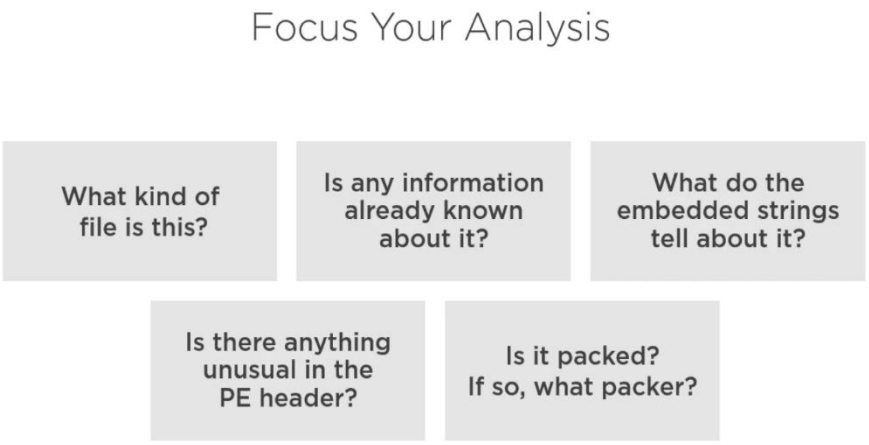
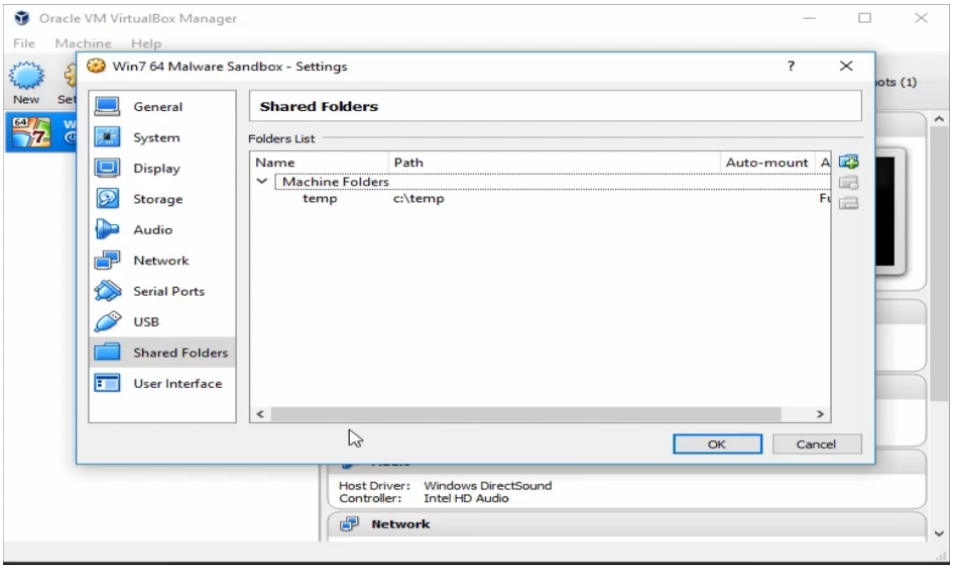


Firstly get the md5 or sha256 and go to virustotal to check it

1. Use floss as a tool instead of strings it makes pch of staff like decrypting and simplifies the things and at bottom section prints the conclusion
2. Always use .malz extension after .exe to dissarm the malware
3. Start basic static analysis using pe view or pe studio it is the strongest and helps alot
4. Check the time date stamp at image file header it gives us the date of compilation of the virus and it is helpful sometimes it may be designed to always give you the same date and sometimes it will give you 1990s dates that indicates the Balkans delfi compiler
5. Check IMAGE_SECTION_HEADER.text to check for the size of raw data and virtual size with comparing them if there is a big difference you would expect it is a packed malware
6. Go to SECTION.rdata you will find the IAT IMPORT ADRESS TABLE it contains the windows apis the malware is running and to check them you can go and see
   ttps://malapi.io
7. Check always first bytes of file to know its type MZ means Executable
8. Check for libraries of dll used something like urlmon.dll which is from OLE32 Extensions for Win32
   It is object linking and embedding and it allows different microsoft office components to talk with each others it is commonly used by hackers or wininet.dll internet extension for windows dll which isn't malicious itself but it is usually used by hackers
9. In pe studio look also at strings as it gets the best guess of what that string is doing

10. 

11. If you set a shared folder then make it a folder that you dont care for things in it as if you are analyzing a ransomware

You may lose all that file data cuz of that ransomware if you dont want shared folder you can use network protocol like ftp
Or use a usb that out vm will access

12. File Identification

Determine the type of file you are examining

Attackers try to disguise files
- Double extensions
- Archives

Packer detection

Main questions to answer in the static analysis phase

Important note even in static analysis phase there are some tools
That needs to execute the malware to collect needed info so always do it on the vm
And note so tools are backdoors and malware themselves so always analyze your tools

They use a double extension as they put an extension that makes it dont look like an executable
While it is really an exec ...
Archives : they try to put their malwares in an archives such as winrar that has its own programming
Language it makes the archive contains the malware and a script that runs by opening the archive
That will launch the malware

To make sure it is started when system is rebooted

Malware will try to resolve to a well known name if it is connected then it is not a sandbox

Uppercase letters lower case letters numbers plus sign and forward slash is important to see in the file it indicates base64 encoding

Means malware has anti analysis ability

Because xor is 00 means it is not hidden

20 and 27 are to switch case

Those are my knowledge from three different courses :

PMAT-TCM

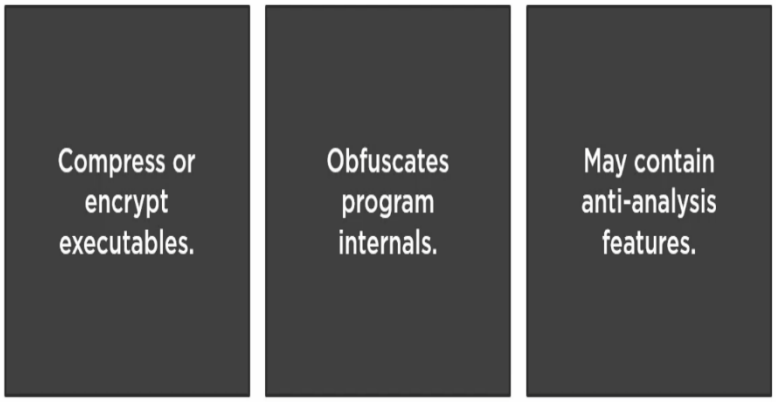
Malware Analysis Fundamentals-Pluralsight

Malware Analysis Fundamentals-Maharatech

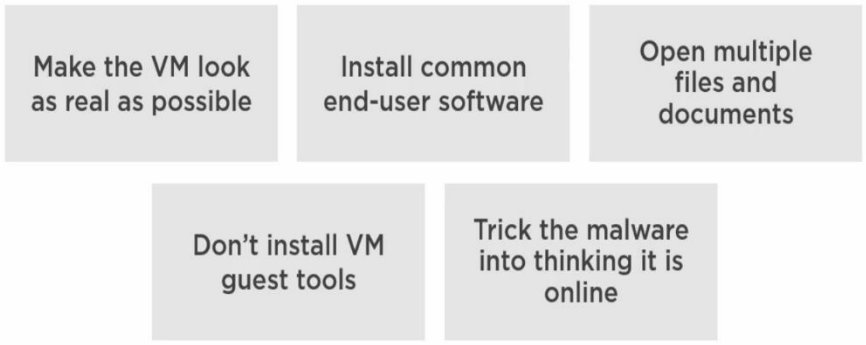
ENG : Mohamed Ewies

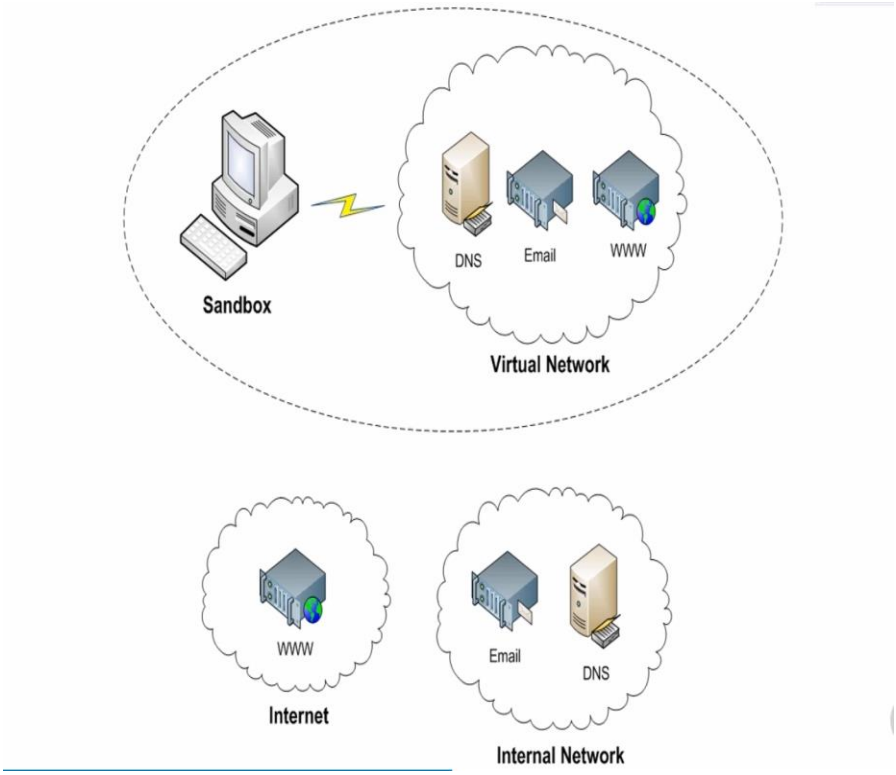
/https://www.linkedin.com/in/mohamed-ewies-59b89a212

