Information Security
CS 3002

Dr. Haroon Mahmood

Assistant Professor

NUCES Lahore

Preface

- What malware are?
- How do they infect hosts?
- How do they propagate?
- How to detect them?
- How to prevent them?
- Malware Analysis
- Conclusion

What is a malware ?

- **“A program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victim's data, applications, or operating system (OS) or of otherwise annoying or disrupting the victim.” (NIST – 2005)**
- **Also called digital pests**

What can it do?

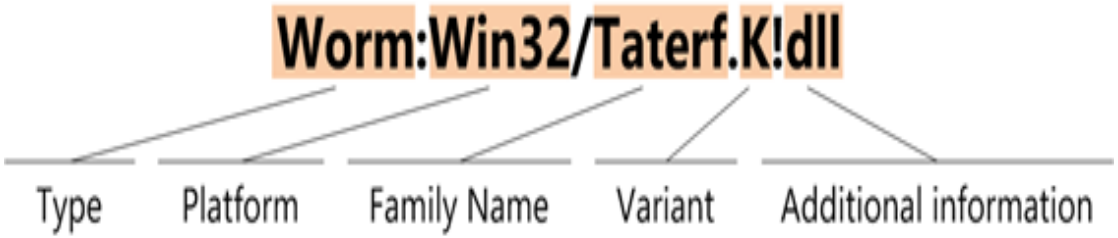
- Steal personal information
- Delete files
- Make you click fraud
- Steal software serial numbers
- Use your computer as relay (Zombie)
- Corrupt files
- etc...

Types of Malware

- **Virus:** ***attaches itself to a program***
- **Worm:** ***propagates copies of itself to other computers***
- **Logic bomb:** ***"explodes" when a condition occurs***
- **Trojan horse**: ***fakes/contains additional functionality***
- **Backdoor (trapdoor):** ***allows unauthorized access to functionality***
- **Spyware:** **used to spy on victim's activities on a system and also for stealing sensitive information of the client.**
- **Ransom-ware**: **steals some functionality and returns after a ransom is paid**
- **Scare-ware:** **users are tricked by scaring and motivated to perform some action. E.g. buying a software license**
- **Key-loggers:** ***capture keystrokes***
- **Browser hijacker:** **modifies a web browser's settings without a user's permission, to inject unwanted advertising into the user's browser**
- **Zombie:** ***software on infected computers that launch attack on others***

Malware naming

- **CARO (computer antivirus researchers organization)**
- **CARO naming convention (1991)**



History

- **1982 First reported virus : Elk Cloner (Apple 2)**
- **1983 Virus gets defined**
- **1986 First PC virus MS DOS (**Brain written by two Pakistani brothers**)**
- **1988 First worm : Morris worm**
- **1998 Back orifice: remote management tool**
- **1999 Melissa virus: macro virus**
- **1999 Zombie concept**
- **2000 love bug: vbs worm: damage: $15B**
- **2001 Nimda worm**
- **2003 SQL Slammer worm: damage $1.2B (**Vulnerability: buffer overflow**)**
- **2001 Code Red: DoS worm, damage: $2.6B**
- **2004 MYDOOM Ddos worm, damage: $38B**

What is a Virus ?

