Intro to IT Security

CS306C—Fall 2022

Prof. Antonio R. Nicolosi

Antonio.Nicolosi@stevens.edu

MALWARE

Grading (revised)

10%: In-lab quizzes

20%: Lab performance & attendance

20%: Project

25%: Midterm*

25%: Final

$$examGrade = \max(finalGrade, avg(midGrade, finalGrade))$$

MALWARE: MALicious SoftWARE

- Malicious code: (part of) a program with unanticipated or undesired effects
- Agent: writer of the program or person who caused its distribution
- NIST '05 defines malware as:

  > "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 or otherwise annoying or disrupting the victim."

Kind of Malicious Code

Name	Description
Backdoor (trapdoor)	Any mechanisms that bypasses a normal security check; it may allow unauthorized access to functionality in a program, or onto a compromised system.
Flooders (DoS client)	Used to generate a large volume of data to attack networked computer systems, by carrying out some form of denial-of-service (DoS) attack.
Keyloggers	Captures keystrokes on a compromised system.
Logic bomb	Code inserted into malware by an intruder. A logic bomb lies dormant until a predefined condition is met; the code then triggers an unauthorized act.
Macro Virus	A type of virus that uses macro or scripting code, typically embedded in a document, and triggered when the document is viewed or edited, to run and replicate itself into other such documents.
Mobile code	Software (e.g., script, macro, or other portable instruction) that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics.
Rootkit	Set of hacker tools used after attacker has broken into a computer system and gained root-level access.
Spammer programs	Used to send large volumes of unwanted e-mail.
Spyware	Software that collects information from a computer and transmits it to another system by monitoring keystrokes, screen data and/or network traffic; or by scanning files on the system for sensitive information.
Trojan horse	A computer program that appears to have a useful function, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entity that invokes the Trojan horse program.
Virus	Malware that, when executed, tries to replicate itself into other executable machine or script code; when it succeeds the code is said to be infected. When the infected code is executed, the virus also executes.
Worm	A computer program that can run independently and can propagate a complete working version of itself onto other hosts on a network, usually by exploiting software vulnerabilities in the target system.
Zombie, bot	Program activated on an infected machine that is activated to launch attacks on other machines.

Classification of Malicious Code

- Two main dimensions:
  - how it spreads or propagates to reach the desired targets
  - on the actions or payloads it performs once a target is reached

Classification of Malicious Code

- Earlier approaches to malware classification distinguished between
  - those that need a host program (parasitic code such as viruses)
  - those that are independent, self-contained programs (worms, trojans, and bots)
  - malware that does not replicate (trojans and spam e-mail)
  - malware that does replicate (viruses and worms)

Viruses

- Piece of software that infects programs
  - Modifies them to include a copy of the virus
  - Replicates and goes on to infect other content
  - Easily spread through network environments
- When attached to an executable program a virus can do anything that the program is permitted to do
  - Executes secretly when the host program is run
- Specific to operating system and hardware
  - Takes advantage of their details and weaknesses

Virus Components

A computer virus has three parts:

- Infection mechanism
  - means by which a virus spreads or propagates
  - also referred to as the *infection vector*
- Trigger
  - event or condition that determines when the payload is activated or delivered
  - sometimes known as a *logic bomb*
- Payload
  - what the virus does (besides spreading)
  - may involve damage or benign but noticeable activity

Virus Components

A computer virus has three parts:

- Infection mechanism
  - means by which a virus spreads or propagates
  - also referred to as the *infection vector*
- Trigger
  - event or condition that determines when the payload is activated or delivered
  - sometimes known as a *logic bomb*
- Payload
  - what the virus does (besides spreading)
  - may involve damage or benign but noticeable activity

Virus Components

A computer virus has three parts:

- Infection mechanism
  - means by which a virus spreads or propagates
  - also referred to as the *infection vector*
- Trigger
  - event or condition that determines when the payload is activated or delivered
  - sometimes known as a *logic bomb*
- Payload
  - what the virus does (besides spreading)
  - may involve damage or benign but noticeable activity

Virus Life-Cycle

A typical virus goes through the following four phases:

- Dormant phase
  - virus is idle
  - will eventually be activated by some event
  - not all viruses have this stage
- Propagation phase
  - virus places a copy of itself into other programs or into certain system areas on the disk
  - may not be identical to the propagating version
  - each infected program will now contain a clone of the virus which will itself enter a propagation phase
- Triggering phase
  - virus is activated to perform the function for which it was intended
  - can be caused by a variety of system events
- Execution phase
  - function is performed
  - may be harmless or damaging

Virus Life-Cycle

A typical virus goes through the following four phases:

- Dormant phase
  - virus is idle
  - will eventually be activated by some event
  - not all viruses have this stage
- Propagation phase
  - virus places a copy of itself into other programs or into certain system areas on the disk
  - may not be identical to the propagating version
  - each infected program will now contain a clone of the virus which will itself enter a propagation phase
- Triggering phase
  - virus is activated to perform the function for which it was intended
  - can be caused by a variety of system events
- Execution phase
  - function is performed
  - may be harmless or damaging

Virus Life-Cycle

A typical virus goes through the following four phases:

- Dormant phase
  - virus is idle
  - will eventually be activated by some event
  - not all viruses have this stage
- Propagation phase
  - virus places a copy of itself into other programs or into certain system areas on the disk
  - may not be identical to the propagating version
  - each infected program will now contain a clone of the virus which will itself enter a propagation phase
- Triggering phase
  - virus is activated to perform the function for which it was intended
  - can be caused by a variety of system events
- Execution phase
  - function is performed
  - may be harmless or damaging

Virus Life-Cycle

A typical virus goes through the following four phases:

- Dormant phase
  - virus is idle
  - will eventually be activated by some event
  - not all viruses have this stage
- Propagation phase
  - virus places a copy of itself into other programs or into certain system areas on the disk
  - may not be identical to the propagating version
  - each infected program will now contain a clone of the virus which will itself enter a propagation phase
- Triggering phase
  - virus is activated to perform the function for which it was intended
  - can be caused by a variety of system events
- Execution phase
  - function is performed
  - may be harmless or damaging

Virus Structure

- A virus attaches itself to a program:
  - It copies itself before the program's first executable instruction
  - It runs the program, but has control before and after its execution
  - It replaces some of the program, integrating itself into the original code

Virus Structure: Example

```
    program V :=

{goto main;
    1234567;

    subroutine infect-executable :=
        {loop:
        file := get-random-executable-file;
        if (first-line-of-file = 1234567)
          then goto loop
          else prepend V to file; }

    subroutine do-damage :=
        {whatever damage is to be done}

    subroutine trigger-pulled :=
        {return true if some condition holds}

main:   main-program :=
        {infect-executable;
        if trigger-pulled then do-damage;
        goto next;}

next:

}
```