Packers



Hiding Your Virtual Machine





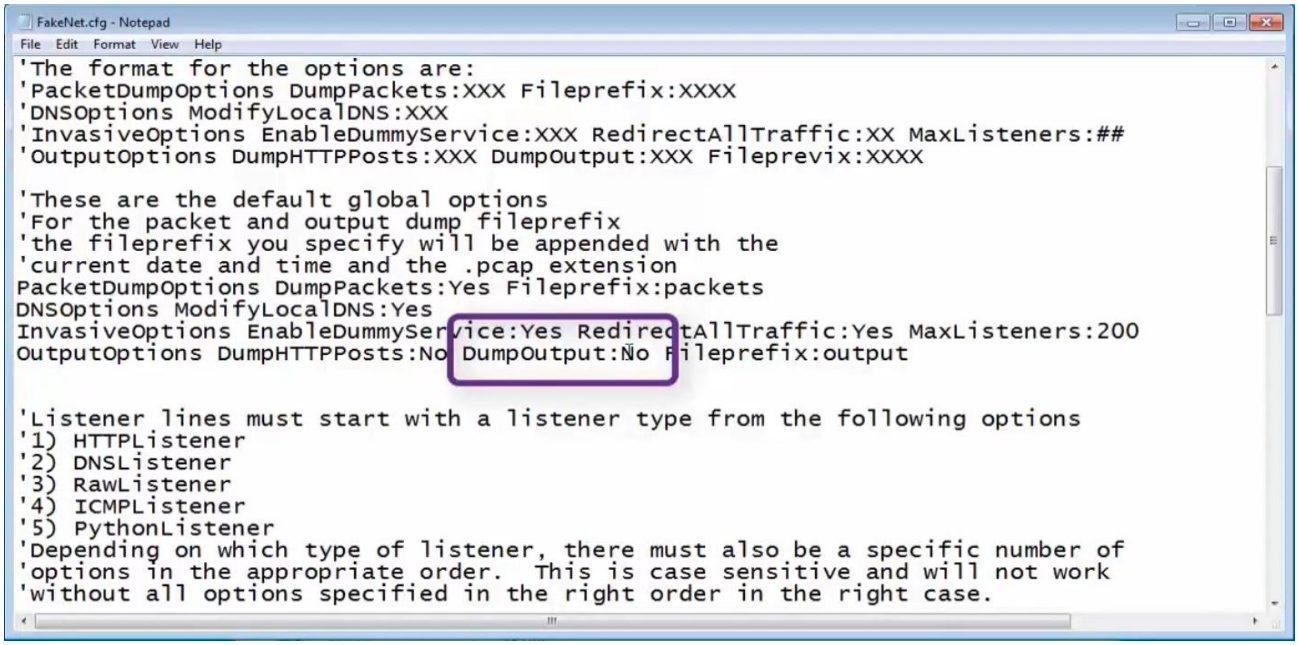
FakeNet

Tricks malware into thinking it's online

Intercepts on HTTP, SMTP, and more

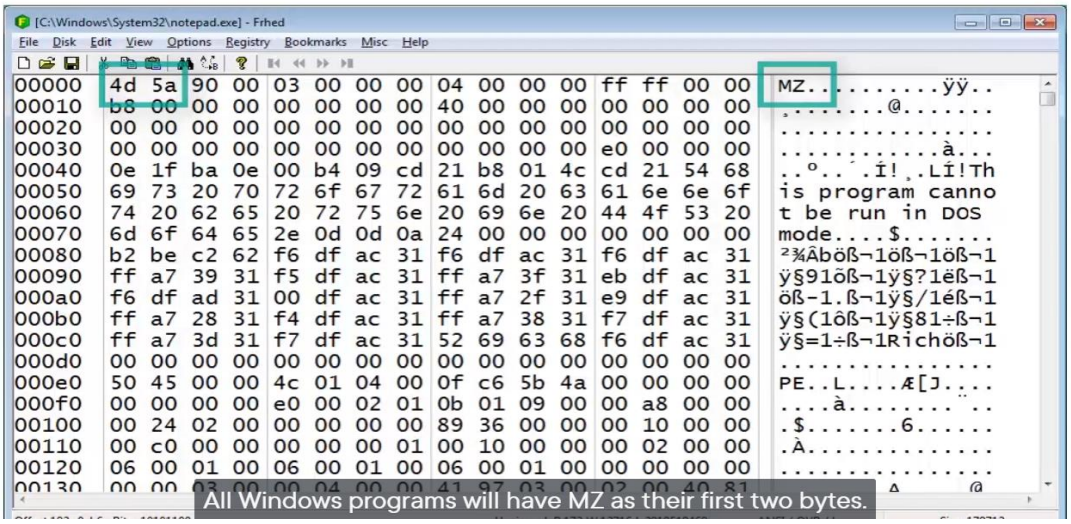
Logs connections and PCAP

https://practicalmalwareanalysis.com/fakenet/

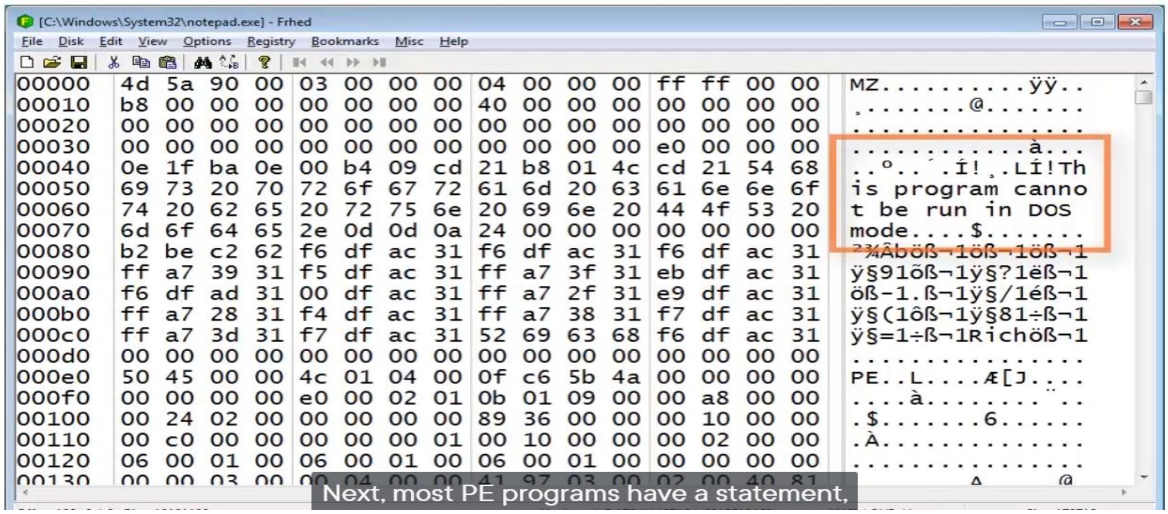


Fakenet by default wont log network connections so you have to set it up and make it LOG change that to yes and save it make sure you run it as ADMINSTRATOR

File Signatures

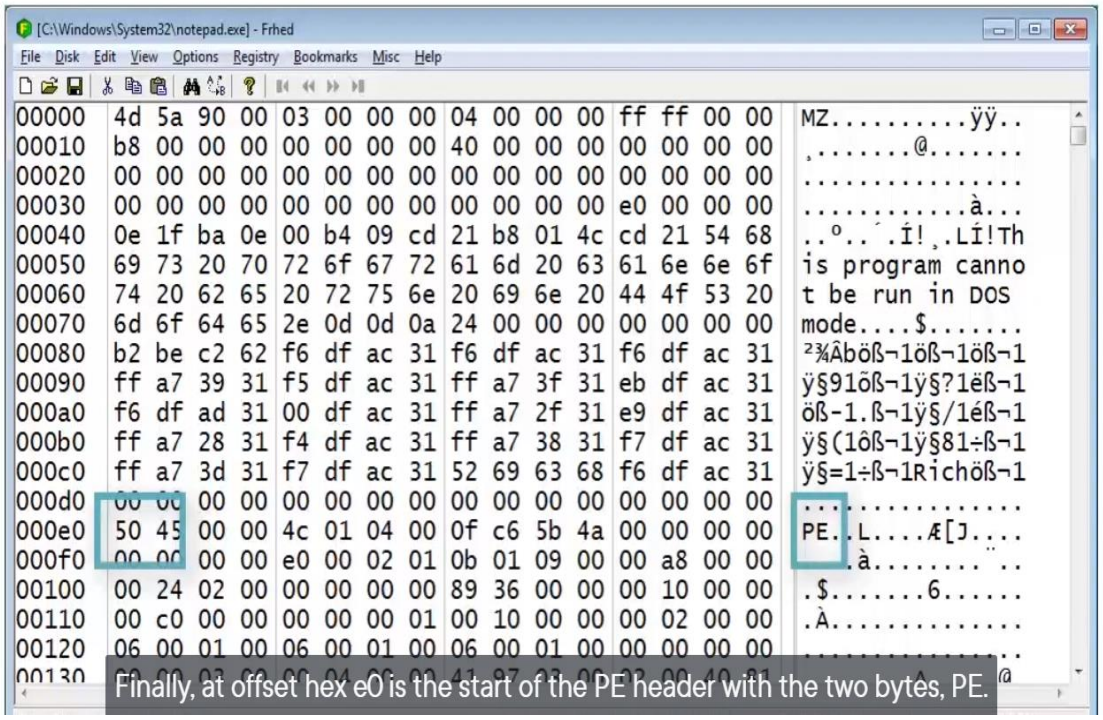


File Signatures



That statement not necessary and some packers might change it or remove it

File Signatures



File Signatures You Should Know

Windows Executable	"MZ" in bytes 0-1
PDF	"%PDF-" followed by number within first 1024 bytes
Old Office Doc	0xD0CF11E0 ("DOCFILE")
Zip Archive	"PK" in bytes 0-1

File

Compares file header with known signatures

but does not provide much in-depth information on the file you are looking at.