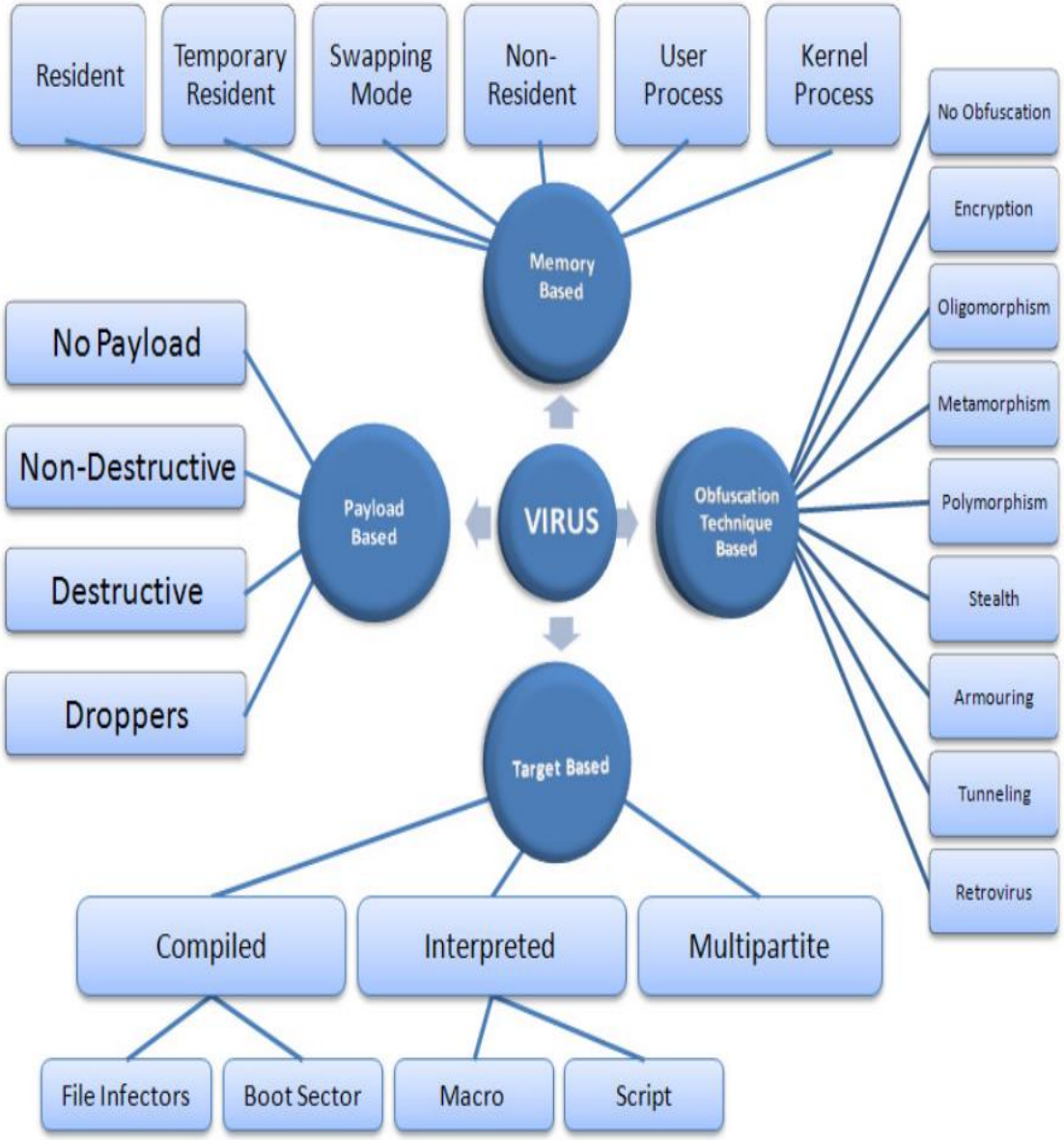
- ***A program that can infect other programs by modifying them to include a, possibly evolved, version of itself (*Fred Cohen 1983*)***

- **It executes secretly when host program is run**
- **Inserts copies of itself into host programs/data files**
- **Requires user interaction**
- **Often specific to operating system and hardware**
  - **taking advantage of their details and weaknesses**

Classification of Virus

- **Classification of viruses can be done as follows:**
  - **Memory Based**
    - **How they live (stay) in memory**
  - **Target Based**
    - **How they spread to others**
  - **Obfuscation Technique Based**
    - **What they do to hide**
  - **Payload Based**
    - **What they do after infection**

Virus classification



Memory based Classification

- **This categorization is based on their behavior while they operate in memory.**

  1. **Resident**
  2. **Temporary Resident**
  3. **Swapping Mode**
  4. **Non-Resident**
  5. **User Process**
  6. **Kernel Process**

Cont.

Resident Virus:

- The virus code is loaded into memory and is copied to all the relevant host files that are running in the memory.
- For example: A TSR [Terminate and Stay Resident] program that stays in the allocated memory even after the termination of the main program.

Temporary Resident Virus:

- Stays in memory temporarily and remove itself out of memory when a certain event occurs
- Extremely difficult to detect

Cont.

Swapping Memory Virus:

- **Such kind of viruses load only a part of their code into memory on occurrence of a certain event, infect the files present in memory and unload the code from memory**

- **These viruses may be spotted by the increase in disk activity due to loading and unloading of viral code and infection of other host files.**

Cont.

Non-Resident Virus:

- Such viruses do not reside in physical memory.
- They have an offline mechanism to search for and infect files present in the hard disk.

- They contain two key sub-routines.
  - Finder or search sub-routine that searches the hard disk for the relevant files to infect
  - Copy sub-routine that copies the virus code into the files found
- If writable network shares are present, these can spread to other systems using them. These are also called *'Directaction viruses'.*

Cont.

