Information Security
CS3002
(Sections BDS-7A/B)
Lecture 14

Instructor: Dr. Syed Mohammad Irteza

Assistant Professor, Department of Computer Science

07 October, 2024

Memory based Classification

- This categorization is based on their behavior while they operate in memory.
  - Resident
  - Temporary Resident
  - Swapping Mode
  - Non-Resident
  - User Process
  - Kernel Process

Memory based Classification (cont'd)

- **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 removes itself out of memory when a certain event occurs
  - Extremely difficult to detect

Memory based Classification (cont'd)

- **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.

Memory based Classification (cont'd)

- **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 'Direct action viruses'.

Memory based Classification (cont'd)

- **User Process**:
  - These viruses run as a user process and infect 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 cases, the virus exist as a DLL and uses DLL injection method (through registry keys) to load the DLL into the process.
  - Autorun is an example of this type of virus

Memory based Classification (cont'd)

- **Kernel 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/superuser 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

Obfuscation Techniques (Cont'd)

- **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 covered with a simple brute force by the virus itself

Obfuscation Techniques (Cont'd)

- **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

Obfuscation Techniques (Cont'd)

- **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

Obfuscation Techniques (Cont’d)

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

Obfuscation Techniques (Cont'd)

- **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 a 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 a 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
  - Piece of code with a unique binary pattern that identifies a computer virus or family of viruses
- The companies *include* the *new signature* in their *database*
- Their scanner *now can detect* the virus

Trojan (Definition)

- 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
  - Backdoor Trojans can be deployed to multiple devices in order to create a botnet, or zombie network, that can be *used to carry out a DDoS attack*.
- 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 does 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

- 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 (article)
- 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

Size: 376 bytes of malicious code

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

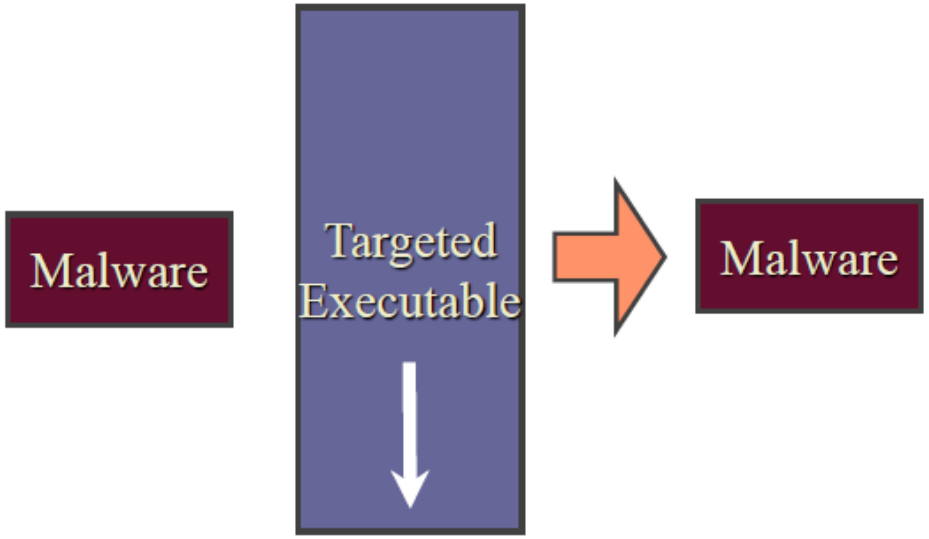
- Host required: malware needs user interaction

What they infect?