Exeinfo PE

Analyzes Windows PE header information

Packer detection

Gives hints on how to unpack

http://exeinfo.atwebpages.com/

TrID

Uses pattern database to determine type

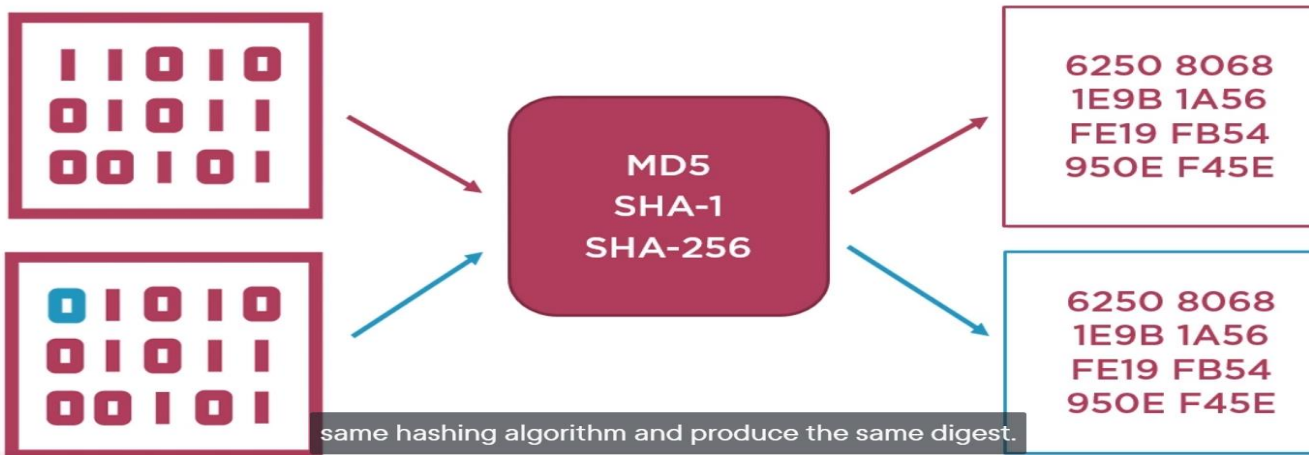
Gives likelihood of detected type

http://mark0.net/soft-trid-e.html

Options

-d:PATH : Path to pattern database

Collisions



Using Cryptographic Hashes

Organize and identify specific samples

Hash with multiple algorithms

Find additional online information

Embedded strings analysis extracts and examines groups of readable characters from a file.

Types of Strings

ASCII

7-bits long

128 characters

45	78	70	6C	6F	72	65	72	00
E	x	p	l	o	r	e	r	.

Unicode

8-, 16-, 32-bit characters

Over 128,000 characters

0045		0078		0070		006C		006F		0072		0065		0072		0000	
.	E	.	x	.	p	.	l	.	o	.	r	.	e	.	r	.	.

you need to make sure your tool is extracting both ASCII and Unicode strings,

Strings

Extracts ASCII and Unicode at the same tim

Minimum string length: 3

https://technet.microsoft.com/en-us/sysinternals/strings.aspx

Options

-n # : Set minimum string length

-o : Print decimal offset of string

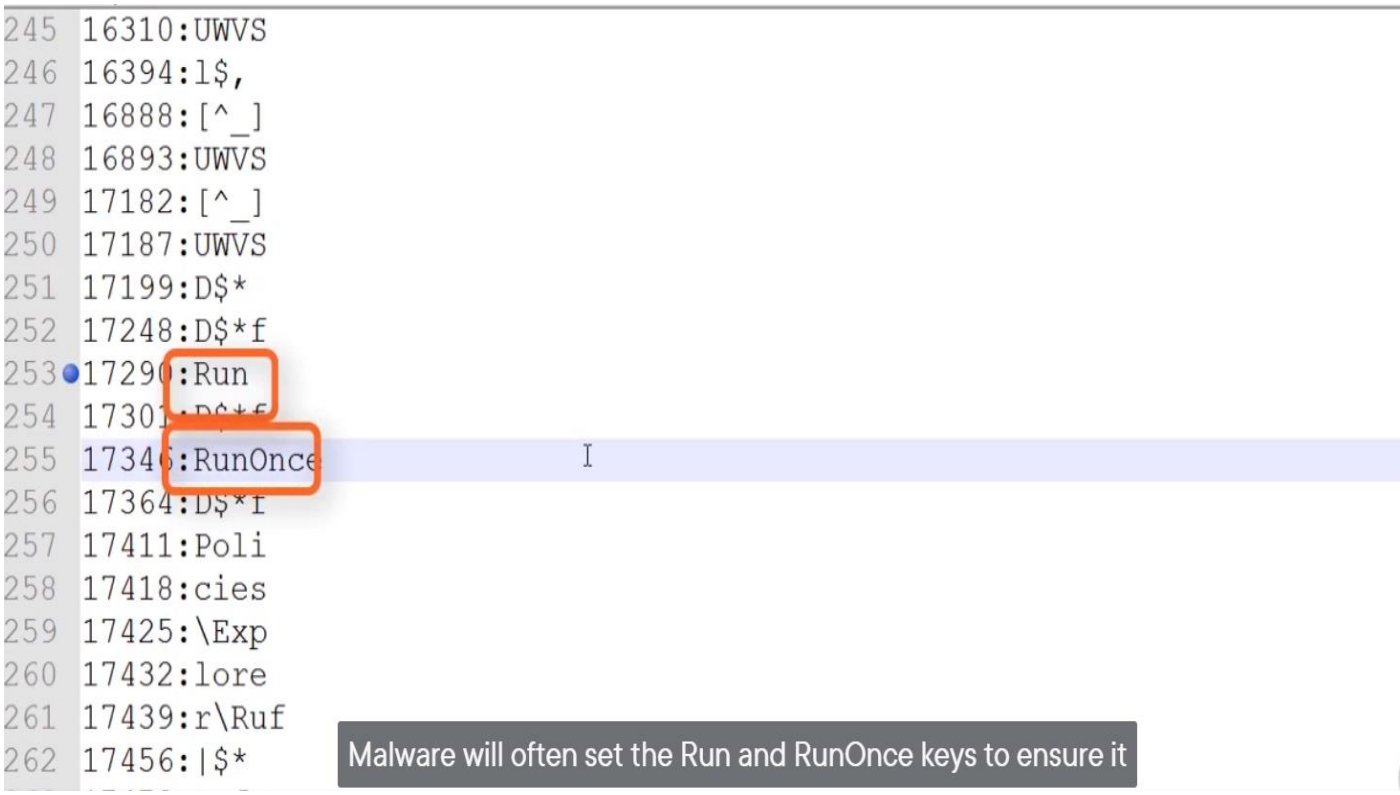
BinText

Extracts ASCII and Unicode at the same ti

Minimum string length: 5

Watch out for the repeating bug!

https://www.mcafee.com/us/downloads/free-tools/bintext.aspx



```
991  55212:LID
992  55222:RESP
993  55232:ONSE
994  55325:[^_]
995  55448:D$(
996  55461:D$(
997  55469:L$)
998  55523:gfff
999  55653:update.microsoft.com
1000 55690:D$$
1001 55708:www.update.microsoft.com
1002 55753:D$(
1003 55771:wind
1004 55779:owsu
1005 55787:pdat
1006 55795:e.mi
1007 55800:D$ cros
1008 55808:D$$oft.
```

If it does, then the malware knows it isn't in a sandbox.

strings.txt

```
2625 179412:%s:%i
2626 179418:%s\browser%li.html
2627 179440:<meta http-equiv="refresh" content="%i">
2628 179484:%s "%s"
2629 179496:%s@%s:%i
2630 179508:%s--%s
2631 179516:%s:%i
2632 179524:%s--%s
2633 179532:ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
2634 179600:%s%02X
2635 179607:%02X
2636 179614:Shell_TrayWnd
2637 179633:.com
2638 179638:.exe
2639 179643:.scr
2640 179648:.vbs
2641 179653:.cmd
2642 179661:LOCALAPPDATA
```

it's good to know that the malware is probably doing some Base64 coding.

Normal text file length : 66,554 lines : 4,650 Ln : 2,633 Col : 1 Sel : 0 | 0 Windows (CR LF) ANSI

```
2742 181457:recv
2743 181462:sendto
2744 181469:inet_ntoa
2745 181479:inet_addr
2746 181490:rpcrt4.dll
2747 181512:UuidCreate
2748 181523:UuidToStringA
2749 181537:RpcStringFreeA
2750 181552:The Wireshark Network Analyzer
2751 181583:autoruns.exe
2752 181596:wuauclt.exe
2753 181608:%s@%s
2754 181616:SbieDll.dll
2755 181628:snxhk.dll
2756 181638:dbghelp.dll
2757 181652:SOFTWARE\Microsoft\Windows NT\CurrentVersion
2758 181697:ProductId
2759 181707:
```

These are all programs that malware analysts might use during malware analysis.

