Dr. Ammar Haider
Assistant Professor
School of Computing

CS3002 Information Security

Malware Study and Prevention

Reference: Stallings SPP chap 6

Outline

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

What is 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 otherwise annoying or disrupting the victim.” (NIST – 2005)
- Also called malicious software or digital pests

What it can do?

- Steal personal files (documents, photos etc.)
- Delete files
- Corrupt files
- Cause annoyance
- Steal information (software serial numbers, passwords etc.)
- Use your computer as relay (zombie)
- etc.

Some Malware Terminology

How malware infects a system

- Virus: attaches itself to a host file, replicates itself when executed
- Worm: propagates copies of itself to other computers in network, without any user action
- Trojan horse: apparently useful program, contains additional malicious functionality
  - Scareware: users are tricked by scaring and motivated to perform some action. e.g. buying a software license
- Browser hijacker: modifies a web browser's settings without a user's permission, to inject unwanted advertising into the user's browser

Some Malware Terminology

What actions a malware takes after infection

- Logic bomb: "explodes" when a condition occurs
- Backdoor (trapdoor): allows unauthorized access to functionality or a system, bypassing normal checks
- Spyware: used to spy on victim's activities on a system and also for stealing sensitive information
- Ransomware: blocks access to data or functionality and asks for a ransom payment
- Keylogger: capture keystrokes
- Zombie (bot): software on infected computers that can be used to launch attacks on *other machines*

Some Malware Terminology

What actions a malware takes after infection

- Rootkit: Set of hacker tools that allow attacker to maintain root-level (admin) access to a system, while hiding their presence
- Bootkit: infects the master boot record and spreads when a system is booted from the disk containing the malware.

Example Scenarios

- A user comes across a free media player with a nice UI. When they download and install the app, the player works fine but their computer starts behaving strangely, and they notice that their personal files are missing.

  trojan

- A company's network experiences a sudden surge in traffic, and employees report receiving numerous spam emails. The IT department investigates and discovers that a malicious program is rapidly spreading through the network, replicating itself and sending copies to other devices.

  worm

- A disgruntled employee embeds a hidden code in the company's payroll software. Months later, when the employee is fired, the code activates, deleting all employee records and causing significant disruption to the company's operations.

  logic bomb

Example Scenarios

- A user’s files become inaccessible, and they see a message demanding a payment in exchange for the decryption of their locked files.

ransomware

- A hacker gains access to a company's servers and installs a hidden program that allows them to monitor the network's activity and steal sensitive data without being detected.

rootkit, backdoor

- A company's network is infected with a malicious program that modifies the boot sector of infected computers. When these computers are restarted, the program loads before the operating system, giving the hacker control over the machine.

bootkit

Example Scenarios

- A user receives a pop-up message on their computer, warning them that their system is infected with a dangerous virus and their personal data is at risk. The message urges the user to download a “free antivirus scanner” to remove the virus and protect their computer. If the user clicks on the download link, they are directed to a website that attempts to install malicious software on their computer.

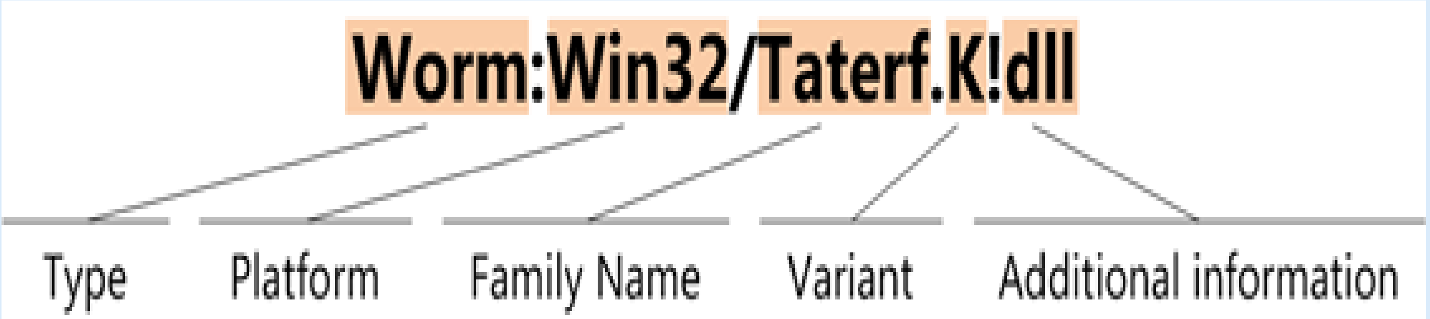
scareware

- A user downloads a free PDF reader from an untrusted website. Unbeknownst to the user, the PDF reader contains malware that secretly tracks their online activity, collects their photos, and captures screenshots. The malware sends this collected data to a remote server, allowing the attackers to monitor the user's browsing habits, steal sensitive information, and potentially commit identity theft.

