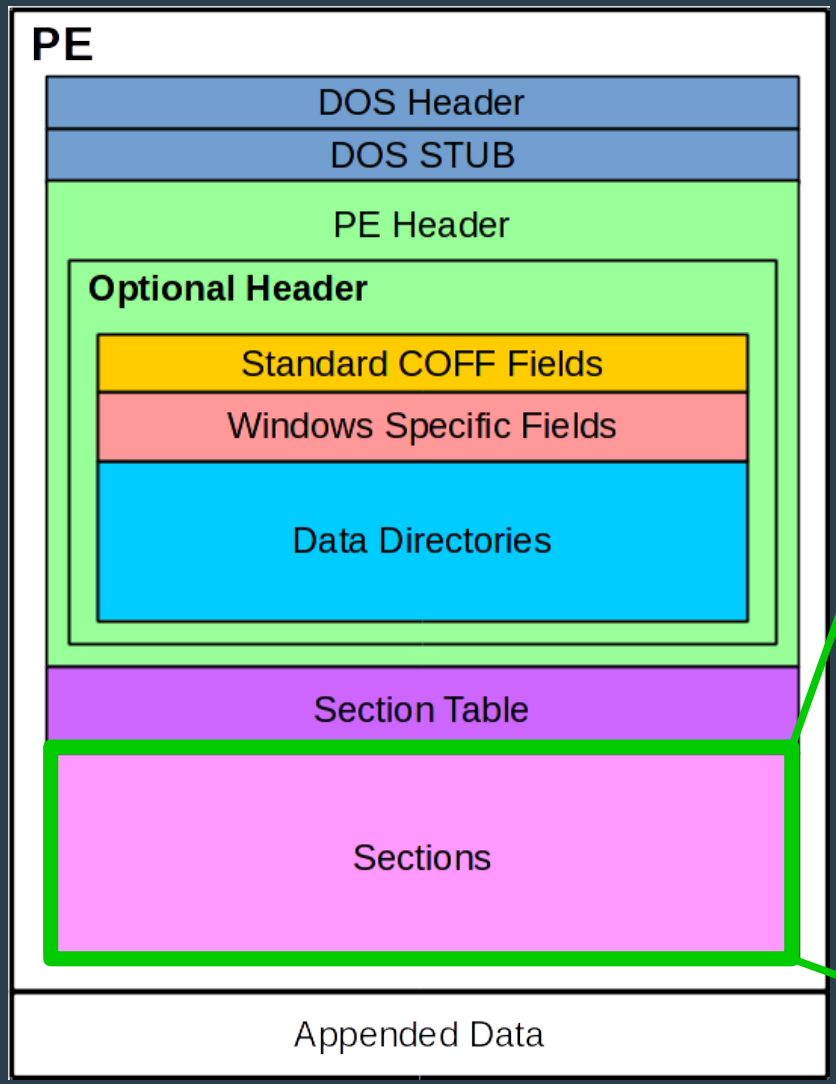
# Code Caves

```
01100100 10010010 11101100 10010010 11101100 10010010
10010010 10010010 11101100 10010010 10111101 10100101
01001000 11101100 10010010 10111101 00101000 01100100
11101100 10010010 10111101 10010010 11101100 10010010
10111101 10100101 10010010 11101100 10010010 01001000
00101000 01100100 11101100 10010010 10111101 10010010
11101100 10010010 10111101 10010010 11101100 10010010
10100101 10010010 11101100 10010010 01001000 00101000
01100100 10010010 11101100 10010010 10010010 11101100
00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000
00101000 10010010 11101100 10010010 01100100 10010010
11101100 10010010 10111101 10010010 11101100 00101000
                          .
                          .
                          .
01001000 10010010 11101100 10010010 00101000 01100100
10010010 11101100 00101000 10010010 11101100 10010010
10111101 10100101 10010010 11101100 10010010 01001000
```

# Code Caves

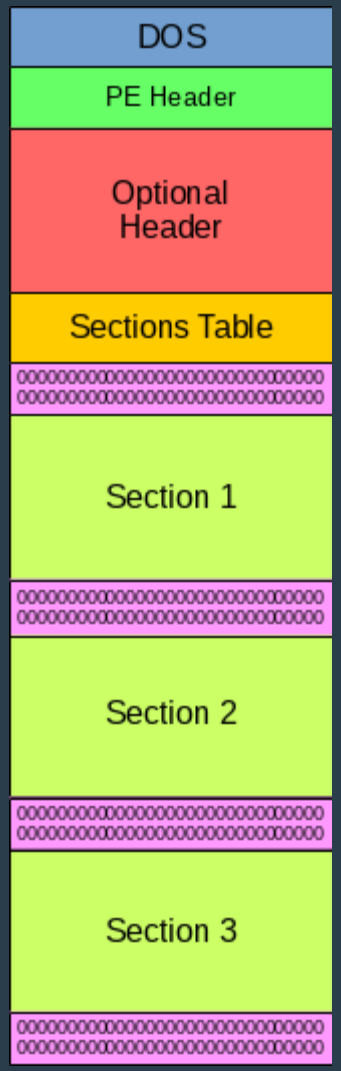
```
01100100 10010010 11101100 10010010 11101100 10010010
10010010 10010010 11101100 10010010 10111101 10100101
01001000 11101100 10010010 10111101 00101000 01100100
11101100 10010010 10111101 10010010 11101100 10010010
10111101 10100101 10010010 11101100 10010010 01001000
00101000 01100100 11101100 10010010 10111101 10010010
11101100 10010010 10111101 10010010 11101100 10010010
10100101 10010010 11101100 10010010 01001000 00101000
01100100 10010010 11101100 10010010 10010010 11101100
00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000
00101000 10010010 11101100 10010010 01100100 10010010
11101100 10010010 10111101 10010010 11101100 00101000
                          .
                          .
                          .
01001000 10010010 11101100 10010010 00101000 01100100
10010010 11101100 00101000 10010010 11101100 10010010
10111101 10100101 10010010 11101100 10010010 01001000
```

