AMIRHOSSEIN AMIRAN

Malware analysis and review

Malware Writing and Analastic

!=

Reverse Engineering

MALWARE TYPES

- Rootkit
- Ransomware
- Worms
- Spyware
- Cryptojackers
- RATs
- Wipers
- Polymorphic Malware
- Bootkit
- Viruses
- Trojans
- Backdoors
- Adware
- Logic Bombs
- IoT Malware

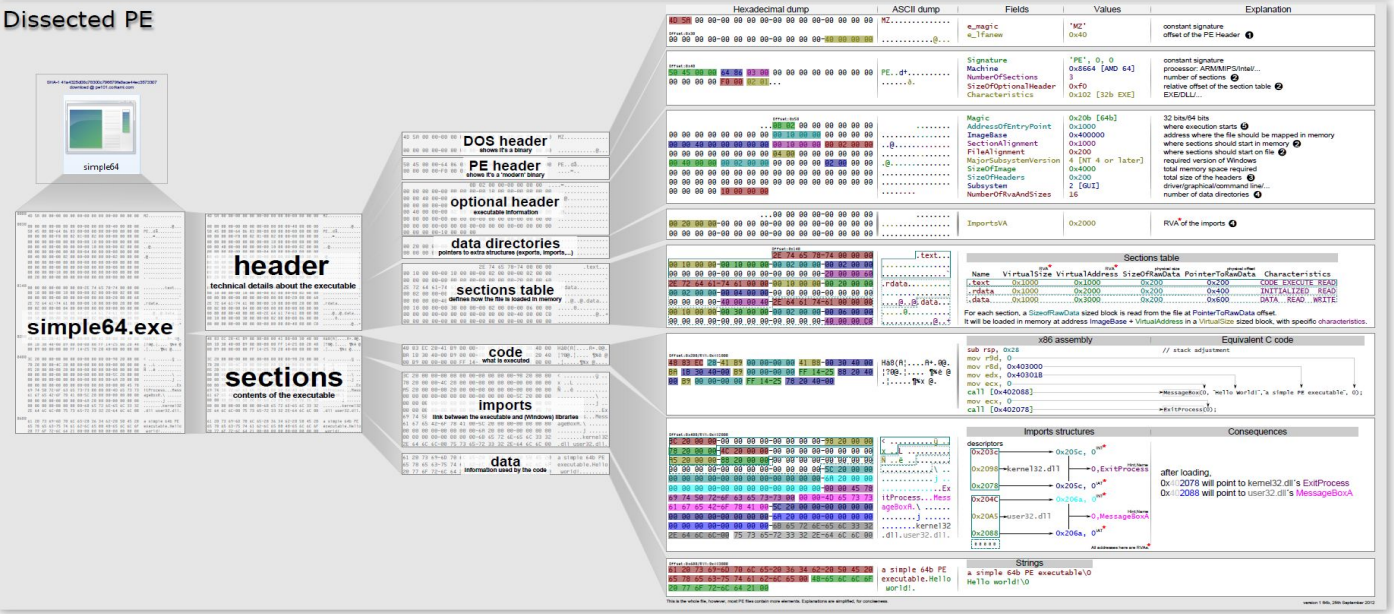
STUXNET

THE NEW ERA

PE101 a windows executable walkthrough (64bits)

Ange Albertini
corkami.com

Dissected PE

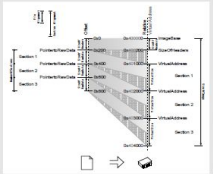


Loading process

1 Headers
the *DOS Header* is parsed
the *PE Header* is parsed
(its offset is *DOS Header*'s e_lfanew)
the *Optional Header* is parsed
(it follows the *PE Header*)

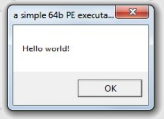
2 Sections table
Sections table is parsed
(it is located at: offset (*OptionalHeader*) + *SizeOfOptionalHeader*)
it contains *NumberOfSections* elements
it is checked for validity with alignments:
FileAlignments and *SectionAlignments*

3 Mapping
the file is mapped in memory according to:
the *ImageBase*
the *SizeOfHeaders*
the Sections table



4 Imports
DataDirectories are parsed
they follow the *OptionalHeader*
their number is *NumOfRVAAndSizes*
imports are always #2
Imports are parsed
each descriptor specifies a *DLLname*
this DLL is loaded in memory
IAT and *INT* are parsed simultaneously
for each API in *INT*
its address is written in the *IAT* entry

5 Execution
Code is called at the *EntryPoint*
the calls of the code go via the IAT to the APIs



Notes

MZ HEADER aka DOS_HEADER
Starts with 'MZ' (initials of *Mark Zbikowski* MS-DOS developer)
PE HEADER aka IMAGE_FILE_HEADERS / COFF file header
Starts with 'PE' (Portable Executable)
OPTIONAL HEADER aka IMAGE_OPTIONAL_HEADER
Optional only for non-standard PEs but required for executables
RVA Relative Virtual Address
Address relative to ImageBase (at ImageBase, RVA = 0)
Almost all addresses of the headers are RVAs
In code, addresses are *not* relative.

INT Import Name Table
Null-terminated list of pointers to Hint, Name structures
IAT Import Address Table
Null-terminated list of pointers
On file it is a copy of the INT
After loading it points to the imported APIs
HINT
Index in the exports table of a DLL to be imported
Not required but provides a speed-up by reducing look-up

What do we want to do now?

With ***RUST***

Thanks For Your Attention