Virus Structure: Example 2

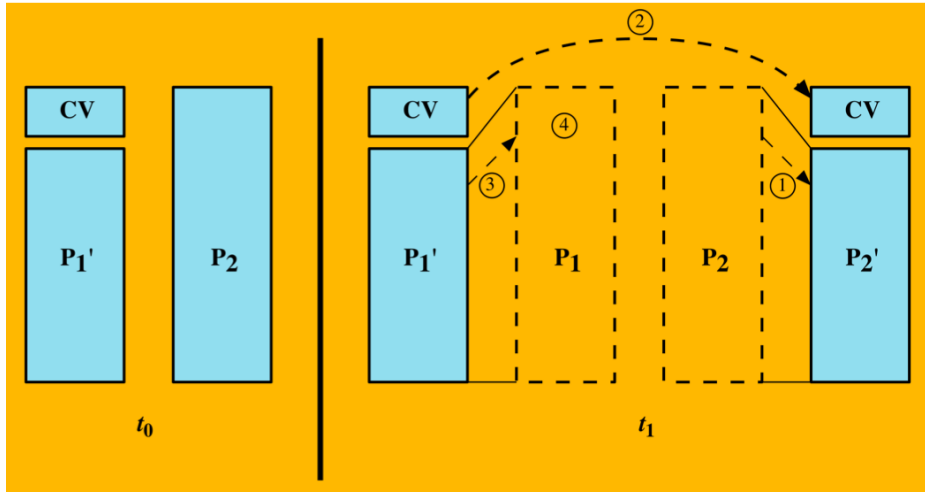
```
    program CV :=

{goto main;
    01234567;

    subroutine infect-executable :=
          {loop:
               file := get-random-executable-file;
          if (first-line-of-file = 01234567) then goto loop;
       (1)     compress file;
       (2)     prepend CV to file;
       }

main:   main-program :=
          {if ask-permission then infect-executable;
       (3)     uncompress rest-of-file;
       (4)     run uncompressed file;}
       }
```

Virus Structure: Example 2



How to Detect a Virus

- Signature: characteristics of a virus
- Virus Scanner: using the signature, try to detect and (possibly) remove the virus
- Storage Patterns
  - at the beginning of the file
  - a handful of instructions at the top, followed by a JUMP to more detailed instruction elsewhere
- Virus can attach itself to a file: file's size grows
- Virus may obliterate all or part of the underlying program: program's functioning will be impaired

Classification of Viruses (by Target)

- Boot sector infector
  - Infects a master boot record
  - Spreads when system is booted
- File infector
  - Infects executable files
- Macro virus
  - Infects files with macro code or scripting code that is interpreted by an application

Classification of Viruses (by Concealment Strategy)

- Encrypted Virus
  - a portion of the virus creates a random encryption key
  - the key is stored with the virus
  - when an infected program is invoked, the virus uses the stored random key to decrypt the virus
  - uses encryption under various keys to make the stored forms of the virus different (no constant bit pattern)
- Stealth virus
  - a form of virus explicitly designed to hide itself from detection by anti-virus software
- Polymorphic Viruses
  - mutates with every infection
- Metamorphic Viruses
  - a virus that mutates and rewrites itself completely at each iteration and may change behavior as well as appearance
    - add *no-ops* to distort any pattern
    - randomly reposition all parts of itself and randomly change all fixed data

Classification of Viruses (by Concealment Strategy)

- Encrypted Virus
  - a portion of the virus creates a random encryption key
  - the key is stored with the virus
  - when an infected program is invoked, the virus uses the stored random key to decrypt the virus
  - uses encryption under various keys to make the stored forms of the virus different (no constant bit pattern)
- Stealth virus
  - a form of virus explicitly designed to hide itself from detection by anti-virus software
- Polymorphic Viruses
  - mutates with every infection
- Metamorphic Viruses
  - a virus that mutates and rewrites itself completely at each iteration and may change behavior as well as appearance
    - add *no-ops* to distort any pattern
    - randomly reposition all parts of itself and randomly change all fixed data

Classification of Viruses (by Concealment Strategy)

- Encrypted Virus
  - a portion of the virus creates a random encryption key
  - the key is stored with the virus
  - when an infected program is invoked, the virus uses the stored random key to decrypt the virus
  - uses encryption under various keys to make the stored forms of the virus different (no constant bit pattern)
- Stealth virus
  - a form of virus explicitly designed to hide itself from detection by anti-virus software
- Polymorphic Viruses
  - mutates with every infection
- Metamorphic Viruses
  - a virus that mutates and rewrites itself completely at each iteration and may change behavior as well as appearance
    - add *no-ops* to distort any pattern
    - randomly reposition all parts of itself and randomly change all fixed data