User Process:

- These viruses run as a user process and infect the files that are accessible.
- Although the virus can exist as its own process. Most of the time, they exist as a sub-process loading before or after the main process.
- In some of the cases, the virus exist as a DLL and uses DLL Injection method (through registry keys) to load the DLL into the process.
- Autorun.abt is an example of this type of virus.

Cont.

Kernal Process:

- These types of viruses generally hook themselves into the kernel through a system driver like program.
- They have the highest privileges after infection as they are present in the kernel space.
- These generally infect/modify the IDT [Interrupt Descriptor Table] to get themselves executed every time a particular interrupt is generated.
- As these viruses require changes to the main file system, they need administrator/super user privileges to run.
- CIH, Infis are examples of this type of virus.

Obfuscation Techniques

- **Techniques that are being used to avoid detection and analysis!**

1. **No Obfuscation**
2. **Encryption**
3. **Oligomorphism**
4. **Polymorphism**
5. **Metamorphism**
6. **Stealth**
7. **Armoring**
8. **Tunneling**
9. **Retro**

Cont…

No Obfuscation

- easier to build
- detection and analysis of such a virus is trivial

Encryption

- use of cryptography to hide the functionality
- a de-crypter along with the encrypted body that decrypts the virus on-the-fly
- The decryption key can:
  - exist in the virus body along with the decryption algorithm
  - Be recovered with a simple brute force by the virus itself

Cont…

Oligomorphism

- also called '*Semi-polymorphic*'
- use of multiple decryption routines to avoid giving a signature for the antivirus software.
- The decryption routine is chosen randomly on infection.

Polymorphism

- change the look of the virus code every time it infects a new file by changing the decryption routine
- high number of decryption routines using a 'mutation engine, which does all the logic in creating a new decryption routine

Cont…

Metamorphism

- change the virus body instead of appearance by using equivalent and unneeded functions (or code) or by changing the sequence of statements in the code slightly (as long as the logic remains relevant)
- every specimen looks different and generation of a signature is harder

Stealth

- Tries to remain undiscovered by hiding the infection events from everyone, instead of trying to obfuscate its code
- Achieves this by restoring certain original properties
  - Timestamp
  - File size

Cont…

Armoring

- use various anti-debugging, anti-heuristics and anti-VM (virtualmachine detection) techniques
- Use of file packers, copying itself to multiple sections in the host file

Tunneling

- Use of Operating system interrupts
- virus executes first and after that the control is passed to the original destination

Cont…

Retro Virus

- tries to bypass or hinder the operation of an antivirus, personal firewall, or other security programs.
- also called '*Anti-antivirus viruses*'
- They generally have a database of identification mechanisms for different security controls like process names, registry keys. Once identified, the security controls can be terminated or corrupted
- block users from updating their antivirus software or opening of system utilities or antivirus vendor websites

Phases of virus

- **4 phases:**
  - **Dormant phase: It is idle, waiting for some event**
  - **Triggering phase: activated to perform some intended actions**
  - **Propagation phase: Copy itself into other programs**
  - **Execution phase: execute the payload**

Lifecycle of virus

- A virus gets created and released
- The virus infects several machines
- Samples are sent to anti-virus companies
- Records a signature from the virus
- The companies include the new signature in their database
- Their scanner now can detect the virus

What is a trojan

A Trojan horse is a non-replicating program that, while appearing to be benign, actually has a hidden malicious purpose. (NIST – 2013)

- **named after the wooden horse the Greeks used to infiltrate Troy**
- **harmful software that looks legitimate**
- **Can replace existing files or add new malicious files to hosts**
- **Can launch multiple attacks (irritating user, damaging files, spreading other malware etc..)**

Examples of Trojan

- **Backdoor Trojan**
  - It can create a “backdoor”
  - lets an attacker access your computer and control it
  - data can be downloaded by a third party and stolen
  - more malware can be uploaded to your device.
- **DDoS attack Trojan**
- **Downloader Trojan**
  - It targets your already-infected computer
  - downloads and installs new versions of malicious programs that can include Trojans and adware

What is a worm

A computer worm is a self-replicating computer program. It uses a network to send copies of itself to other nodes and do so without any user intervention.