Packers and Strings

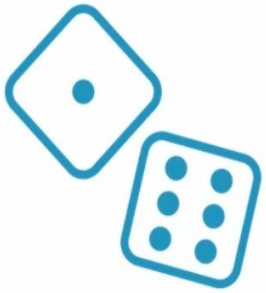
```
!This program cannot be run in DOS mode.
`.data
@.reloc
wdl.dll
 -s -h
attrib
CMD.EXE /C
COMMAND.COM /C
wumsvc.dll
ServiceDll
Parameters
SYSTEM\CurrentControlSet\Services\
%SystemRoot%\System32\svchost.exe -k nets
%SystemRoot%\System32\svchost.exe -k netsvcs
netsvcs
SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost
idel0.dll
DLL_FILE
```hhh
xppwpp
c:\lstudio\projects\config007\insconfig\objfre_wxp_x8
```

Unpacked

```
!This program cannot be run in DOS mode.
P]%_-]
P]%_+]
P]%_=]E
P]%_>]
P]%_.]
P]%_,]
P]%_(]
P]Rich
attrib
CMD.EXE /C
wumsvcB
\;ServiceD
Paramet
YSTEM\CurrentCo
,t\0s<%Sys-mR
```

Packed
```

String Hiding Tricks



Random Strings

Packers may add random or legitimate looking strings to trick you.

Encrypted Strings

Strings may be encoded or encrypted.

Base64 encoding and XOR encryption.

Base64 Encoding

Converts 3 bytes to 4 ASCII characters

A-Za-z0-9+/ are the default characters used

Strings padded to length of 4 with "="

All Base64 encoded strings must be of a length divisible by four.

notice how the encoded string changes.

Detecting Base64 Strings

Base64 alphabet

```
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
```

Random ASCII strings that end in =

```
Q29tbWFuZHMgZnJvbSB0aGUgd2ViIHNlcnZlciBzaG91bGQgcmVzcG9uZCBpbiB0aGUgZmFzaGlvbj
oKCjxIVFRQIEhFQURFUj4KXHJcblxyXG4KPGNvbW1hbmQ+IDxhcmd1bWVudHM+Cgp3aGVyZSBcclxu
XHJcbiBhcmUgdGhlIGNhcnJpYWdlIHJldHVybiwgbGluZSBmZWVkcwoK==
```

but only have the Base64 alphabet characters in them,

Decoding Base64 Strings

```
doskey atob=powershell
"[Text.Encoding]::UTF8.GetString([convert]::FromBase64String(\"$*\"))"
```

```
C:\>doskey atob=powershell "[Text.Encoding]::UTF8.GetString([convert]::FromBase6
4String(\"$*\"))"

C:\>atob SGVsbG8gV29ybGQh
Hello World!
```

XOR

XOR Truth Table		
Input		Output
0	0	0
0	1	1
1	0	1
1	1	0

$$A \oplus K = C$$

$$C \oplus K = A$$

$$A \oplus 0 = A$$

$$A \oplus A = 0$$

XOR Encoding

E	x	p	l	o	r	e	r	.
45	78	70	6C	6F	72	65	72	00
			⊕ 65	⊕ 65	⊕ 65	⊕ 65	⊕ 65	⊕ 65
20	1D	15	09	0A	17	00	17	65

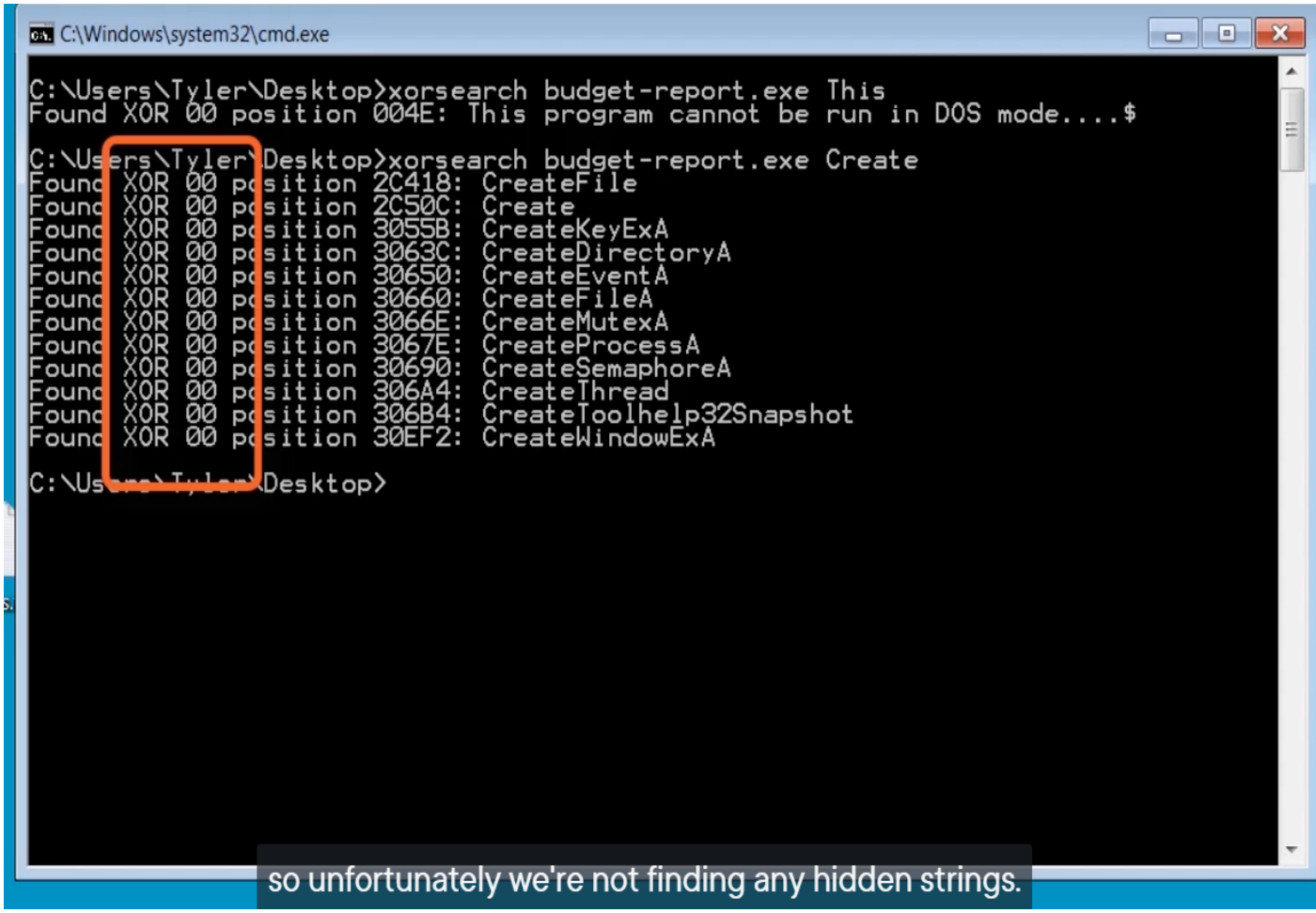
xorsearch

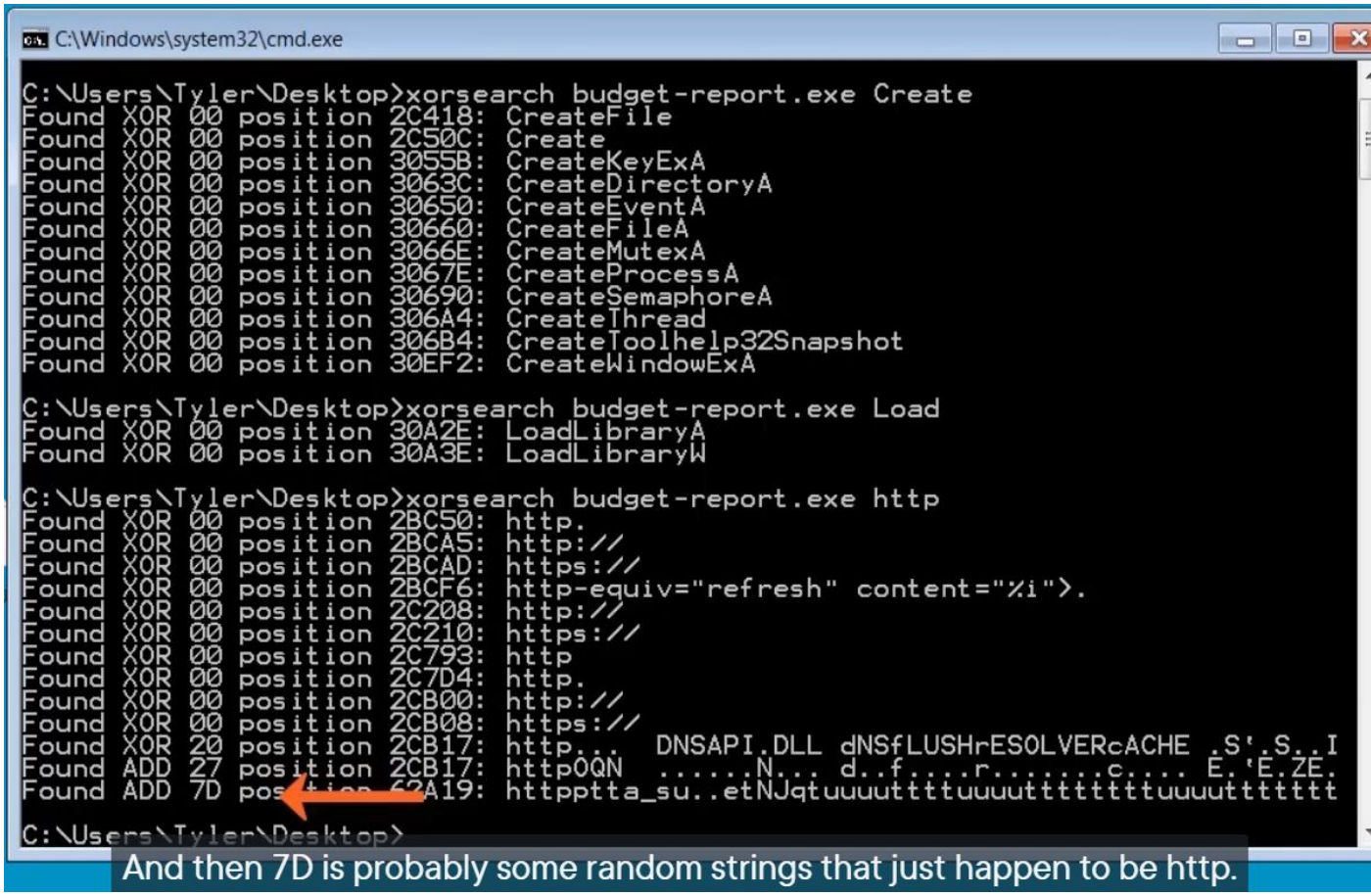
Searches for XOR encoded version of string