Classification of Viruses (by Concealment Strategy)

- Encrypted Virus
  - a portion of the virus creates a random encryption key
  - the key is stored with the virus
  - when an infected program is invoked, the virus uses the stored random key to decrypt the virus
  - uses encryption under various keys to make the stored forms of the virus different (no constant bit pattern)
- Stealth virus
  - a form of virus explicitly designed to hide itself from detection by anti-virus software
- Polymorphic Viruses
  - mutates with every infection
- Metamorphic Viruses
  - a virus that mutates and rewrites itself completely at each iteration and may change behavior as well as appearance
    - add *no-ops* to distort any pattern
    - randomly reposition all parts of itself and randomly change all fixed data

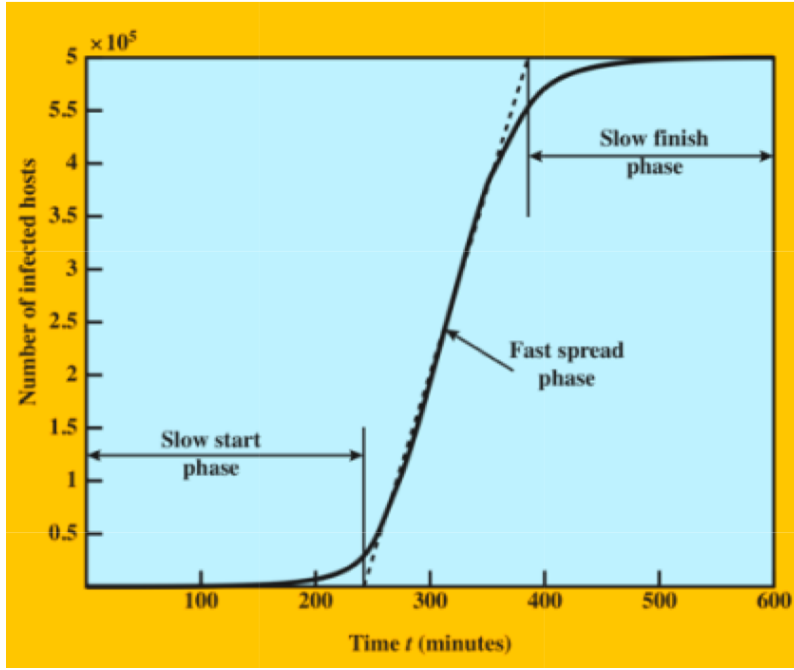
Macro/Scripting Code Viruses

- Very common in mid-1990s
  - platform independent
  - infect documents (not executable portions of code)
  - easily spread (email)
- Exploit macro capability of MS Office applications
  - more recent releases of products include protection
- Various anti-virus programs have been developed so these are no longer the predominant virus threat

Worms

- Program that actively seeks out more machines to infect and each infected machine serves as an automated launching pad for attacks on other machines
- Exploits software vulnerabilities in client or server programs
- Can use network connections to spread from system to system
- Spreads through shared media (USB drives, CD, DVD data disks)
- E-mail worms spread in macro or script code included in attachments and instant messenger file transfers
- Upon activation the worm may replicate and propagate again
- Usually carries some form of payload
- First known implementation was done in Xerox Palo Alto Labs in the early 1980s
  - non malicious
  - searching for idle systems to use to run a computationally intensive task

Worm Propagation Model



A Bit of History: Morris Worm

- Worm was released in 1988 by Robert Morris
  - Graduate student at Cornell, son of NSA chief scientist
  - Convicted under Computer Fraud and Abuse Act, sentenced to 3 years of probation and 400 hours of community service
  - Now an EECS professor at MIT
- Worm was intended to propagate slowly and harmlessly measure the size of the Internet
- Due to a coding error, it created new copies as fast as it could and overloaded infected machines
- $10-100M worth of damage