- **using email, remote exec, remote login**
- **First replicates then does damage**
- **Exploits a vulnerability in existing software**
- **It has phases like a virus:**
  dormant, propagation, triggering, execution

propagation phase: searches for other systems, connects to it, copies itself and executes

Worm

- Can cause enormous damage
  - Launch DDOS attacks, install bot networks using zombies/bots
  - Access sensitive information
  - Cause confusion by corrupting the sensitive information
  - May disguise itself as a system process

SQL Slammer

- exploited buffer-overflow vulnerability
- Vulnerability disclosed : 25th of June 2002
- Better scanning algorithm

- Consequences
  - ATM systems not available
  - Phone network overloaded (no 911!)
  - 5 DNS root down
  - Planes delayed

Malware Properties

Malware	Host Required	Replication Mechanism
Virus	Yes	Self
Worm	No	Self
Logic Bomb	No	Manual
Backdoor	No	Manual
Trojan	Yes	Manual
Spyware	No	Manual
Rootkit	No	Manual
Bots	No	Manual

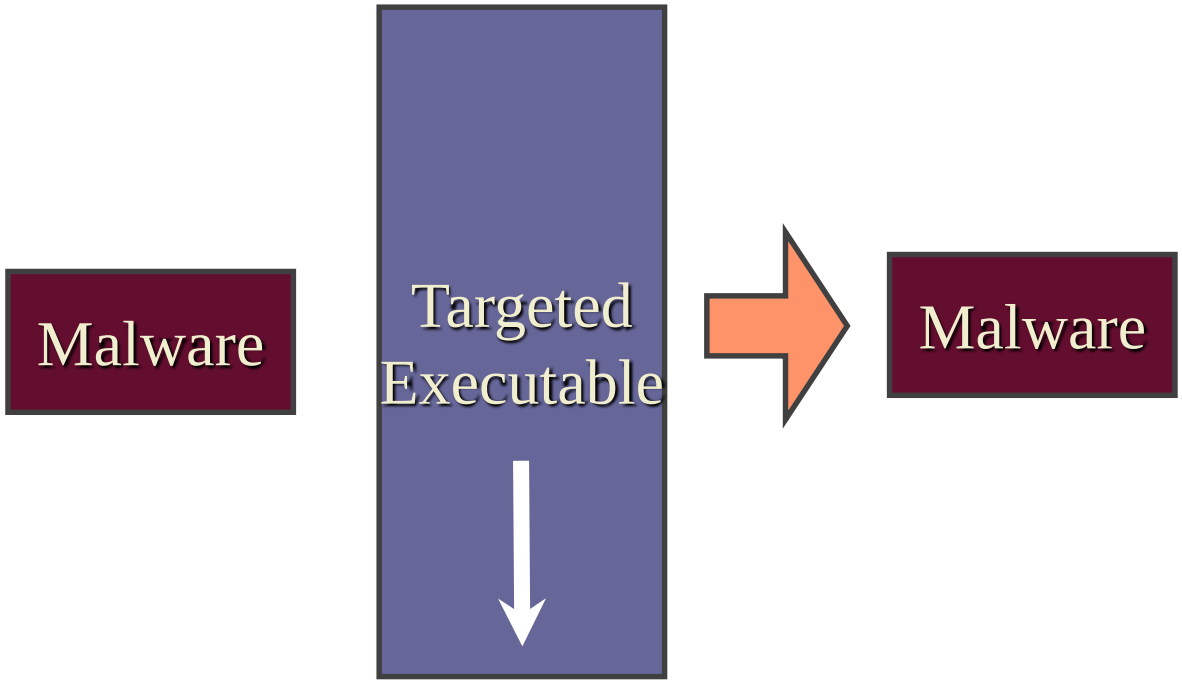
- **Host required: malware needs user interaction**

What they Infect?