https://blog.didierstevens.com/programs/xorsearch/

Strings to Search For
This
http
kernel
Create
~~www~~

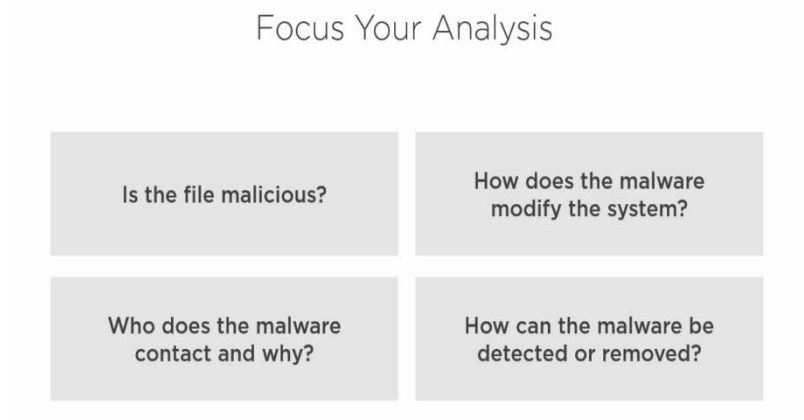
One string that is not good to search for is www.


C:\Windows\system32\cmd.exe
C:\Users\Tyler\Desktop>xorsearch budget-report.exe This
Found XOR 00 position 004E: This program cannot be run in DOS mode....$
C:\Users\Tyler\Desktop>xorsearch budget-report.exe Create
Found XOR 00 position 2C418: CreateFile
Found XOR 00 position 2C50C: Create
Found XOR 00 position 3055B: CreateKeyExA
Found XOR 00 position 3063C: CreateDirectoryA
Found XOR 00 position 30650: CreateEventA
Found XOR 00 position 30660: CreateFileA
Found XOR 00 position 3066E: CreateMutexA
Found XOR 00 position 3067E: CreateProcessA
Found XOR 00 position 30690: CreateSemaphoreA
Found XOR 00 position 306A4: CreateThread
Found XOR 00 position 306B4: CreateToolhelp32Snapshot
Found XOR 00 position 30EF2: CreateWindowExA
C:\Users\Tyler\Desktop>
so unfortunately we're not finding any hidden strings.

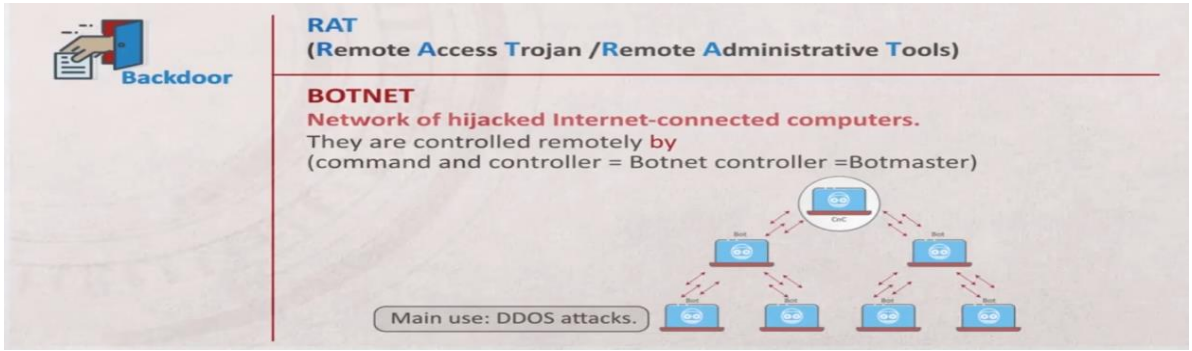


My main goals in malware analysis from a to z is to answer those questions

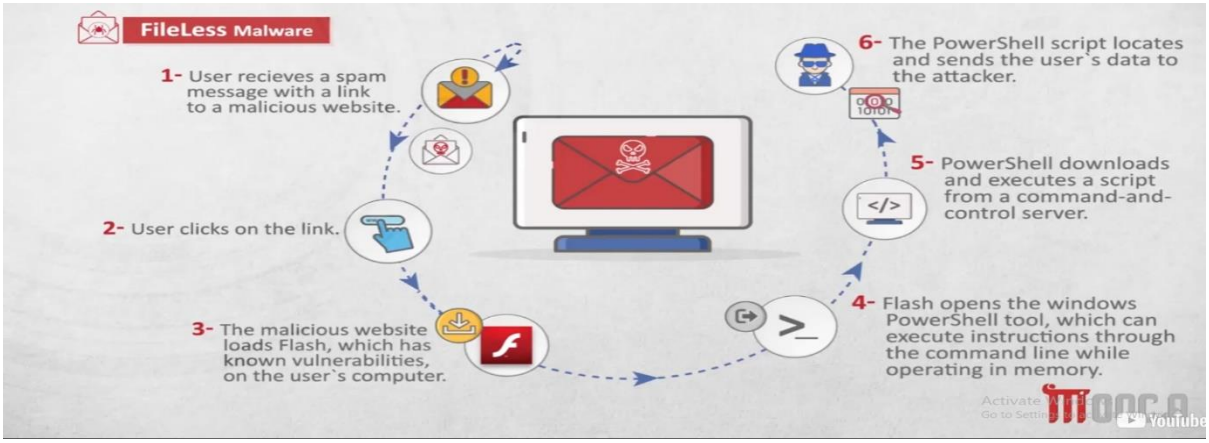
Basic Static Analysis

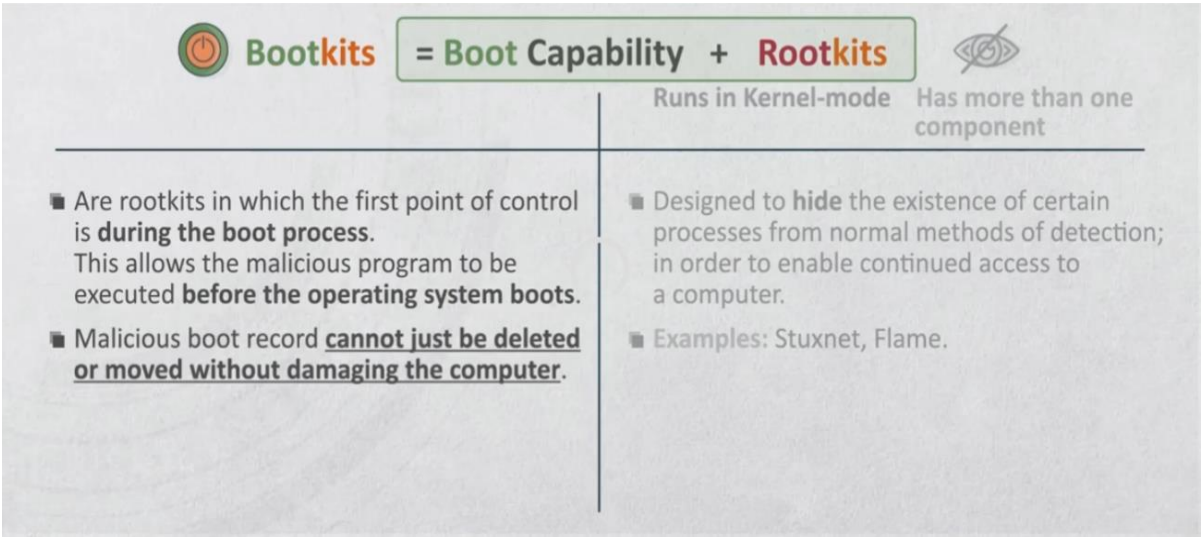


Monday, 10 October 2022
8:43 PM




Spyware
Data sent from infected device to attacker.
AD
Adware
(Advertise)
Data sent from attacker to infected device.


FileLess Malware
1- User recieves a spam message with a link to a malicious website.
2- User clicks on the link.
3- The malicious website loads Flash, which has known vulnerabilities, on the user`s computer.
4- Flash opens the windows PowerShell tool, which can execute instructions through the command line while operating in memory.
5- PowerShell downloads and executes a script from a command-and-control server.
6- The PowerShell script locates and sends the user`s data to the attacker.


Bootkits = Boot Capability + Rootkits
Runs in Kernel-mode
Has more than one component
Are rootkits in which the first point of control is during the boot process.
This allows the malicious program to be executed before the operating system boots.
Malicious boot record cannot just be deleted or moved without damaging the computer.
Designed to hide the existence of certain processes from normal methods of detection; in order to enable continued access to a computer.
Examples: Stuxnet, Flame.

CH01_VID04_PUPs

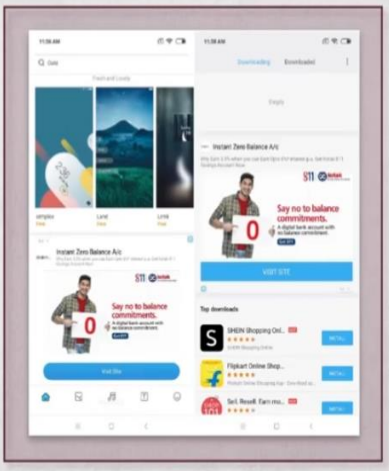
Watch later Share

PUPs (Potentially-Unwanted Programs) Examples

Toolbars for the browser
Source: Raymond.cc

Add-ons / Extensions for the browser
Source: Bleepingcomputer.com

Desktop programs
that show fake promotions or show the weather
Source: 2-spyware.com



Android apps
that spam you with notifications or ads
Or that show ads in the home screen or in the file manager
Source: androidcentral.com

Why They're Called PUPs and Not Malware ?