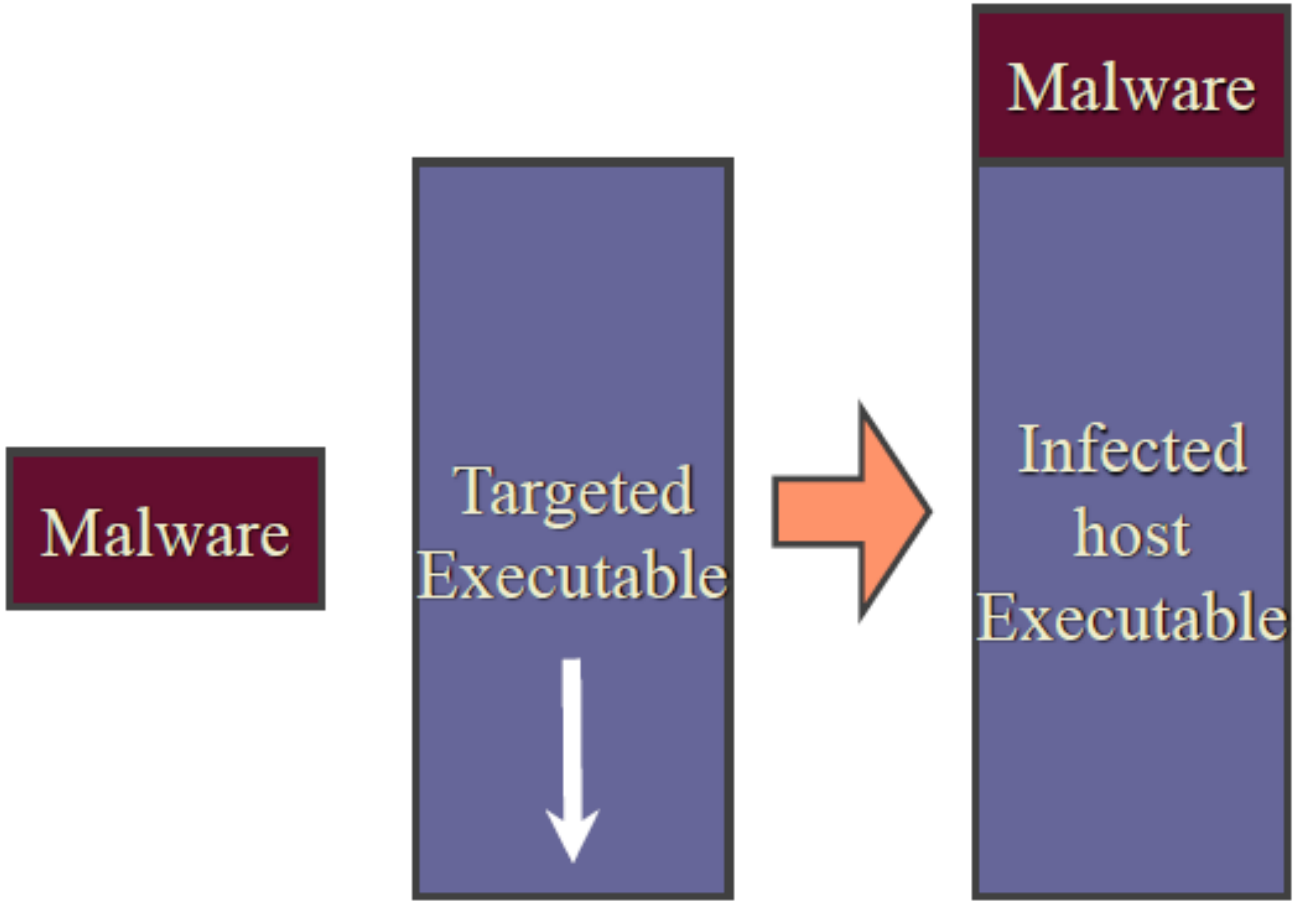
- Executable
- Interpreted file
- Kernel
- Service
- MBR (Master Boot Record)
- Hypervisor