trojan,
spyware

Naming Scheme

- Computer Antivirus Researchers Organization (CARO) created a naming convention in 1991



- It is not a universal standard, but most anti-virus vendors have adopted similar schemes

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

History

- 1999 Zombie concept
- 2000 love bug: vbs worm. damage: $15B
- 2001 Nimda worm
- 2003 SQL Slammer worm. damage $1.2B (buffer overflow vulnerability)
- 2001 Code Red: DoS worm, damage: $2.6B
- 2004 MyDooM DDoS worm, damage: $38B

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 for initial trigger
- Often specific to operating system and hardware
  - taking advantage of their details and weaknesses

4 phases of Virus

- 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

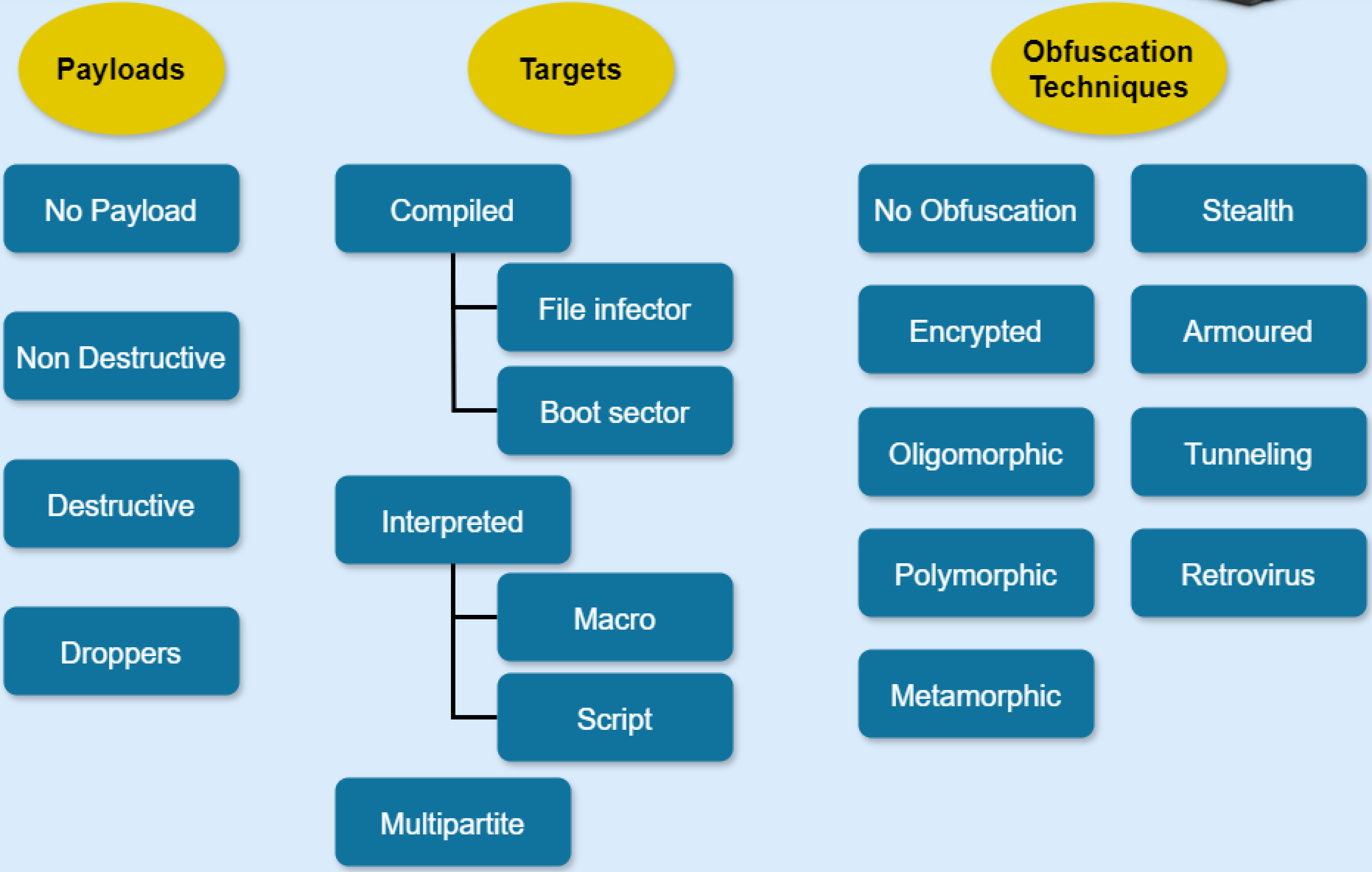
Lifecycle of virus

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

Classification of Virus

- 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

Classification of Viruses



- Payloads
  - No Payload
  - Non Destructive
  - Destructive
  - Droppers
- Targets
  - Compiled
    - File infector
    - Boot sector
  - Interpreted
    - Macro
    - Script
  - Multipartite
- Obfuscation Techniques
  - No Obfuscation
  - Encrypted
  - Oligomorphic
  - Polymorphic
  - Metamorphic
  - Stealth
  - Armoured
  - Tunneling
  - Retrovirus

Dropper Payloads

- A virus that is designed to drop more malware on the computer
- Does not do malicious activities by itself, but download/decompress more malware and install it
- It acts as a container or carrier that encapsulates additional malware components.

Obfuscation Techniques

Techniques that are being used to avoid detection and analysis!

1. No Obfuscation

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

Obfuscation Techniques

obfuscating the virus code

2. Encryption

- use of cryptography to hide the functionality
- a decryptor along with the encrypted body that decrypts the virus on-the-fly
- A new encryption key is chosen when infecting a new file
- 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

Obfuscation Techniques

obfuscating the virus code

3. Oligomorphism

- also called ‘Semi-polymorphic’
- use of multiple decryption routines to avoid giving a signature for the antivirus software.
- There are some pre-defined decryption routines, one of them chosen randomly on infection.
  - So can still be detected via signatures

4. Polymorphism

- change the look of the virus code every time it infects a new file by dynamically changing the decryption routine