- Because mostly the **user agrees** to installing the PUP.

  (EULA)

- The **e**nd **u**ser **l**icense **a**greement informs the user that this additional program is being installed, and **the user accepts it**.

Steps of malware analysis

Examining a file without running it — Running the malware

A. Basic static analysis	B. Basic Dynamic analysis	C. Advanced static analysis	D. Advanced Dynamic analysis
Confirm whether a file is malicious or not. Provide info. about its functionality. Provide info. to produce **simple network signatures**.	**Running** the malware in safe and isolated lab environment. **Observing** its behavior on the system. **Remove** the infection. Produce effective **signatures**.	Loading the executable into a "**disassembler**", and looking at the program instructions which executed by the CPU.	**Using a debugger** to examine the internal state of the malware **while it is running**. with the ability to **modify** its execution path and state.
✓ Straightforward and quick. ✓ Can be used by most people **without deep programming knowledge**. ✗ It's not effective with all malware. ✗ It doesn't reveal the full extent of the malware's capabilities.		✓ It tells you **exactly what the program does**. ✗ It requires knowledge of disassembly, code constructs Windows OS concepts and internals. ✗ It takes much longer time than basic static analysis.	✓ Extract detailed information from themalicious executable.

Can malware **detect** it's running inside a VM? YES

How?

By detecting artifacts of the virtualization infrastructure:

- Hypervisor (VMM). → Operating system **uptime** is < 1 hour. / No mouse movement or user interaction.
- Virtualized hardware components. → Disk size is < 60 GB. / RAM is < 1 GB.
- Processes and/or Services.
- Files. → No files on the "Desktop". / No files in "My Documents" **or** "Downloads" folders.
- Registry.

Can malware **escape** the VM and infect the host? YES

When?

After elevating privileges on the guest.

Why?

To infect the **Host** → Analyst's machine.

→ Whole web server (Host of the VPS).

How?

- **Guest-To-Host exploits**

  0-day exploits, exploiting vulnerabilities in the hypervisor (VMM).

Examples | Guest-To-Host exploits

Vmware

- UHC (CVE-**2019**-5518).
- NIC (CVE-**2018**-6973).
- Drag n Drop (CVE-**2017**-4901).

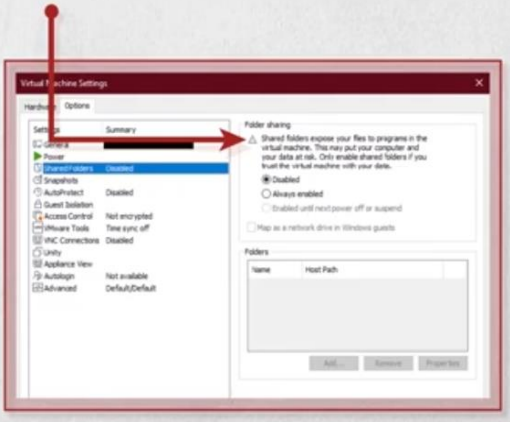
VirtualBox

- Core graphics framework (CVE-**2018**-2698).

unpatched exploit in oracle virtualbox guest to host escape

DON'T FORGET the shared folders

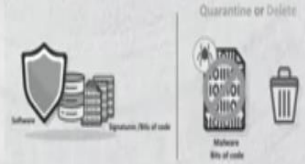
- Ransomware **will encrypt** any folders shared from the host to the guest.
- So be careful when Enabling the "Shared folders" feature.

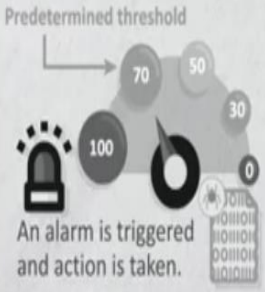


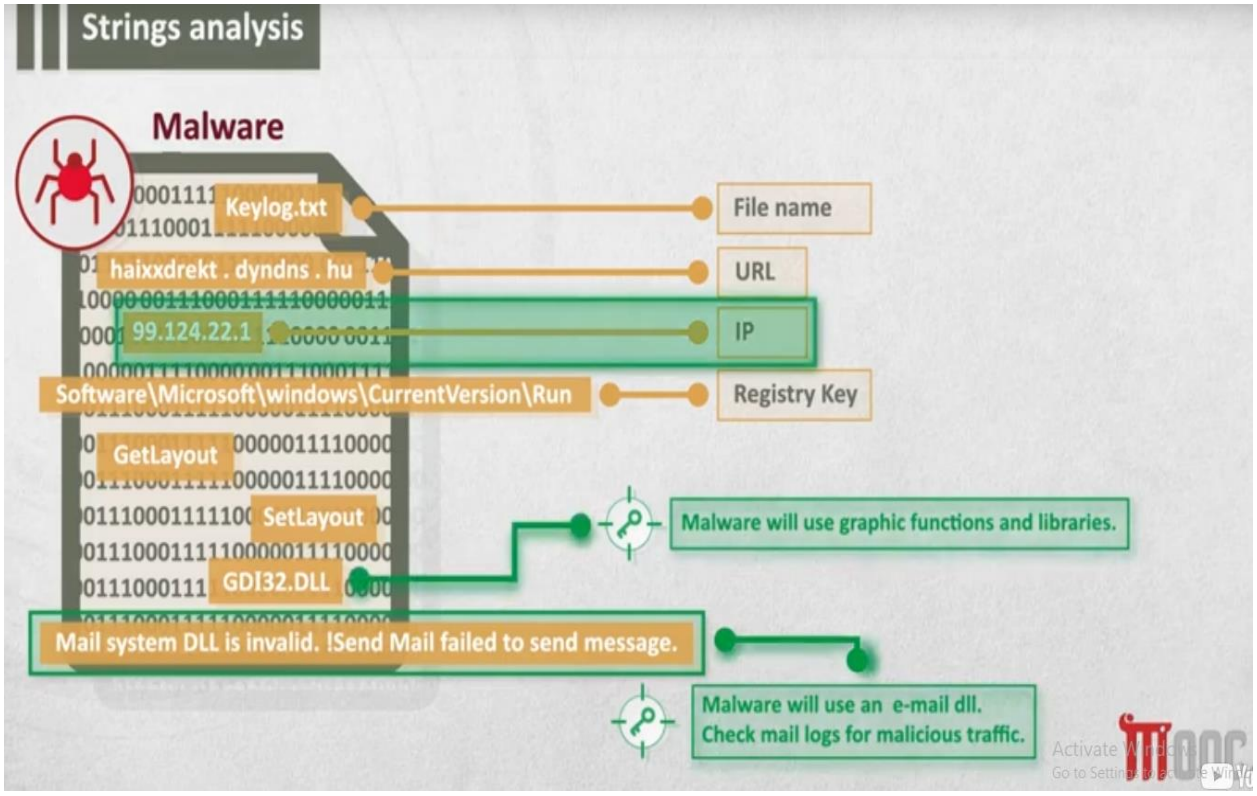
A. Basic static analysis

Antivirus Scanning

Signatures-based detection	Heuristics-based detection	Behavior-based detection	Machine learning
• It looks for the specific digital code of a virus. • **Disadvantage:** It does not detect new threats.	• It looks for specific instructions using a rule or weight-based system.	• It looks for software attempting to perform malicious functions. • **Disadvantage:** Identifying what the malware does **when** files are open.	• **Analyze** the code of applications and decide **based on its understanding** of malicious and benign programs.