Overwriting malware

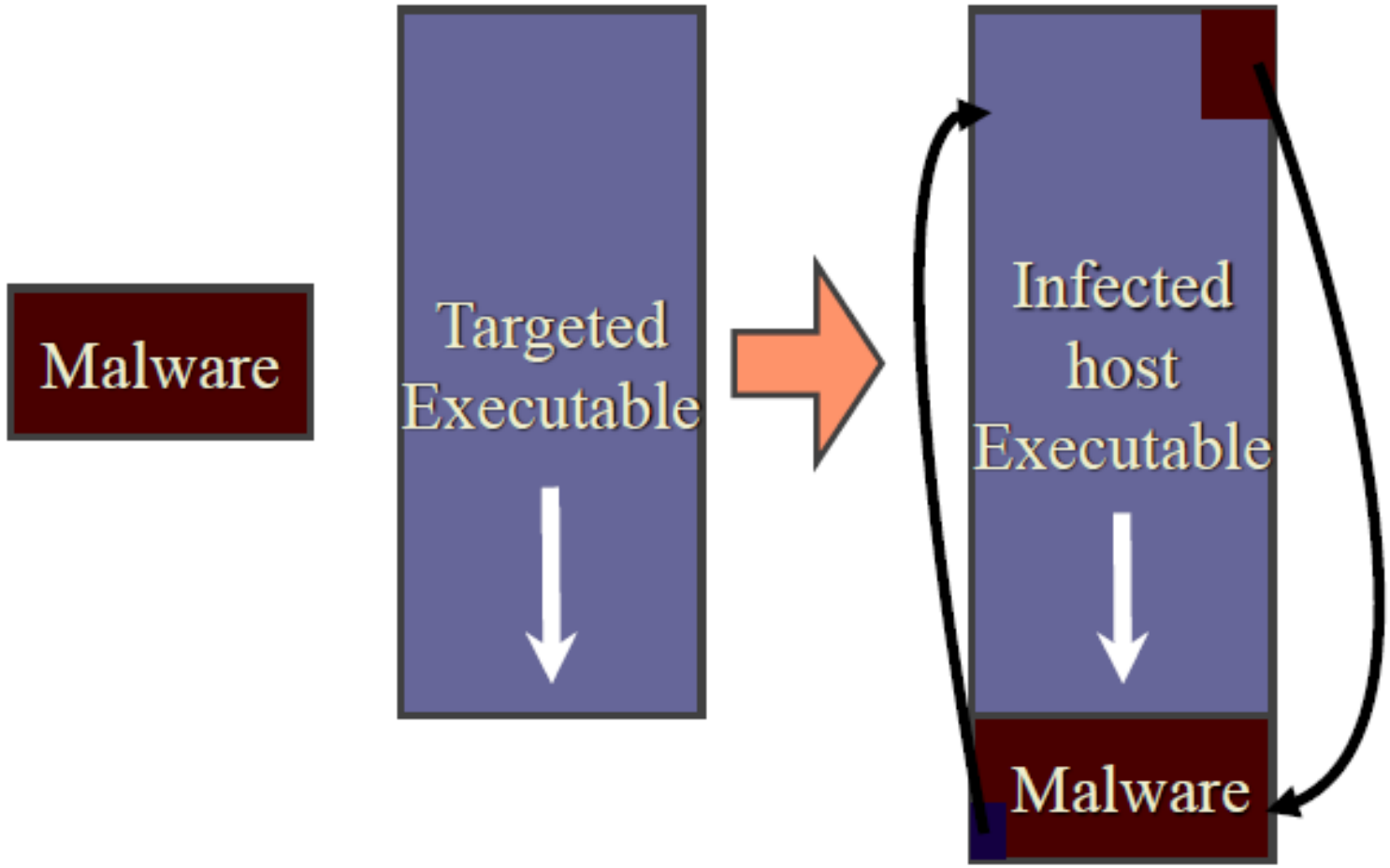
- After infection, it will effectively *destroy the original program code by overwriting data in the system's memory*.
- *Trj.reboot* virus: It can restart the user's computer, and was active in targeting WindowsNT and Windows 2000 systems in the 2000s.
- *Trivial.88.D* virus: A 'direct-action virus'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