# Portable Executable




```
01100100 10010010 11101100 10010010
10111101 10100101 10010010 11101100
10010010 01001000 00101000 01100100
11101100 10010010 10111101 10010010
11101100 10010010 10111101 10010010
11101100 10010010 10100101 10010010
11101100 10010010 01001000 00101000
01100100 10010010 11101100 10010010
10010010 11101100 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00101000 10010010 11101100 10010010
01100100 10010010 11101100 10010010
10111101 10010010 11101100 00101000
                  .
                  .
                  .
01001000 10010010 11101100 1001001
00101000 01100100 10010010 1110110
```

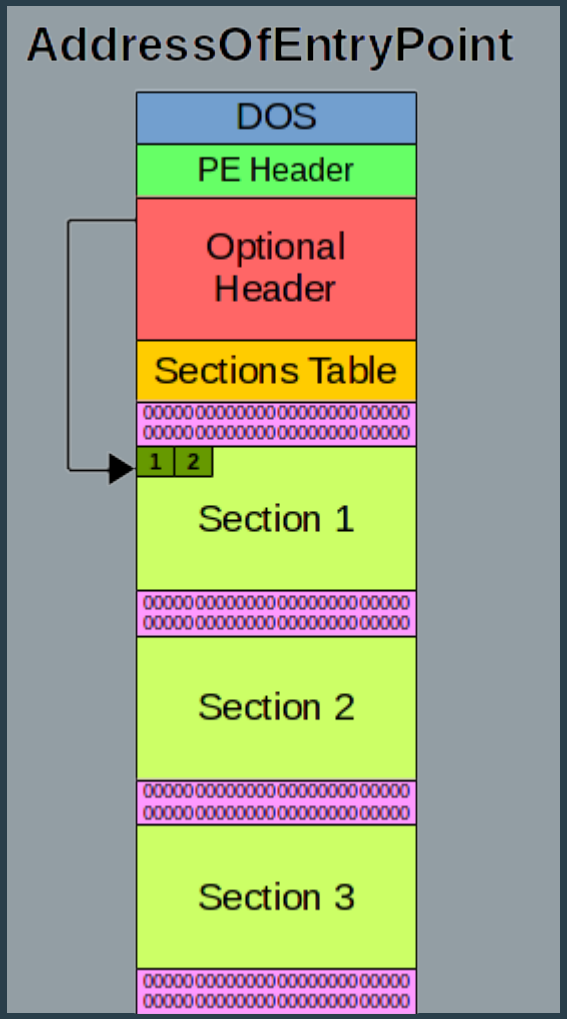
# Portable Executable



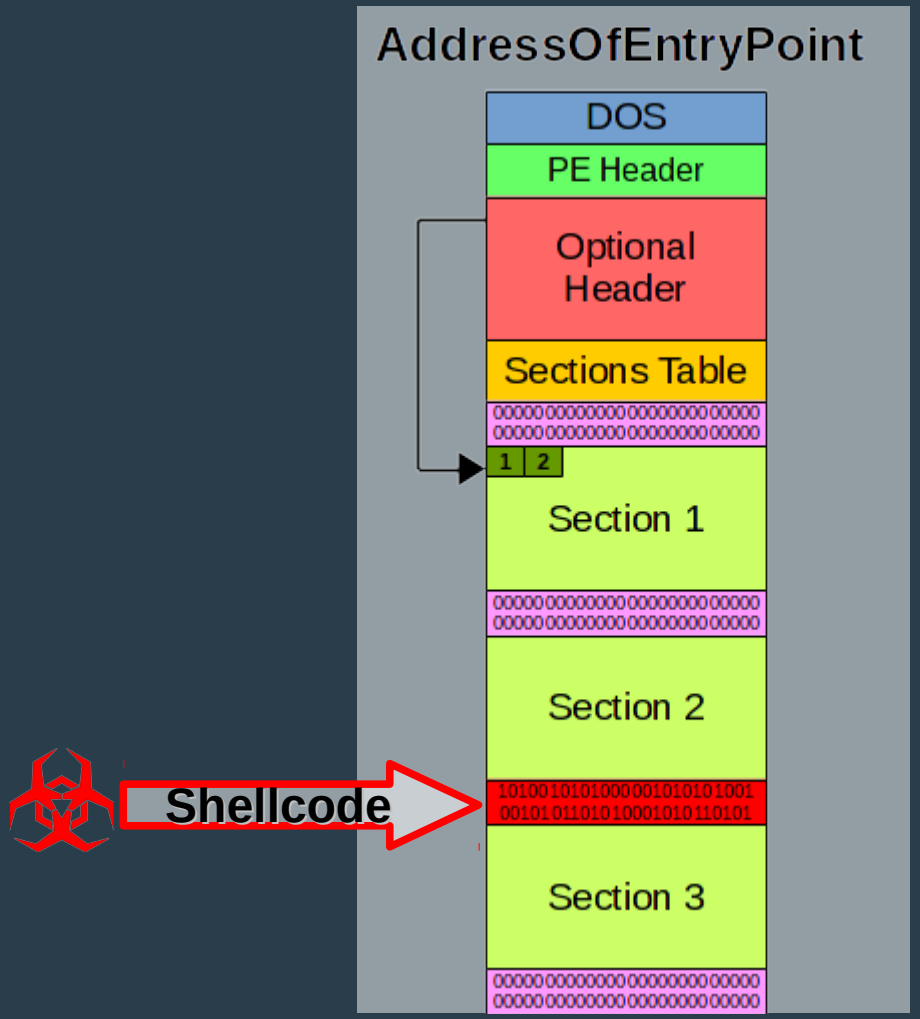

## Three Ways to Inject

- Single Cave
- Multiple Caves
- Adding a Section

# Portable Executable

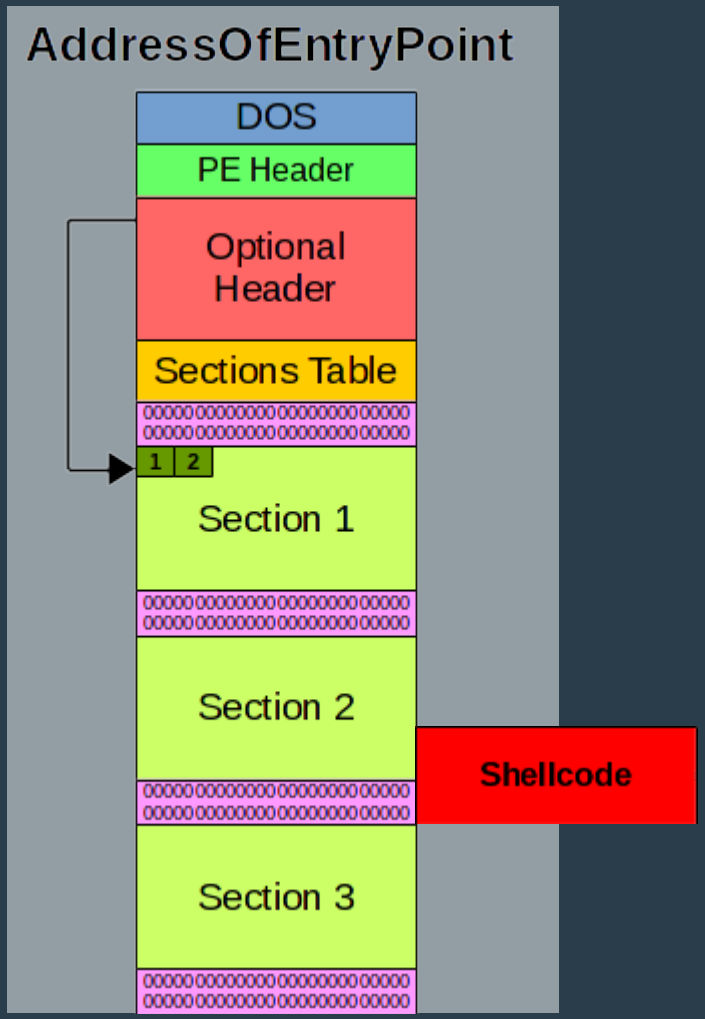


# Portable Executable



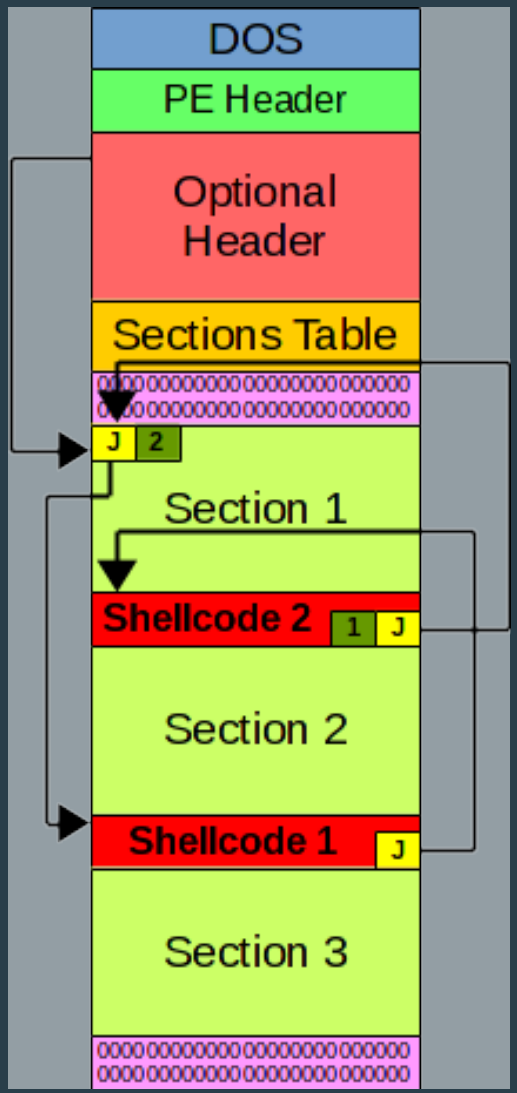
# Portable Executable

# PE – Multiple Caves



# PE – Multiple Caves

# PE – Multiple Caves



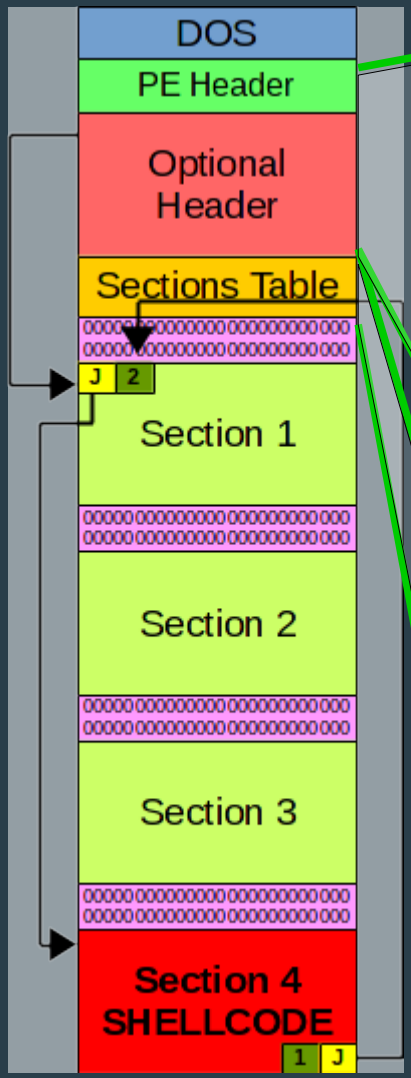