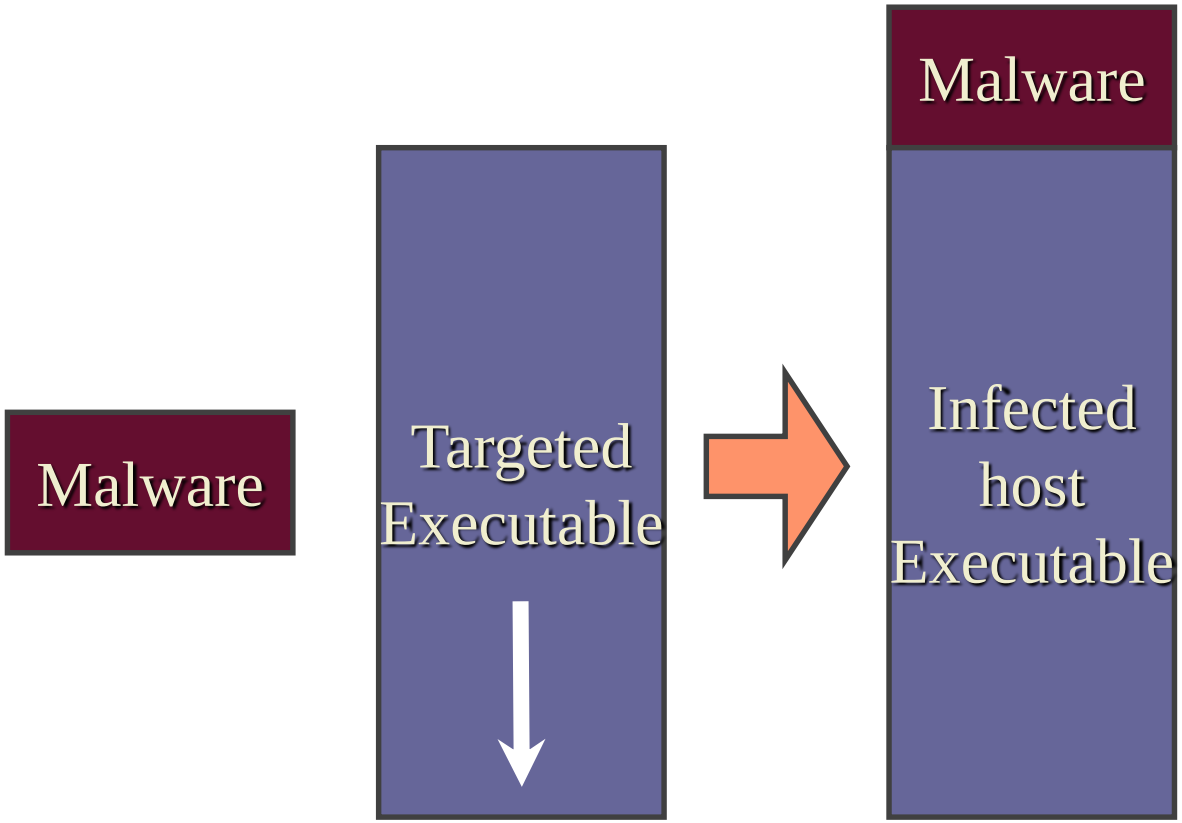
- Executable
- Interpreted file
- Kernel
- Service
- MBR (Master Boot Record)
- Hypervisor

Overwriting malware

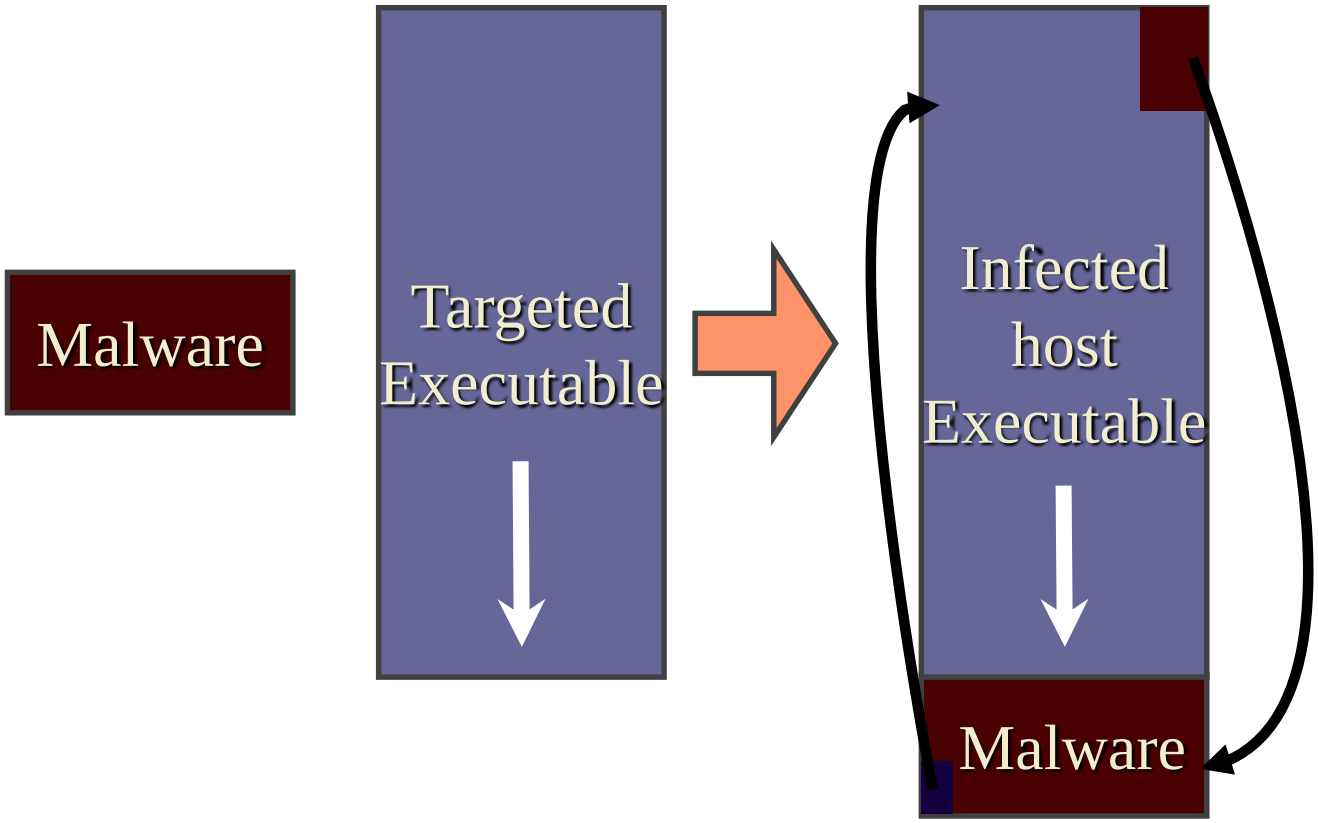
- After infection, it will effectively destroy the original program code by overwriting data in the system's memory.
- TRj.reboot virus: It can restart the user's computer, and was active in targeting Windows NT and Windows 2000 systems in the 2000s. Trivial.88.D virus: A 'direct action virus' that infects executable files.