Pre-pending malware

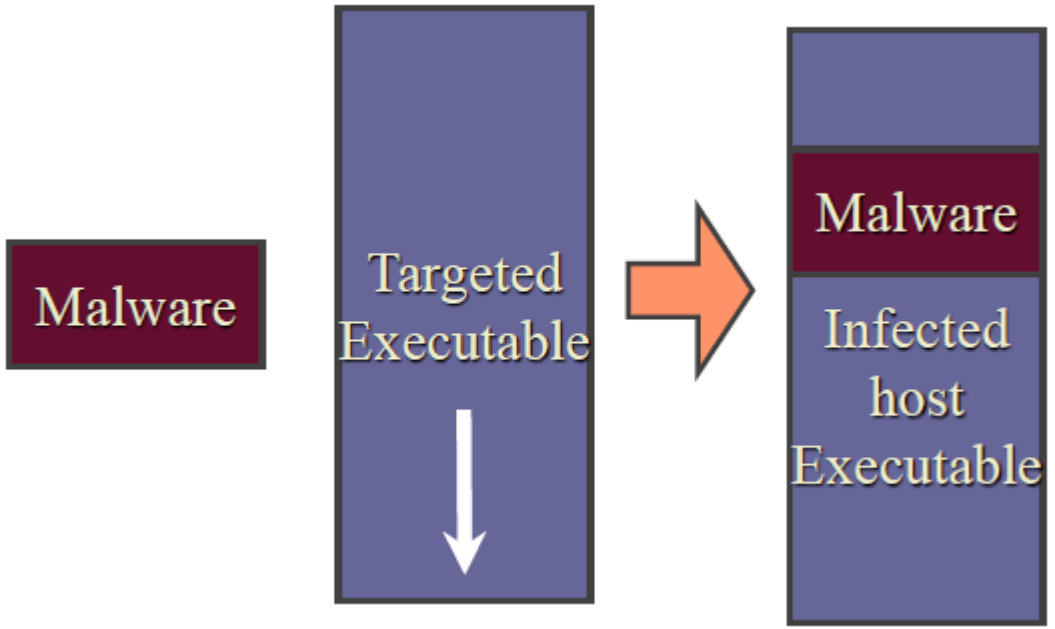


Appending malware

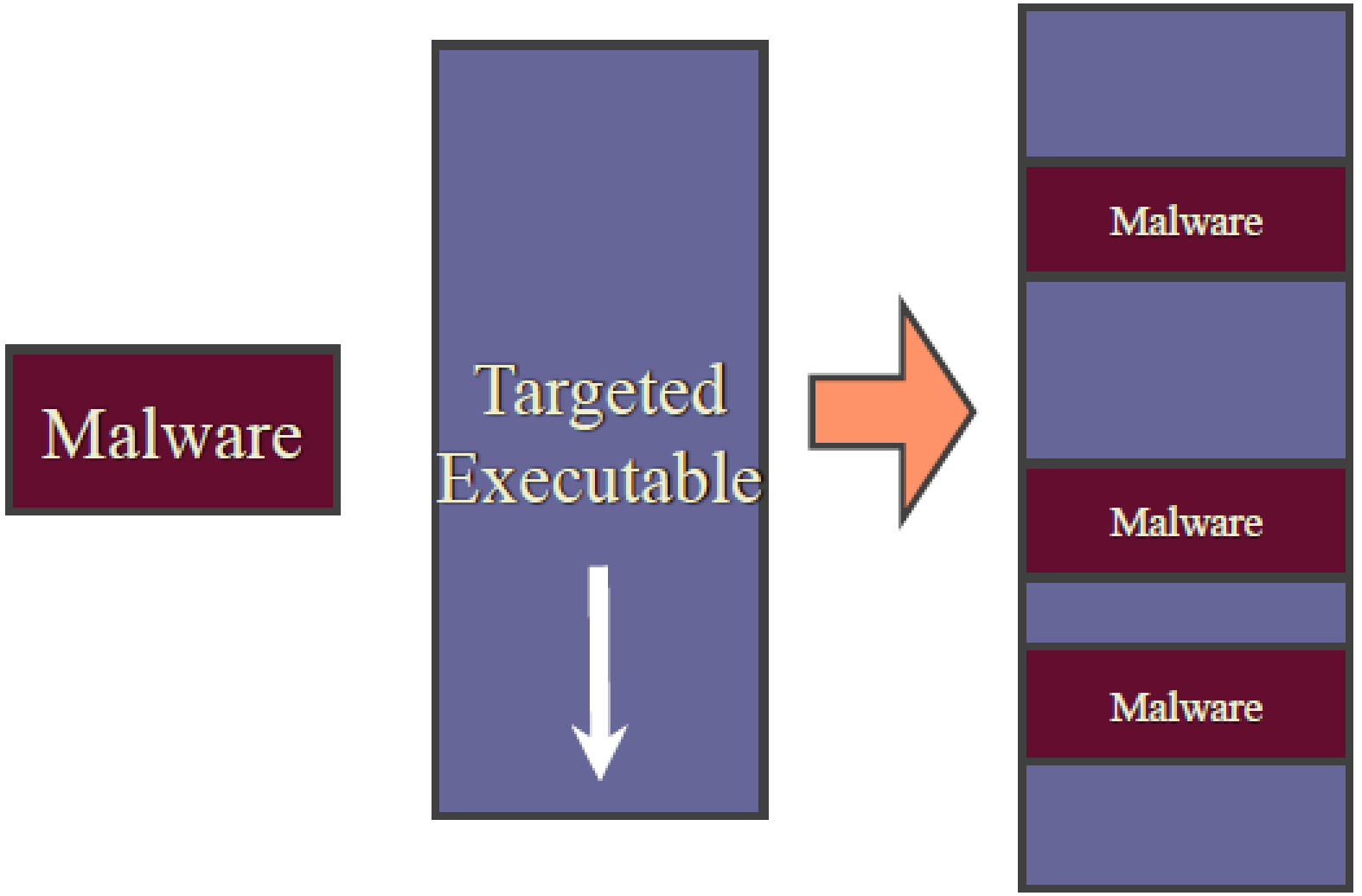


Cavity malware

- 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.

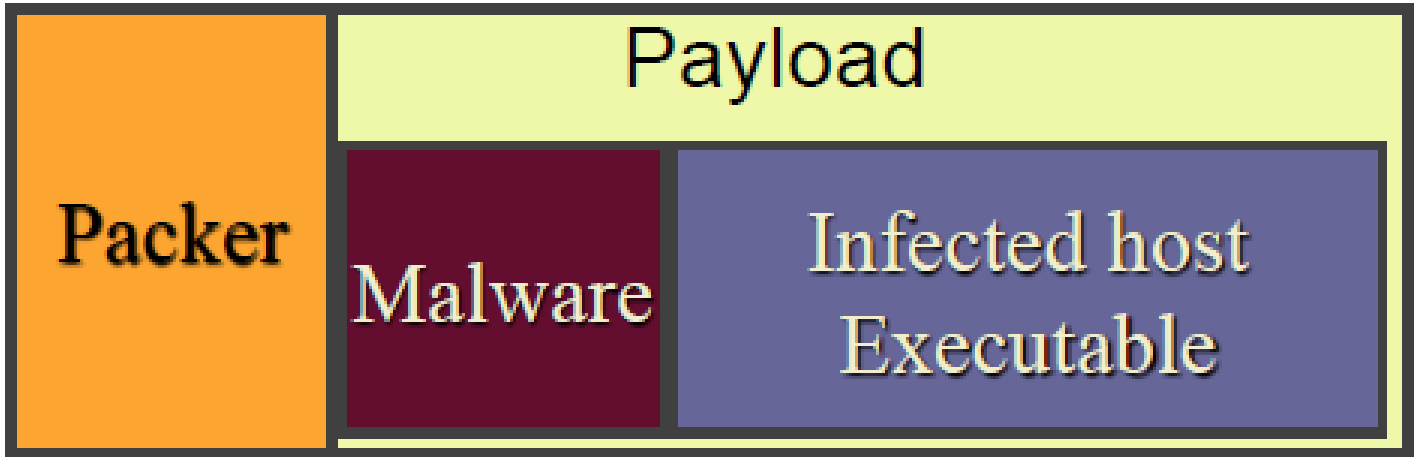


Multi-cavity malware



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
  - Indicators Of Compromise (IOCs) extraction
    - See Malware Analysis: Steps & Examples – CrowdStrike
- Types:
  - Static
  - Dynamic