# How does AV work?

```
11101011 00011011 01011011 00110001 11000000 10001001 01000011
00001000 01010000 01010011 10001001 11100001 10001001 11001010
10000011 11000010 00000100 10000011 11000000 00001011 11001101
10000000 10000011 11101000 00001010 00110001 11011011 11001101
10000000 11101000 11100000 11111111 11111111 11111111 00101111
01100010 01101001 01101110 00101111 01110011 01101000
```

**Definizioni**

| Virus | Signature |
|---|---|
| XYZ | 11000000 00001011 11001101 10000000 10000011 |

# PE – Multiple Caves




64 bit

| | | | |
|---|---|---|---|
| 0x0000 | Signature 0x50450000 | Machine | #NumberOfSections |
| 0x0008 | TimeDateStamp | PointerToSymbolTable (deprecated) | |
| 0x0010 | # NumberOfSymbolTable (deprecated) | SizeOfOptionalHeader | Characteristics |
| 0x0018 | Magic / MajorLinker Version / MinorLinker Version | SizeOfCode (sum of all sections) | |
| 0x0020 | SizeOfInitializedData | SizeOfUninitializedData | |
| 0x0028 | AddressOfEntryPoint (RVA) | BaseOfCode (RVA) | |
| 0x0030 | BaseOfData (RVA) | ImageBase | |
| 0x0038 | SectionAlignment | FileAlignment | |
| 0x0040 | MajorOperating SystemVersion / MinorOperating SystemVersion | MajorImage Version | MinorImage Version |
| 0x0048 | MajorSubsystem Version / MinorSubsystem Version | Win32VersionValue (zeros filled) | |
| 0x0050 | SizeOfImage | SizeOfHeaders | |
| 0x0058 | CheckSum (images doesn't checked) | Subsystem | DllCharacteristics |
| 0x0060 | SizeOfStackReserve | SizeOfStackCommit | |
| 0x0068 | SizeOfHeapReserve | SizeOfHeapCommit | |
| 0x0070 | LoaderFlags (zeros filled) | # NumberOfRvaAndSizes | |

COFF Header

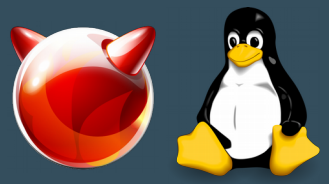
Standad COFF Fields

Windows Specific Fields

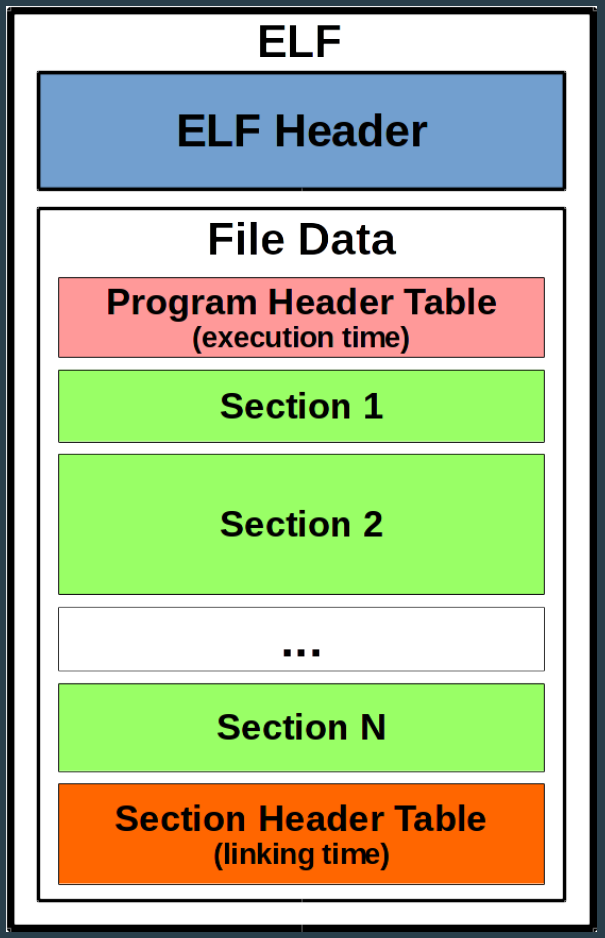
64 bit

| | |
|---|---|
| Name | |
| VirtualSize | VirtualAddress (RVA) |
| SizeOfRawData | PointerToRawData |
| PointerToRelocations | PointerToLinenumbers |
| NumberOfRelocations / NumberOfLinenumbers | Characteristics |