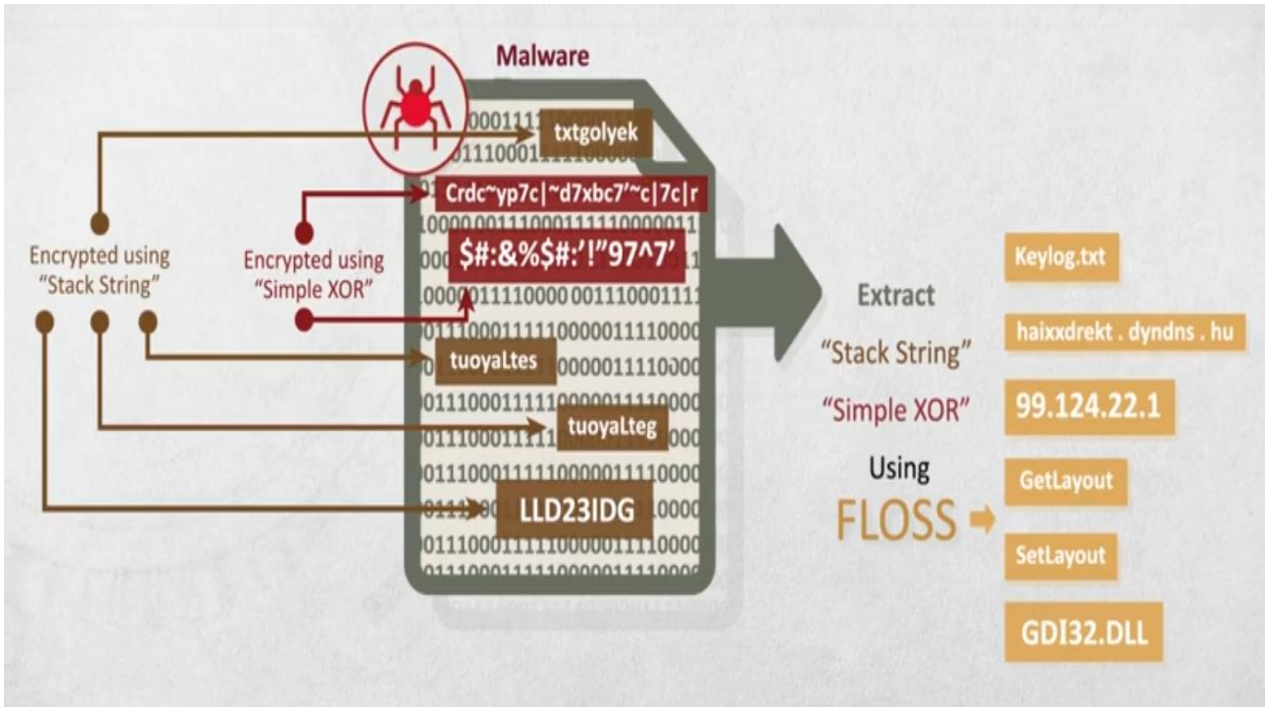


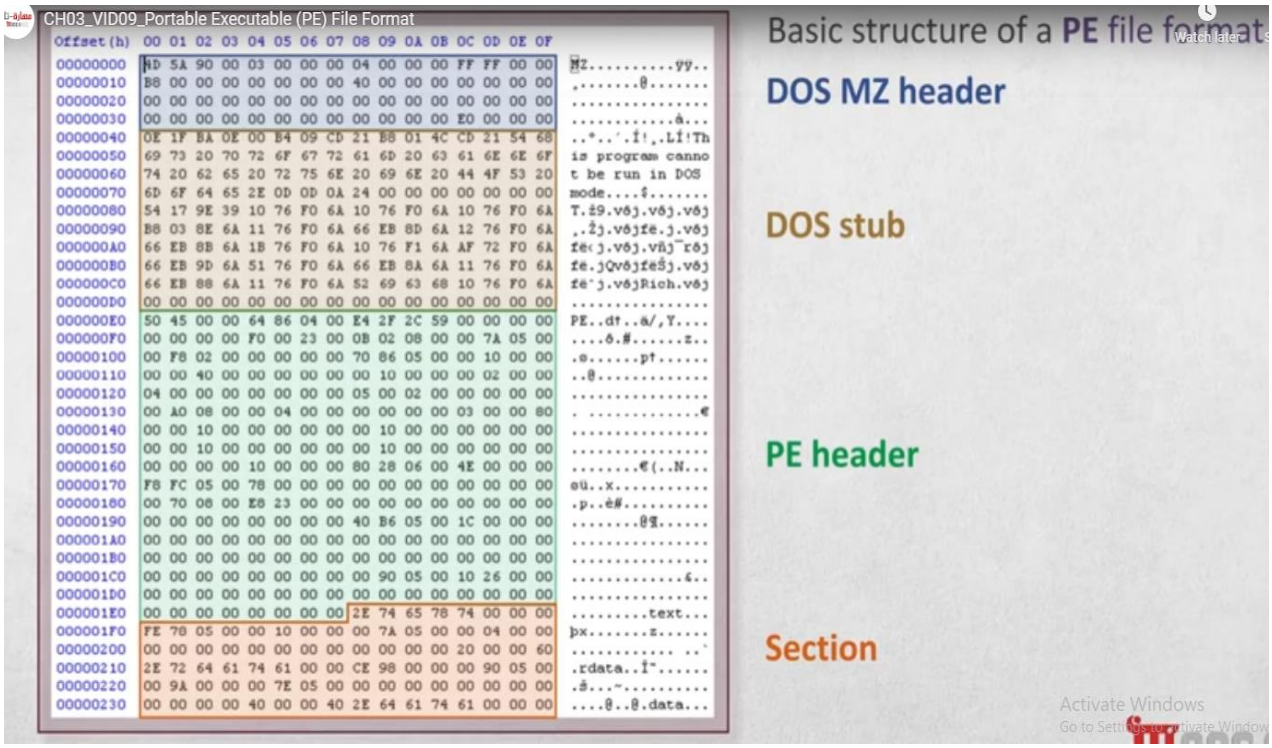




ASCII American Standard Code for Information Interchange.	UNICODE Universal character set.
Uses **7 bits** to represent a character.	Uses **8 bit, 16 bit, 32 bit** UTF-8 , UTF-16 , UTF-32
Requires **less** space.	Requires **more** space.

	Encoding	Bits
A	ASCII	01000010
A	UTF-8	01000010
A	UTF-16	00000000 01000010
A	UTF-32	00000000 00000000 00000000 01000010





Address of entry point in optional header is important and in most of reputable companies it is .text and that Is the address at which the code starts and other important header size of image it is the size of malware when it is unpacked and running

Microsoft

Common DLLs

Contains all user interface components (button, scroll bar, control and respond user actions).	User32 . dll
Contains display and manipulate graphics function.	Gdi32 . dll
Contains high and low level networking functions.	WSock32 . dll
	Ws2_32 . dll
	Wininet . dll
Provides access to advanced core windows components (service manager and registry).	Advapi32 . dll

Common imported functions by malware

- Naming Conventions.

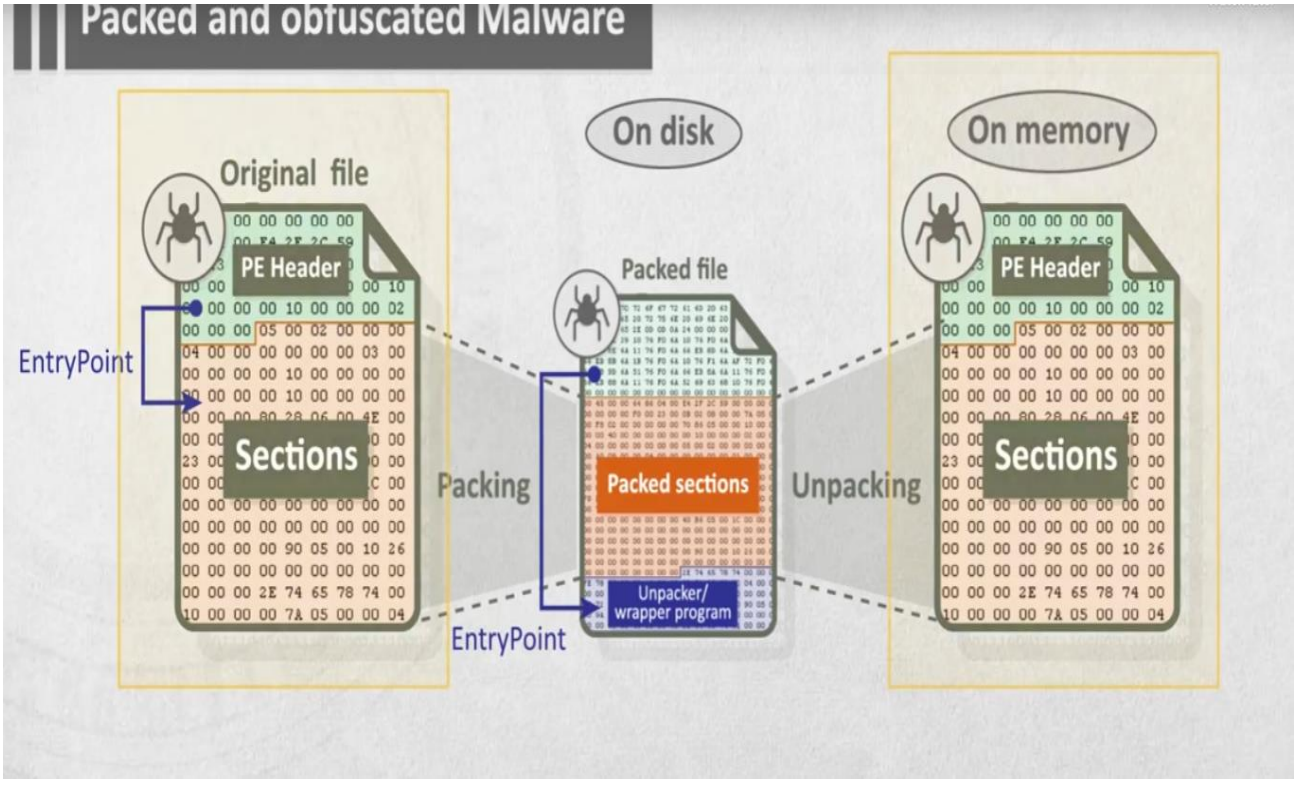
---------- A ASCII strings version.

---------- W Wide character strings version.

---------- Ex Updated function.

Kernel32.dll	User32.dll	User32.dll (continued)
CreateDirectoryW	BeginDeferWindowPos	**ShowWindow**
CreateFileW	CallNextHookEx	ToUnicodeEx
CreateThread	CreateDialogParamW	TrackPopupMenu
DeleteFileW	CreateWindowExW	TrackPopupMenuEx
ExitProcess	DefWindowProcW	TranslateMessage
FindClose	DialogBoxParamW	UnhookWindowsHookEx
FindFirstFileW	EndDialog	UnregisterClassW
FindNextFileW	GetMessageW	UnregisterHotKey
GetCommandLineW	GetSystemMetrics	
GetCurrentProcess	GetWindowLongW	**GDI32.dll**
GetCurrentThread	GetWindowRect	GetStockObject
GetFileSize	GetWindowTextW	SetBkMode
GetModuleHandleW	InvalidateRect	SetTextColor
GetProcessHeap	IsDlgButtonChecked	
GetShortPathNameW	IsWindowEnabled	**Shell32.dll**
HeapAlloc	LoadCursorW	CommandLineToArgvW
HeapFree	LoadIconW	SHChangeNotify
IsDebuggerPresent	LoadMenuW	SHGetFolderPathW
MapViewOfFile	MapVirtualKeyW	ShellExecuteExW
OpenProcess	MapWindowPoints	ShellExecuteW
ReadFile	MessageBoxW	
SetFilePointer	RegisterClassExW	**Advapi32.dll**
WriteFile	**RegisterHotKey**	RegCloseKey
	SendMessageA	RegDeleteValueW
	SetClipboardData	RegOpenCurrentUser
	SetDlgItemTextW	RegOpenKeyExW
	SetWindowTextW	RegQueryValueExW
	SetWindowsHookExW	RegSetValueExW


Sections
Contains the executable code.
Stores global data accessed throughout the program.