- Can create very large number of decryption routines using a ‘mutation engine’, which does all the logic in creating a new decryption routine

Obfuscation Techniques

obfuscating the virus code

5. Metamorphism

- change the virus body for each infection
  - without using encryption
- using equivalent and unneeded functions (or code) or by changing the sequence of statements in the code slightly (as long as the logic remains same)
- every specimen looks different and generation of a signature is much harder
- More advanced metamorphs also change their runtime behavior, not just the code

Obfuscation Techniques

obfuscating the virus’ actions

6. 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,
  - e.g. Timestamp, File size

7. Armoring

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

Obfuscation Techniques

obfuscating the virus’ actions

8. Tunneling

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

Obfuscation Techniques

obfuscating the virus’ actions

9. 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

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.)

Trojan Examples

- 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

Worm

- A 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 execution, remote login
- First replicates then does the 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 them, copies itself and executes

Worm: possible damages

These can be enormous

- 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 Worm (2003)

- exploited buffer-overflow vulnerability in Microsoft SQL Server
  - Vulnerability itself disclosed 24th July 2002, but lot of servers remained unpatched
- Infected servers started infecting other servers, generating lot of traffic on core routers.
  - Routers themselves generated even more traffic for updating routing tables.
- Consequences
  - ATM systems not available
  - Phone network overloaded (no 911!)
  - 5 DNS root servers down
  - Planes delayed

Rootkit

- A rootkit is a stealthy type of malware that alters the system utilities or the OS itself to prevent detection.
  - Often employed by malware like viruses and Trojan to hide their detection
- Example hiding techniques
  - A rootkit can infect the Windows Task Manager (that lists the currently running processes) and hide by removing itself from the list.
  - A rootkit can detect the Windows Explorer and hide itself from being listed when the user browses files.