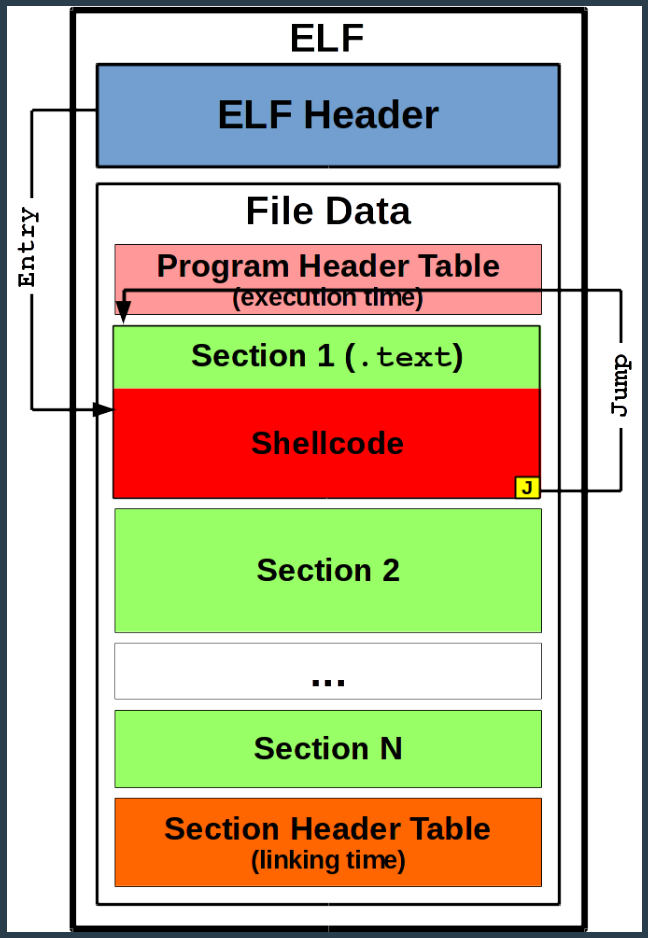
Section Table



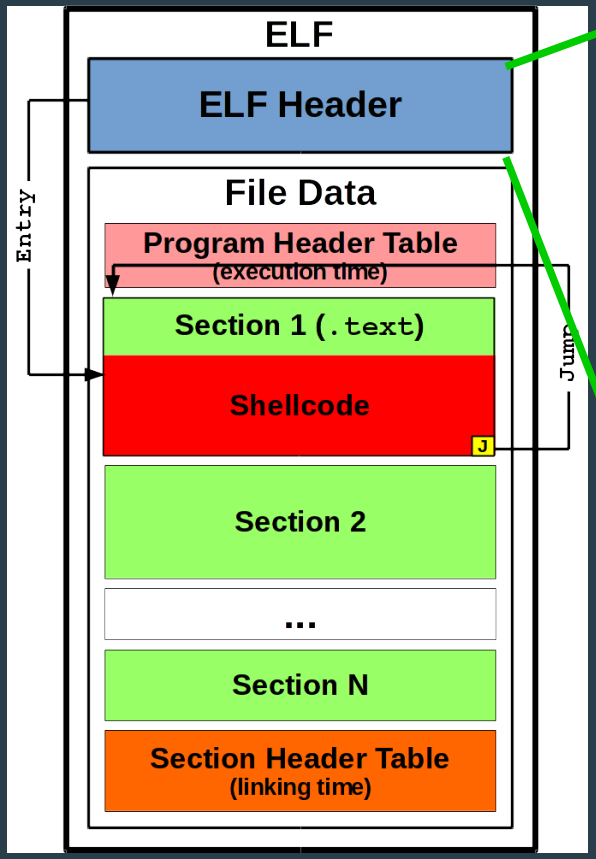
# ELF

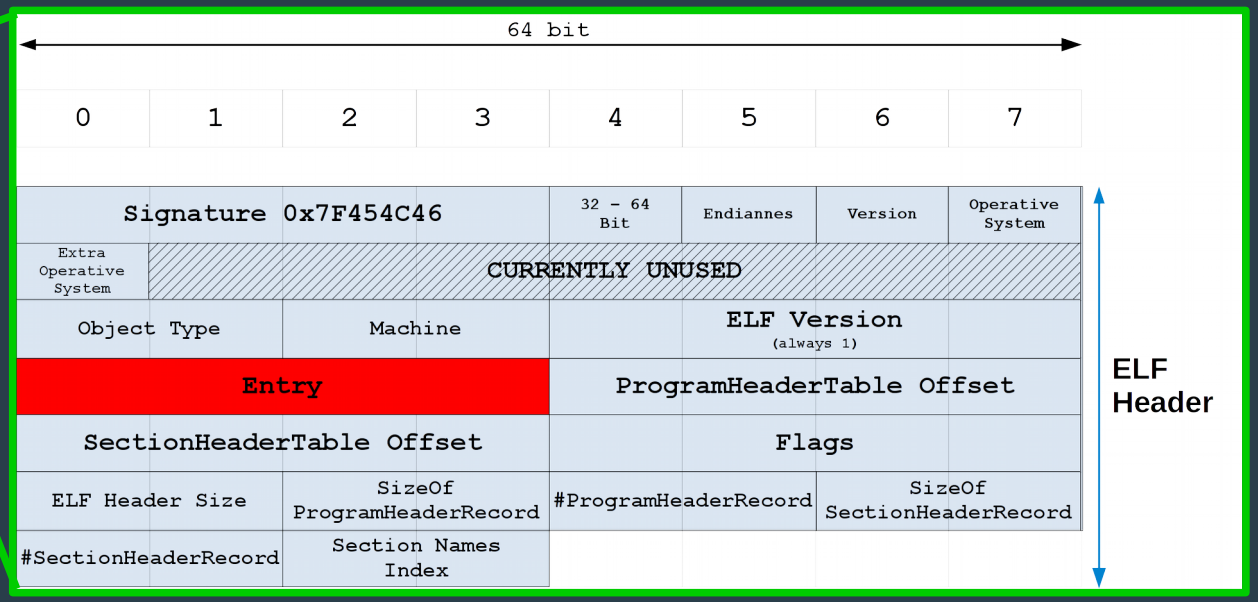

ELF
ELF Header
File Data
Program Header Table
(execution time)
Section 1
Section 2
...
Section N
Section Header Table
(linking time)