Morris Worm and Buffer Overflow

- One of the worm's propagation techniques was a buffer overflow attack against a vulnerable version of fingerd on UNIX systems
  - By sending special string to finger daemon, worm caused it to execute code creating a new worm copy
  - Unable to determine remote OS version, worm also attacked fingerd on Suns running BSD, causing them to crash (instead of spawning a new copy)

Stuxnet

- In 2010, the Stuxnet worm was detected, though it had been spreading quietly for some time previously
- It deliberately restricted its rate of spread to reduce its chance of detection
- It targeted industrial control systems (Iranian nuclear program), with the aim of disrupting the operation of their equipment
- A range of propagation mechanisms (USB drives, network file shares)
- Exploited four zero-day vulnerabilities

Worm Technology

- Multiplatform
  - Not limited to Windows machines but can attack a variety of platforms, or exploit macro or scripting languages
- Multi-exploit
  - Penetrate systems using exploits against Web servers, browsers, e-mail, file sharing
- Polymorphic
  - To evade detection, skip past filters, and foil real-time analysis, worms adopt the virus polymorphic technique
- Metamorphic
  - Have a repertoire of behavior patterns that are unleashed at different stages of propagation

Mobile Phone Worms

- First discovery was Cabir worm in 2004
- Lasco and CommWarrior in 2005
- Communicate through Bluetooth wireless connections or MMS
- Target is the smartphone
- Can completely disable the phone, delete data on the phone, or force the device to send costly messages
- CommWarrior replicates by means of Bluetooth to other phones, sends itself as an MMS file to contacts and as an auto reply to incoming text messages

BotNets

- A bot is a program that secretly takes over another computer (connected to it via the network)
- Uses that computer to launch or manage attacks
- The bot is typically planted on a large collection of computers, referred to as a **botnet**

Some Uses of Bots

- Distributed denial-service (DDoS) attacks
- Spamming
- Sniffing traffic
- Keylogging
- Spreading new malware
- Manipulate online pools/games (every bot has a distinct IP address)

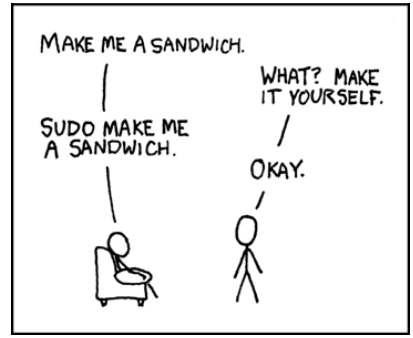
Worm *vs.* Bots

The difference is the remote control facility:

- A worm propagates itself, and activate itself
- A bot is controlled from some central facility (at least initially)
- Typical means of implementing the remote control facility is on an Internet Relay Chat (IRC) server
  - bots join a specific channel on this server and treat incoming messages as commands
  - more recent botnets use covert communication channels via protocols (HTTP)
  - distributed control mechanisms use peer-to-peer protocols to avoid a single point of failure

Rootkits

- Set of hidden programs installed on a system to maintain covert access to that system
- Hides by subverting the mechanisms that monitor and report on the processes, files, and registries on a computer
- Gives administrator (or root) privileges to attacker
  - can add or change programs and files, monitor processes, send and receive network traffic, and get backdoor access on demand
- Unfortunately people are often casual about when to use root privileges...



Rootkits Classification Characteristics

- Persistent
  - Activates each time the system boots
  - Must store code in a persistent area, such as the registry or file system, and executes without user intervention
  - Easier to detect
- Memory based
  - Has no persistent code and therefore cannot survive a reboot
  - Harder to detect
- User mode
  - Intercepts calls to APIs (application program interfaces) and modifies returned results
- Kernel mode
  - Can intercept calls to native APIs in kernel mode
  - Hide a malware process by removing it from the kernel's list of active processes
- Virtual machine based
  - Installs a lightweight virtual machine monitor, and then runs the operating system in a virtual machine above it
  - Can intercept and modify states and events occurring in the virtualized system
- External mode
  - Located in BIOS or system management mode, where it can directly access hardware