Pre-pending malware

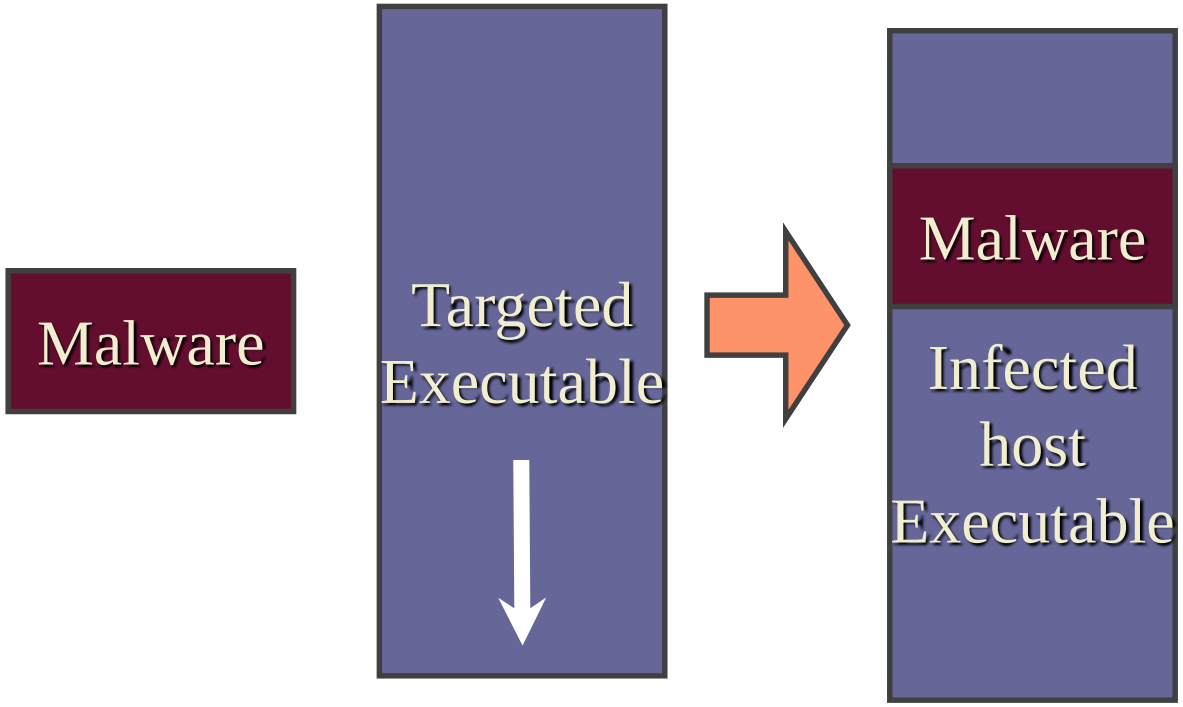


Appending malware

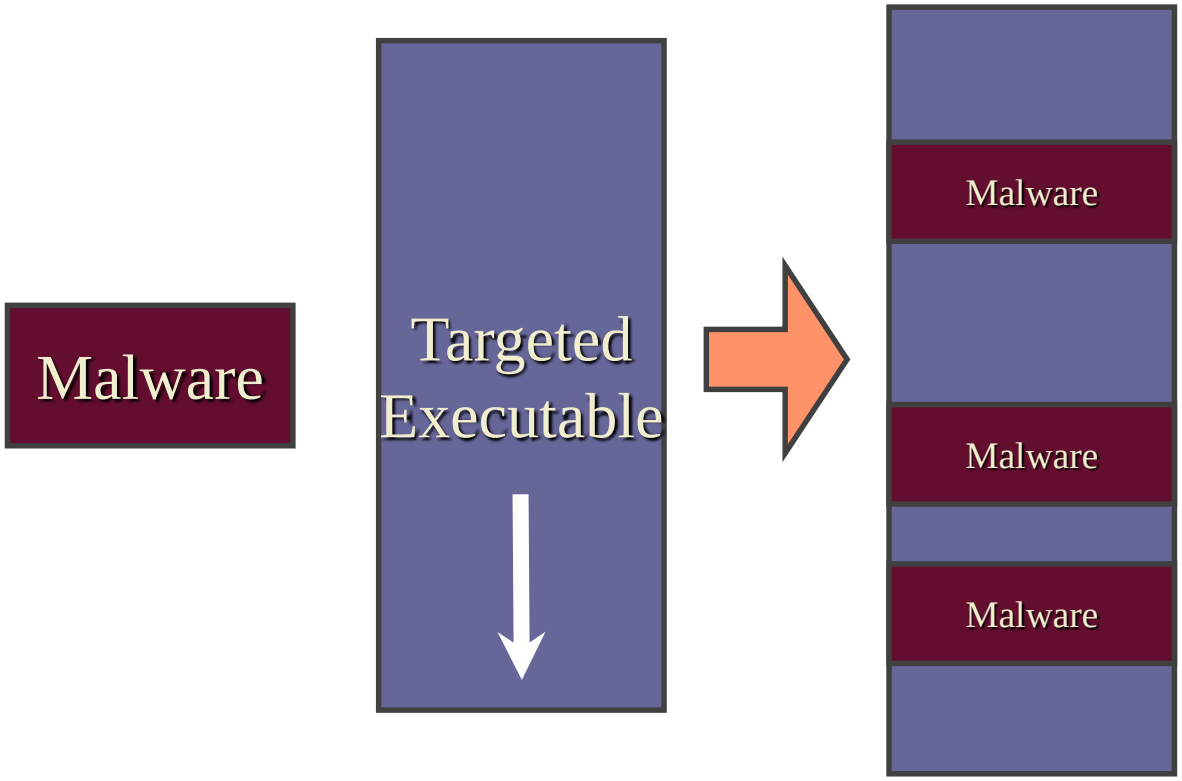


Cavity malware

- **Some viruses can infect files without increasing their sizes or damaging the files by overwriting unused areas of executable files. These are called cavity viruses.**