# ELF

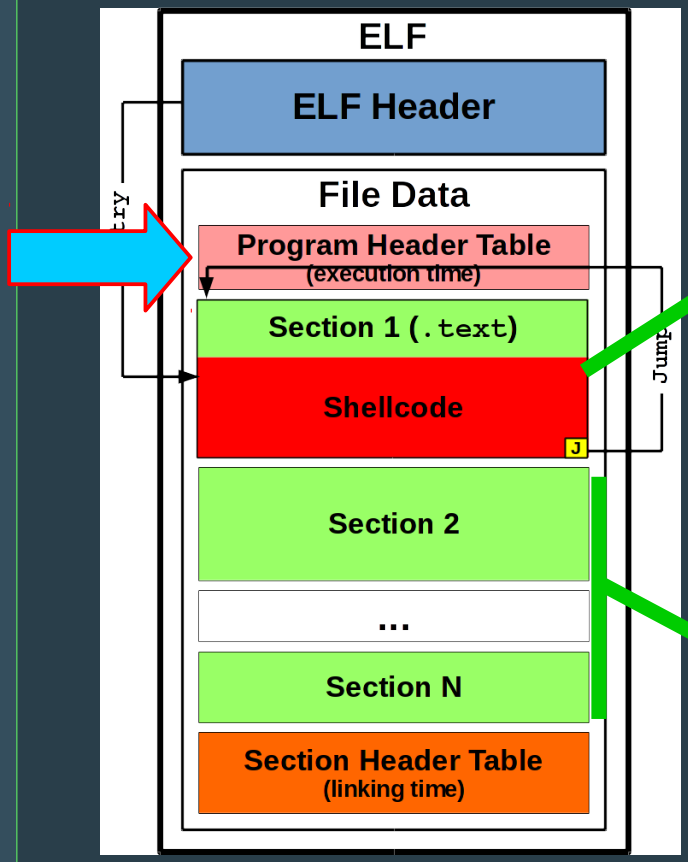

ELF
ELF Header
File Data
Program Header Table
(execution time)
Section 1 (.text)
Shellcode
J
Section 2
...
Section N
Section Header Table
(linking time)
Entry
Jump

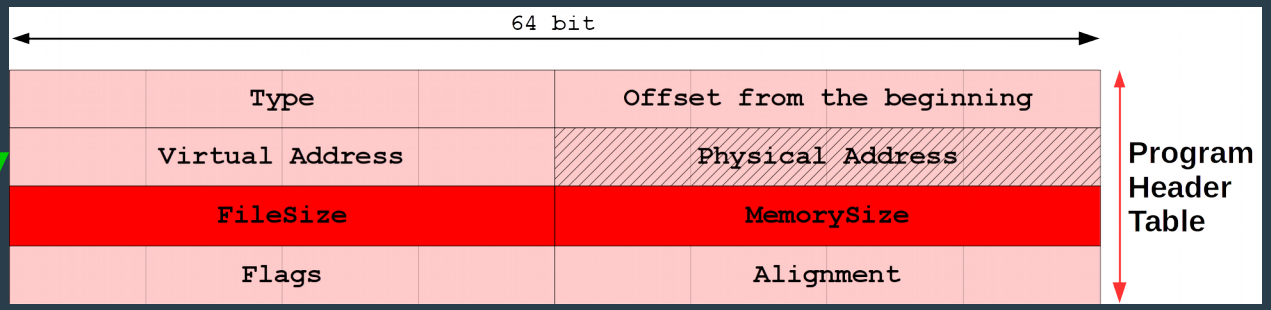
# ELF

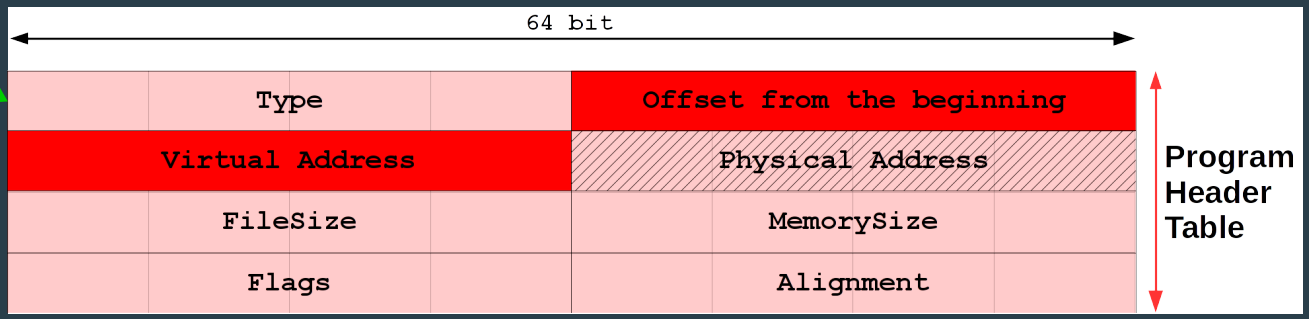

ELF
ELF Header
File Data
Program Header Table
(execution time)
Section 1 (.text)
Shellcode
J
Section 2
...
Section N
Section Header Table
(linking time)
Entry
Jump


64 bit
0 1 2 3 4 5 6 7
Signature 0x7F454C46
32 - 64 Bit
Endiannes
Version
Operative System
Extra Operative System
CURRENTLY UNUSED
Object Type
Machine
ELF Version
(always 1)
Entry
ProgramHeaderTable Offset
SectionHeaderTable Offset
Flags
ELF Header Size
SizeOf ProgramHeaderRecord
#ProgramHeaderRecord
SizeOf SectionHeaderRecord
#SectionHeaderRecord
Section Names Index
ELF Header