Rootkit

- With root access, an attacker has complete control of the system and can add or change programs and files, monitor processes, send and receive network traffic, and get backdoor access on demand.
- The name rootkit can be misleading — it does not provide root access by itself. Instead, rootkits actually depend on that attacker/malicious user already has already exploited the target and gained root access into the system. Once the attacker has root access to the system, rootkits will make sure that the attacker can maintain their control on the target.

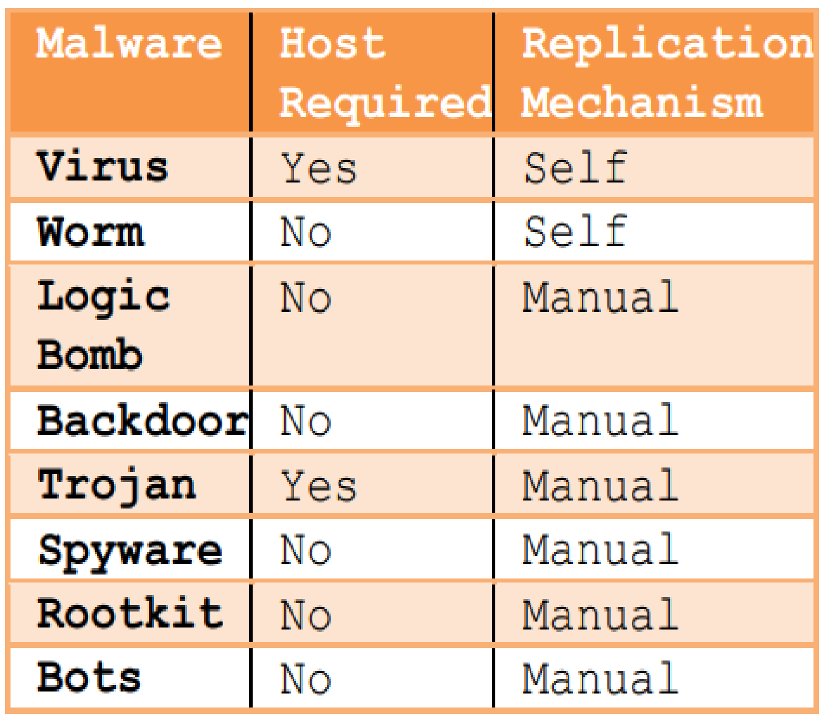
Rootkit installation location

- Persistent: Activates each time the system boots. The rootkit must store code in a persistent store, such as file system, and configure a method by which the code executes without user intervention (e.g. autorun).
- Memory based: Has no persistent code and therefore cannot survive a reboot. Harder to detect.

Rootkit mode of interception

- User mode: Intercepts calls to APIs (e.g. file listing) and modifies returned results.
- Kernel mode: Can intercept calls to native APIs in kernel mode. The rootkit can also hide the presence of a malware process by removing it from the kernel’s list of active processes.

Comparison of Malware Types



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 is first loaded on to a host file (executable, script, document etc.), it will run only when the user executes or opens the host file.

What is infected by malware?

- Executables
- Interpreted scripts
- Kernel
- System services
- MBR (Master Boot Record)
- Hypervisor

How exe files are altered

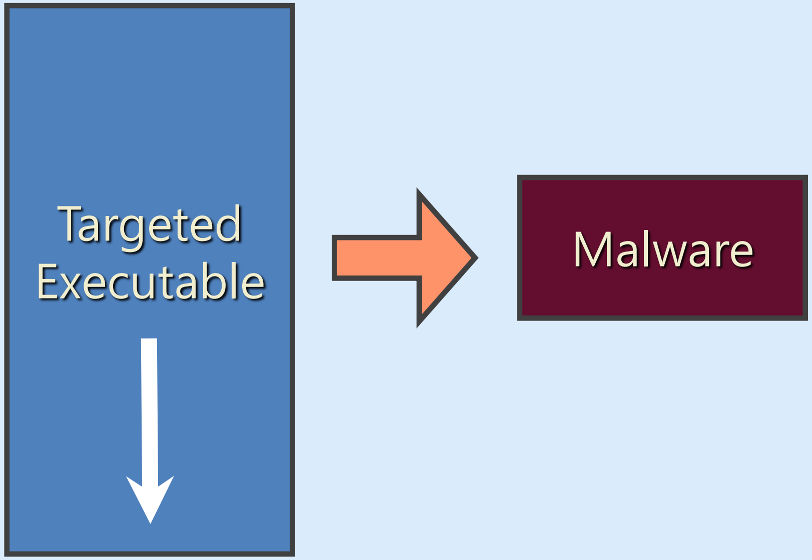
- When a compiled malware infects an executable file, it may:
  - replace all of existing code