- **For example, the CIH virus, or Chernobyl Virus which infects Portable Executable files.**

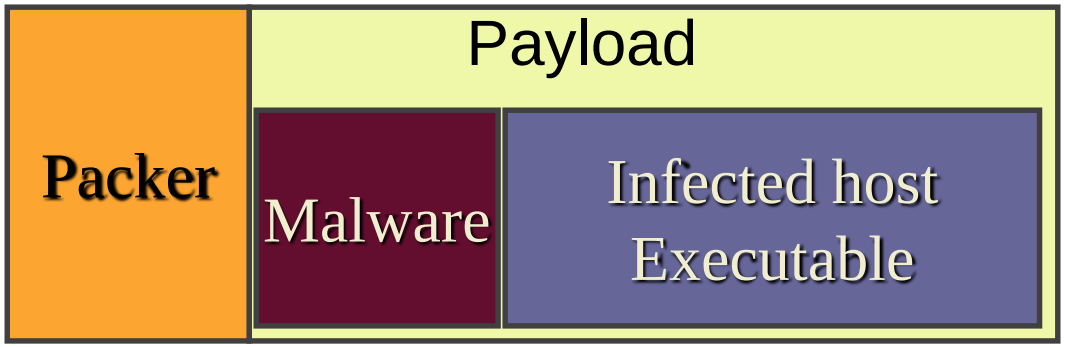


Multi-Cavity malware



Packers

A **packer** compresses or encrypts data. The original file is passed in the **packer** routine and stored in a packed section in the new .exe.