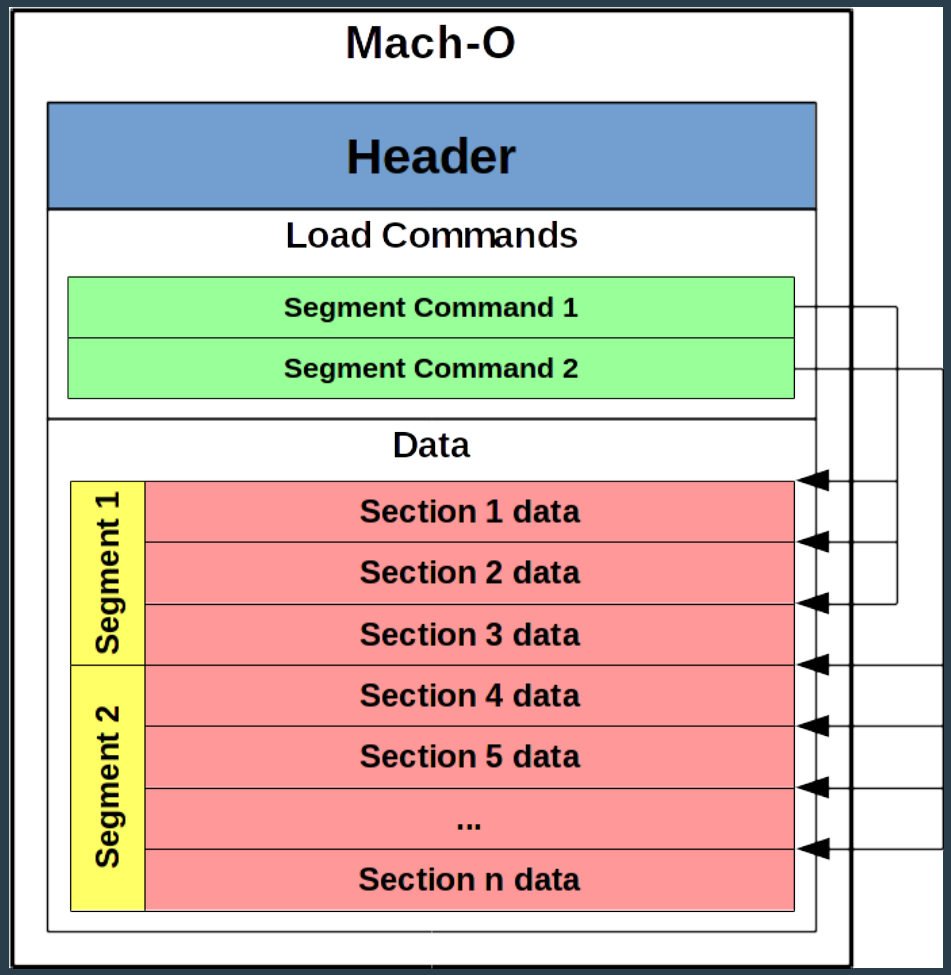
# ELF


ELF
ELF Header
File Data
Program Header Table
(execution time)
Section 1 (.text)
Shellcode
J
Section 2
...
Section N
Section Header Table
(linking time)