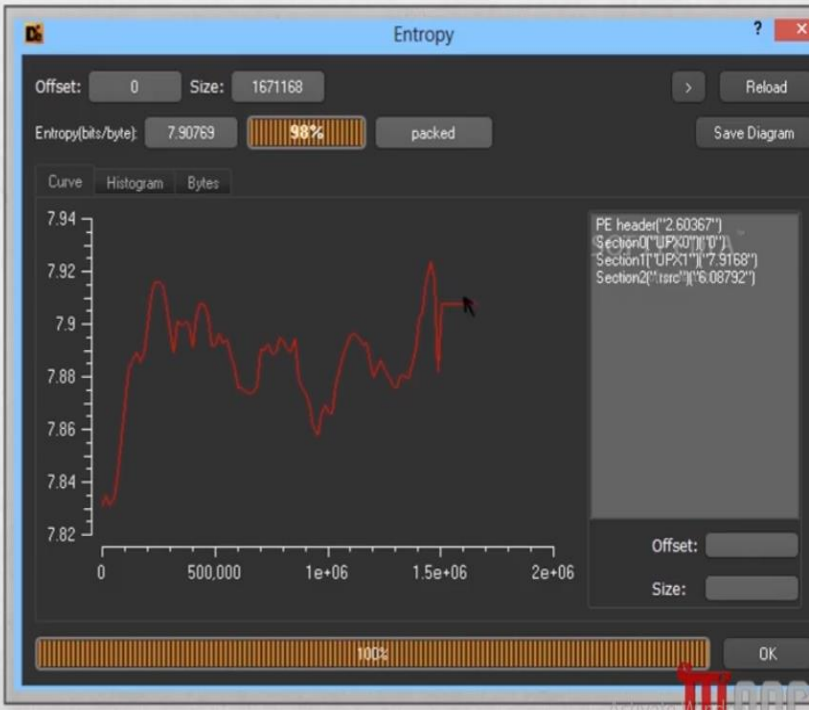

Packed and obfuscated Malware
On disk
On memory
Original file
PE Header
EntryPoint
Sections
Packing
Packed file
Packed sections
Unpacker/
wrapper program
EntryPoint
Unpacking
PE Header
Sections

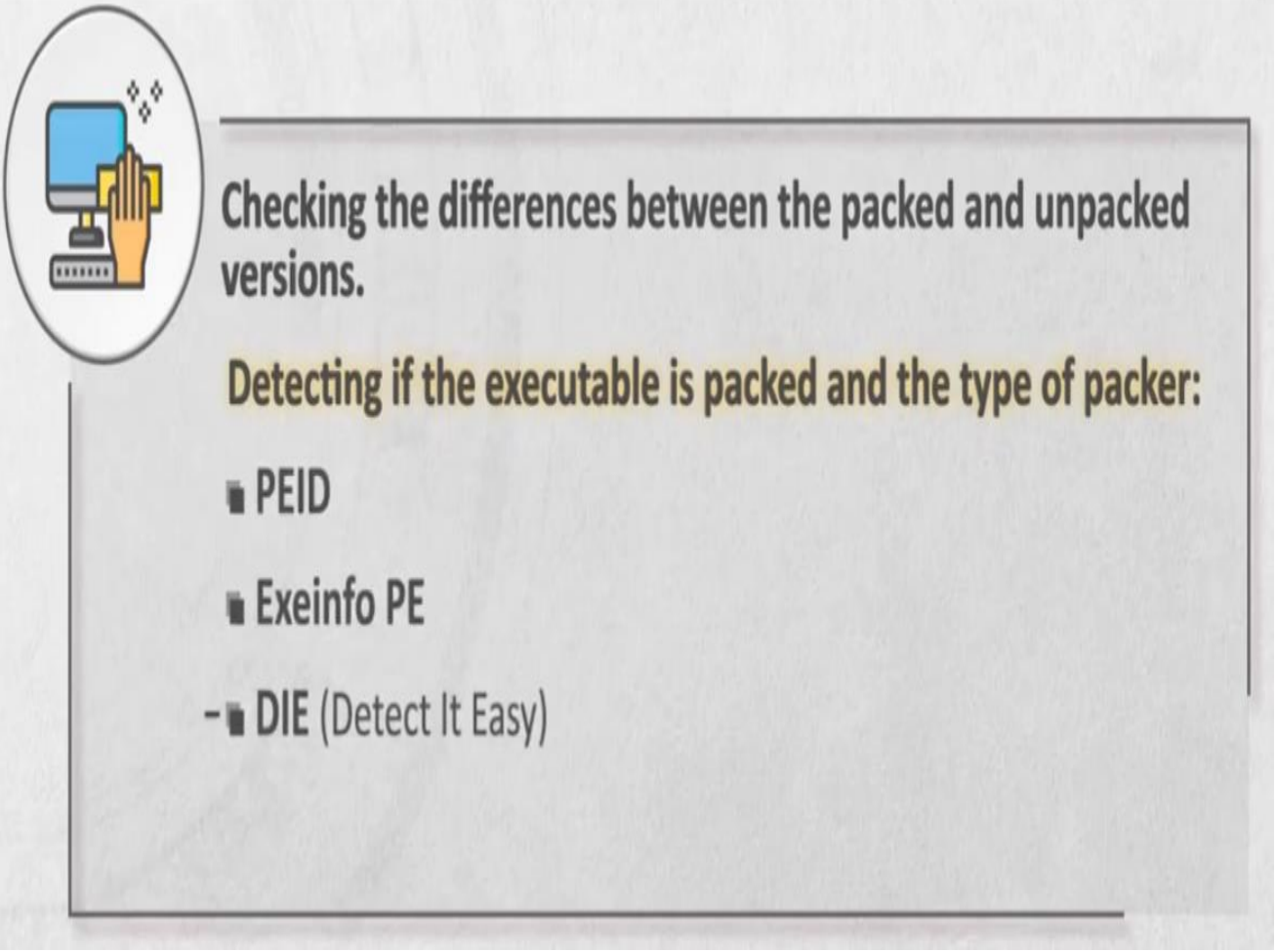
PEID

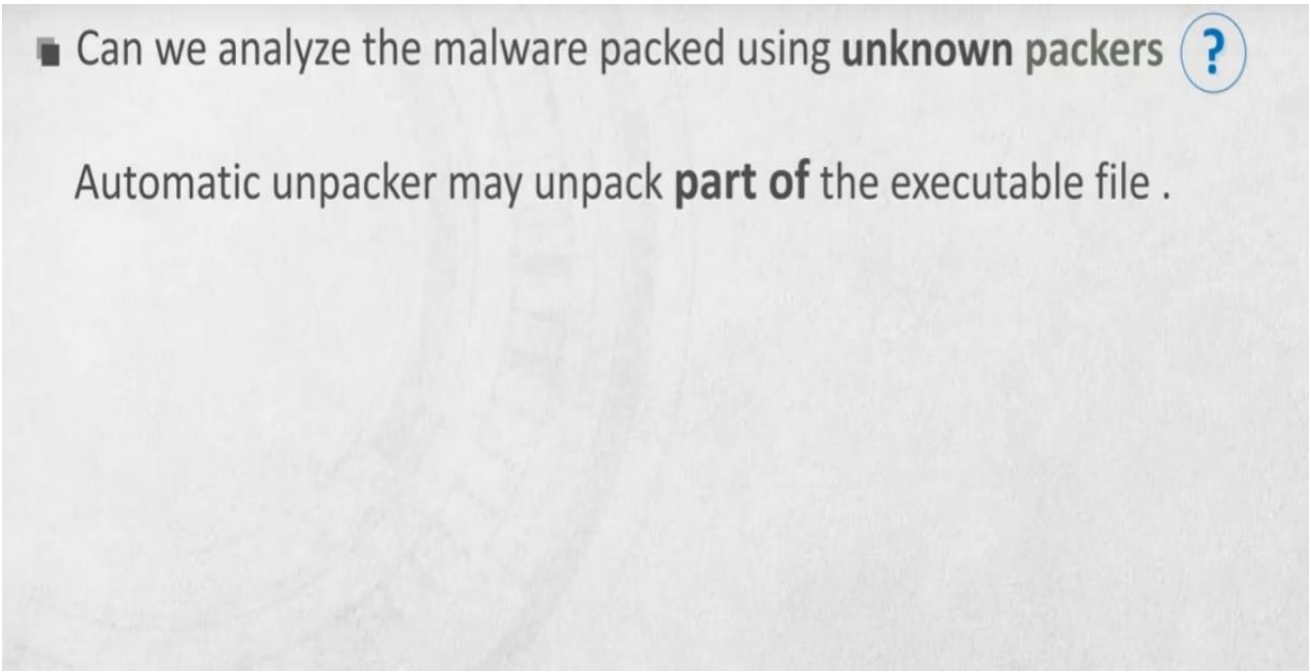
- **"Normal Mode"** scans the PE files at their Entry Point for all documented signatures. This is what all other identifiers also do
- **"Deep Mode"** scans the PE file's Entry Point containing section for all the documented signatures. This ensures detection of around 80% of modified and scrambled files.
- **"Hardcore Mode"** does a complete scan of the entire PE file for the documented signatures. You should use this mode as a last option as the small signatures often tend to occur a lot in many files and so erroneous outputs may result.

Packed and obfuscated Malware | Tools to identify packers

- PEID
- Exeinfo PE
- DIE (Detect It Easy)
  - Entropy







ENG : Mohamed Ewies
/https://www.linkedin.com/in/mohamed-ewies-59b89a212