  OR
  - add the malicious code while keeping original code (and functionality) intact

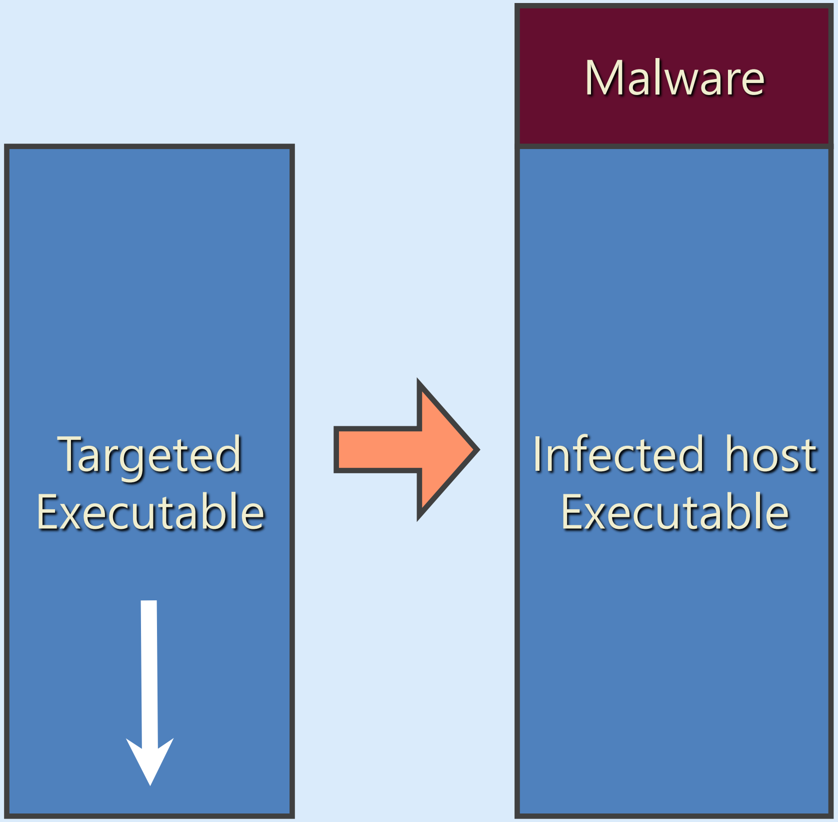
Some common methods employed by malware are shown next…

Overwriting malware

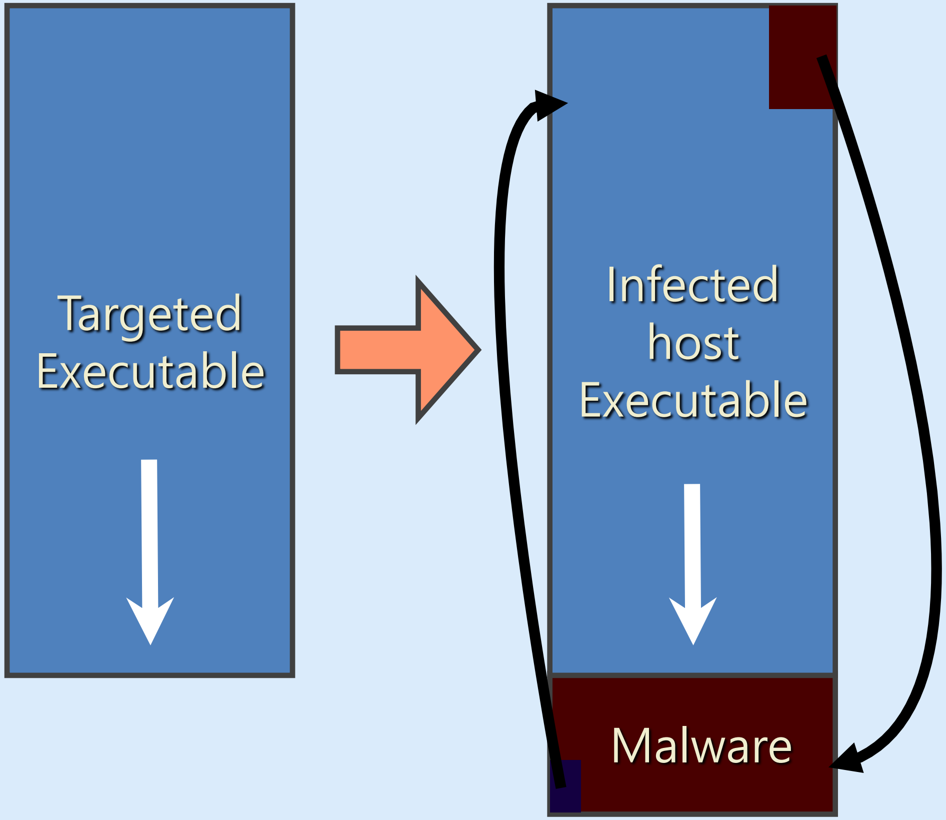
- After infection, it will effectively destroy the original program code by overwriting data in the system's memory.
  - TRj.reboot virus: It can restart the user's computer, and was active in targeting Windows NT and Windows 2000 systems in the 2000s.
  - Trivial.88.D virus: A 'direct action virus'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