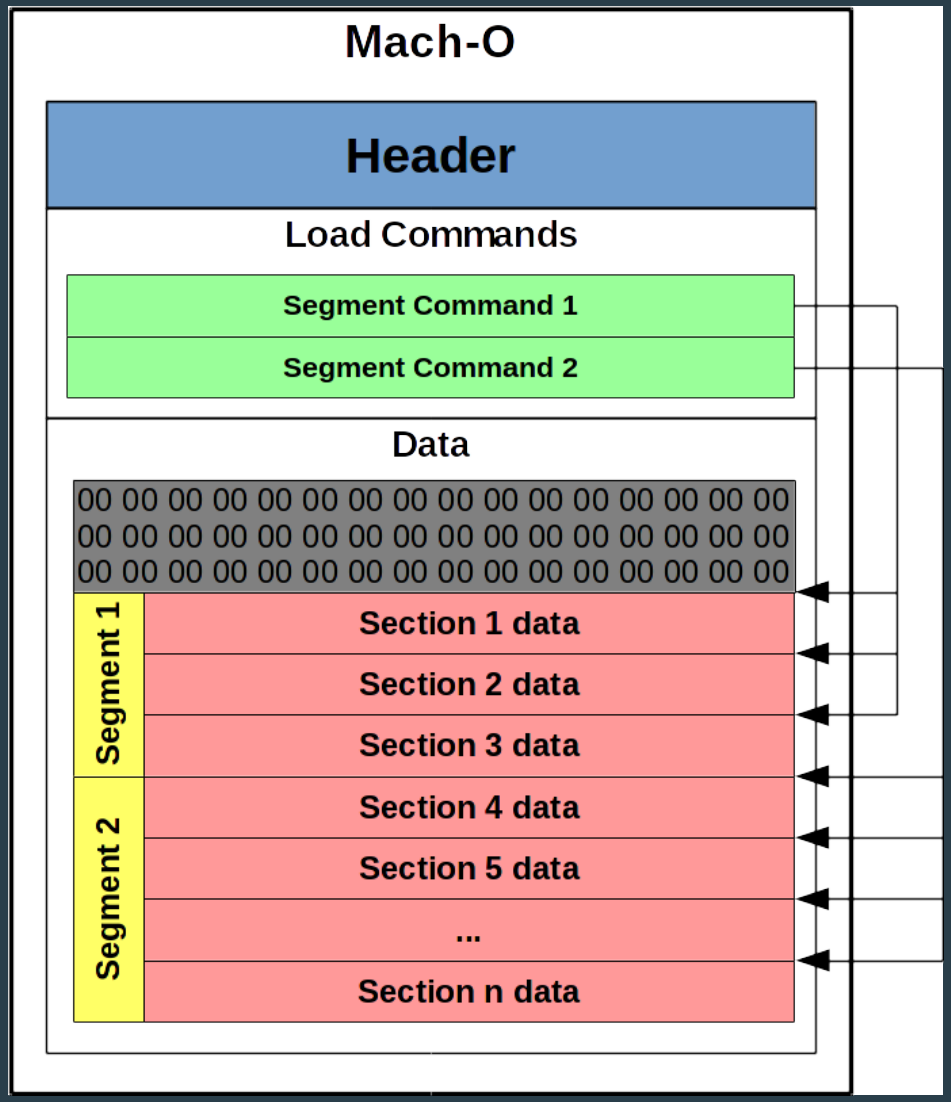

64 bit
Type
Offset from the beginning
Virtual Address
Physical Address
FileSize
MemorySize
Flags
Alignment
Program Header Table


64 bit
Type
Offset from the beginning
Virtual Address
Physical Address
FileSize
MemorySize
Flags
Alignment
Program Header Table

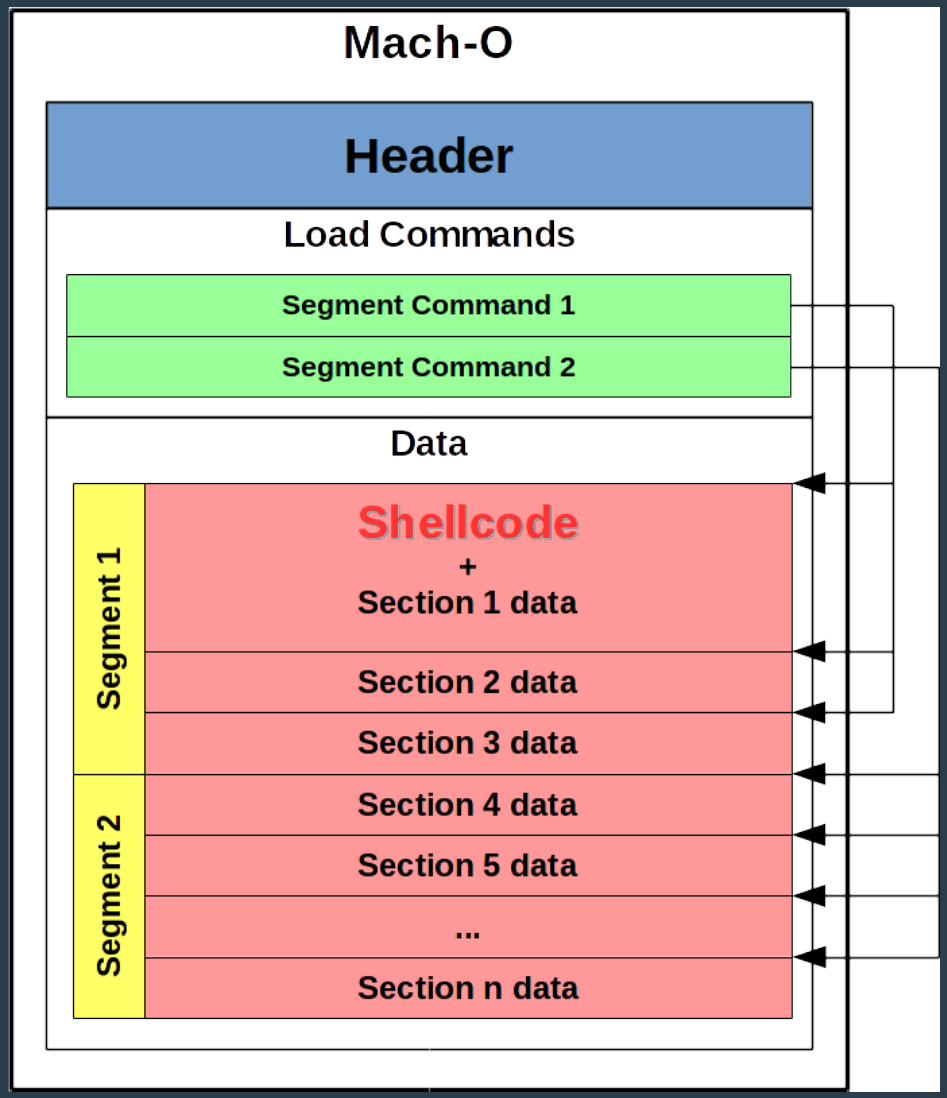
# Mach-O


Mach-O
Header
Load Commands
Segment Command 1
Segment Command 2
Data
Segment 1
Segment 2
Section 1 data
Section 2 data
Section 3 data
Section 4 data
Section 5 data
...
Section n data

# Mach-O


Mach-O
Header
Load Commands
Segment Command 1
Segment Command 2
Data
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Segment 1
Section 1 data
Section 2 data
Section 3 data
Segment 2
Section 4 data
Section 5 data
...
Section n data

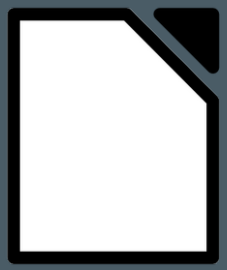
# Mach-O


Mach-O
Header
Load Commands
Segment Command 1
Segment Command 2
Data
Segment 1
Shellcode
+
Section 1 data
Section 2 data
Section 3 data
Segment 2
Section 4 data
Section 5 data
...
Section n data

# THANKS FOR THE ATTENTION

**Thesis**
**Presentation**
**Charts**

**Are made just using**
**Free Software**



LibreOffice
The Document Foundation

**www.libreoffice.org**