Malware Analysis

- **Malware analysis is the study or process of determining the functionality, origin and potential impact of a given malware. (wikipedia)**

- **Three typical use cases**
  - **Computer Security incident management**
  - **Malware research**
  - **Indictor of compromise extraction**

- **Types**
  - **Static**
  - **Dynamic**

Why analyze Malware?

- To assess damage
- To discover indicators of compromise
- To determine sophistication level of an intruder
- To identify a vulnerability
- To catch the “bad guy”
- To answer business related questions
  - How long has it been here, spreads on its own? Etc.
- To answer technical questions
  - Date of installation, compilation, persistence mechanism, network or host based indicators

Static Analysis

- Analysis in which code is not executed
- “Dead” code is read and understood
- Also referred to as: code analysis
- Requires peeking into the code using a hex editor, unpacking and performing string searches.
- Disassembling the malware. Disassemblers take machine code to higher-level code
  - IDA Pro
- Static analysis is safer
- Malware files are fingerprinted before analysis. Just in case malware analysis is being expected by the (malware) developer.
- Virus scan
  - PEiD, Caprica6 tool can tell you about “packed” code

Dynamic Analysis

- Conducted by observing and manipulating malware as it runs
- Needs a safe environment to analyze (run) the code
  - Sandboxed environment.
- Requires monitoring the system
  - Registry files activity
  - File and process/system level activity
  - Network level activity
- Some tools
  - Wireshark
  - SysInternals process monitor
  - Netstat or ResMon in Windows can be used
- Requires analysis while the code is being run using tools like WinDbg

Static vs Dynamic Analysis

- **Static:** Dissecting code via different resources without executing
- **Dynamic:** Behavioral analysis is performed by executing the malware.
- Static is much slower (and exhaustive at times) as compared to dynamic.
- Static is far safer than dynamic.
- Static doesn't (necessarily) need a sandboxed environment while dynamic does.

Six Steps to incident handling process

- **Preparation:** Get our team ready. Jump bags, warning banners, response strategies
- **Identification:** Identify if an event is an incident. Done at network perimeter level or host/system level.
- **Containment**: limit the propagation/spreading of malware incident.
- **Eradication:** Removal of infection from the system.
- **Recovery:** Restoration of services/functionalities
- **Lessons learned:** Be prepared for next time. Study the reason why an incident occurred and take care of it so it wont get repeated.

Malware defenses (1)

- **Detection:** once the infection has occurred, determine that it has occurred and locate the malware
- **Identification:** once detection has been achieved, identify the specific virus that has infected a program
- **Removal:** once the specific malware has been identified, remove the malware from the infected program and restore it to its original state

Malware defenses (2)

- The first generation scanner
  - Malware signature (bit pattern)
  - Maintains a record of the length of programs
- The second generation scanner
  - Looks for fragments of code (neglect unnecessary code)
  - Checksum of files (integrity checking)
- The third generation scanner
  - Identify a malware by its actions
- The fourth generation scanner
  - Include a variety of anti-malware techniques

Malware defenses (3)

- **Malware-specific detection algorithm**
  - **Deciphering**
  - **Filtering**
- **Collection method**
  - **Using honeypots**
- **Analyze program behavior**
  - **Network access**
  - **File open**
  - **Attempt to delete file**
  - **Attempt to modify the boot sector**

How to prevent them

- Simple! Learn about security (Not so simple)
- Use a secure Operating systems
- Use secure browsers and plugins/extensions
- And update/patch regularly
- Install anti-virus (maybe?)
- Avoid torrents
- Surf secure websites
- Don’t download what you don’t understand/need
- Use Instant Messaging apps carefully
- Keep backups

How to prevent them

- Don’t install software that you don’t need or remove after one time use(worms!).
- Install software carefully. Unnecessary bundles gets installed
- Open email attachments with caution
- Monitor the performance of your pc regularly
- Keep frequent restore points and restore your pc if you think you executed a virus/worm/trojan
- Avoid unlicensed software installation
- Layers of authorization for installation of new tools/software

How to prevent them

Two layers:

- **Personal vigilance (First layer)**
  - Knowing what to do and what to install
  - Understanding of the system and security
  - Strong passwords (password checkers)
- **Protective tools (Second layer)**
  - Effective and enough prevention tools
  - They are never enough