Rootkits Classification Characteristics

- Persistent
  - Activates each time the system boots
  - Must store code in a persistent area, such as the registry or file system, and executes without user intervention
  - Easier to detect
- Memory based
  - Has no persistent code and therefore cannot survive a reboot
  - Harder to detect
- User mode
  - Intercepts calls to APIs (application program interfaces) and modifies returned results
- Kernel mode
  - Can intercept calls to native APIs in kernel mode
  - Hide a malware process by removing it from the kernel's list of active processes
- Virtual machine based
  - Installs a lightweight virtual machine monitor, and then runs the operating system in a virtual machine above it
  - Can intercept and modify states and events occurring in the virtualized system
- External mode
  - Located in BIOS or system management mode, where it can directly access hardware

Rootkits Classification Characteristics

- Persistent
  - Activates each time the system boots
  - Must store code in a persistent area, such as the registry or file system, and executes without user intervention
  - Easier to detect
- Memory based
  - Has no persistent code and therefore cannot survive a reboot
  - Harder to detect
- User mode
  - Intercepts calls to APIs (application program interfaces) and modifies returned results
- Kernel mode
  - Can intercept calls to native APIs in kernel mode
  - Hide a malware process by removing it from the kernel's list of active processes
- Virtual machine based
  - Installs a lightweight virtual machine monitor, and then runs the operating system in a virtual machine above it
  - Can intercept and modify states and events occurring in the virtualized system
- External mode
  - Located in BIOS or system management mode, where it can directly access hardware

Rootkits Classification Characteristics

- Persistent
  - Activates each time the system boots
  - Must store code in a persistent area, such as the registry or file system, and executes without user intervention
  - Easier to detect
- Memory based
  - Has no persistent code and therefore cannot survive a reboot
  - Harder to detect
- User mode
  - Intercepts calls to APIs (application program interfaces) and modifies returned results
- Kernel mode
  - Can intercept calls to native APIs in kernel mode
  - Hide a malware process by removing it from the kernel's list of active processes
- Virtual machine based
  - Installs a lightweight virtual machine monitor, and then runs the operating system in a virtual machine above it
  - Can intercept and modify states and events occurring in the virtualized system
- External mode
  - Located in BIOS or system management mode, where it can directly access hardware

Rootkits Classification Characteristics

- Persistent
  - Activates each time the system boots
  - Must store code in a persistent area, such as the registry or file system, and executes without user intervention
  - Easier to detect
- Memory based
  - Has no persistent code and therefore cannot survive a reboot
  - Harder to detect
- User mode
  - Intercepts calls to APIs (application program interfaces) and modifies returned results
- Kernel mode
  - Can intercept calls to native APIs in kernel mode
  - Hide a malware process by removing it from the kernel's list of active processes
- Virtual machine based
  - Installs a lightweight virtual machine monitor, and then runs the operating system in a virtual machine above it
  - Can intercept and modify states and events occurring in the virtualized system
- External mode
  - Located in BIOS or system management mode, where it can directly access hardware

Rootkits Classification Characteristics

- Persistent
  - Activates each time the system boots
  - Must store code in a persistent area, such as the registry or file system, and executes without user intervention
  - Easier to detect
- Memory based
  - Has no persistent code and therefore cannot survive a reboot
  - Harder to detect
- User mode
  - Intercepts calls to APIs (application program interfaces) and modifies returned results
- Kernel mode
  - Can intercept calls to native APIs in kernel mode
  - Hide a malware process by removing it from the kernel's list of active processes
- Virtual machine based
  - Installs a lightweight virtual machine monitor, and then runs the operating system in a virtual machine above it
  - Can intercept and modify states and events occurring in the virtualized system
- External mode
  - Located in BIOS or system management mode, where it can directly access hardware