Prepending malware

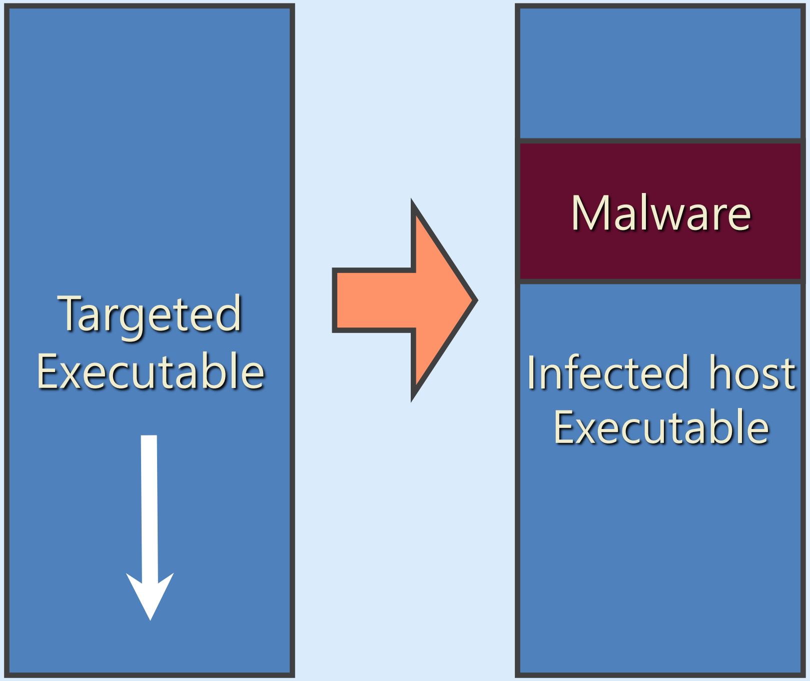


Appending malware



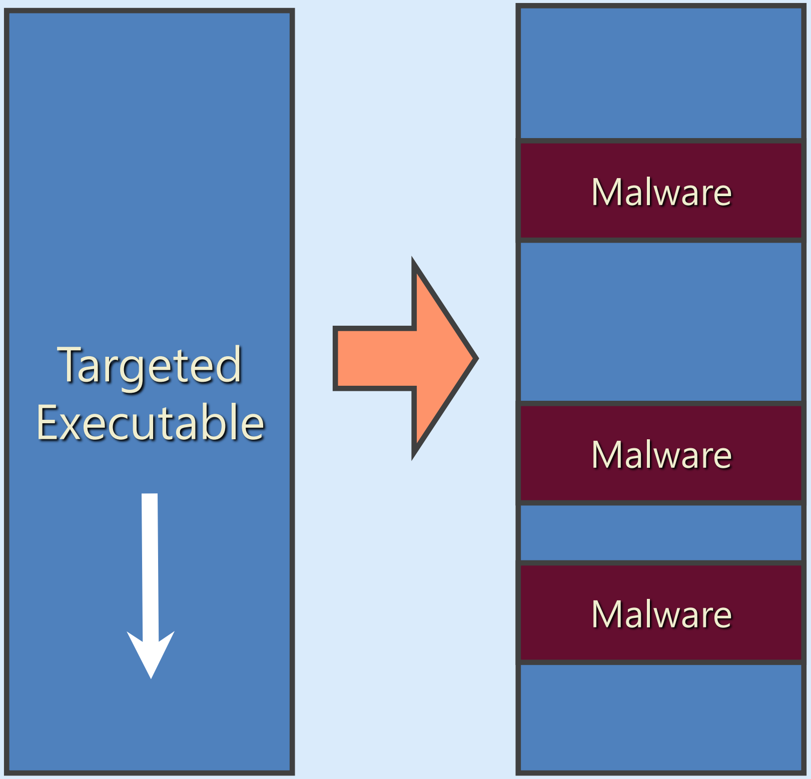
Cavity malware

- 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.



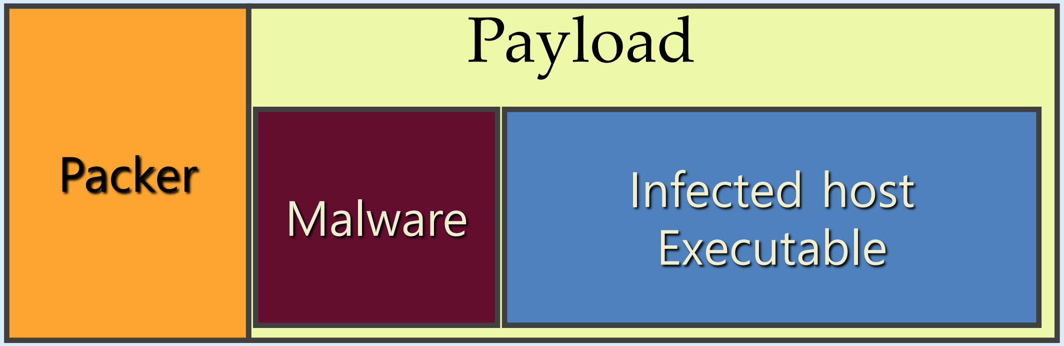
Multi-cavity malware

- In case the malware does not find one cavity big enough to hold the complete code



Packer

- A packer compresses and optionally 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 inspecting file metadata and dependencies, performing string searches, peeking into the code using a hex editor.
- Disassembling the malware
  - Disassemblers (e.g. IDA Pro) convert machine code to higher-level assembly code.
- How about packer malwares?
  - Tools like PEiD, Mandiant Capa can tell you about "packed" code
  - Unpacking is needed before further static analysis

Dynamic Analysis

- Conducted by observing and manipulating malware as it runs
- Needs a safe (**sandboxed**) environment to analyze (run) the code
- Requires monitoring the system
  - Registry files activity
  - File and process/system level activity
  - Network level activity
- Some tools
  - Wireshark
  - SysInternals process monitor (windows)
  - Netstat (linux) or ResMon (Windows)
- Requires analysis while the code is being run using tools like WinDbg
- Malware files are fingerprinted before analysis. Just in case malware analysis is being expected by the (malware) developer.

Sandboxing

- The process of isolating a program on the hard drive in order to minimize or eliminate the exposure to other apps and critical system.
- Usually programs and applications interact with multiple parts of operating system and use shared resources like storage, memory and CPU sometimes causing conflicts.
- A malware if present can utilize such vulnerabilities to cause a disaster.
- Sandboxing helps to reduce the impact that an individual program will have on the system.

Sandboxing Examples

- Browser sandboxing
  - Web browsers like Chrome and Safari open each page in separate sandboxes

- Virtual Machines
  - It is also called manual sandboxing
  - Purposely configures the system to sandbox all applications within the VM.
  - Virtualbox, VMware

- Windows Sandbox
  - A temporary instance of host machine

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.

Malware scanners

- The first generation scanner
  - Malware signature (fixed bit patterns)
  - Maintains a record of the length of programs
- The second generation scanner
  - Looks for fragments of malware-like code
  - Checksum or hash of files (integrity checking)
- The third generation scanner
  - Identify a malware by its actions
  - Like an intrusion detection system
- The fourth generation scanner
  - Include a variety of anti-malware techniques

More countermeasures

- 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

Malware Prevention

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

Malware Prevention

- 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

Malware Prevention

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