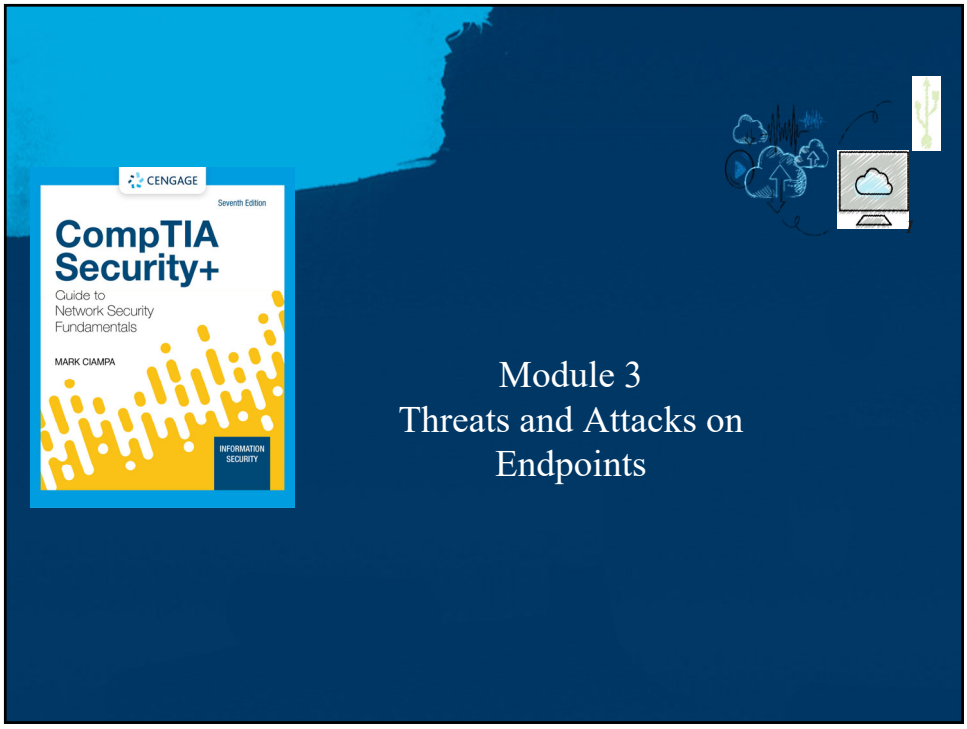


Module 3
Threats and Attacks on Endpoints

Module Objectives

- By the end of this module, you should be able to:
  - Identify the different types of attacks using malware
  - Define application attacks
  - Explain how threat actors use application attacks
  - Define adversarial artificial intelligence attacks

Attacks Using Malware

- **Malware**
  - Most often used as the general term that refers to a wide variety of damaging software programs
  - Enters a computer system without the user's knowledge or consent and then performs an unwanted and harmful action
  - Continually evolving to avoid detection by improved security measures
- Classifying malware based on the **primary action*** that the malware performs:
  - Imprison
  - Launch
  - Snoop
  - Deceive
  - Evade

*** Malware may have more than one action**

3

Imprison

- **Primary Action- Imprison:**
  - Attempt to take away the freedom of the user to do what they want
  - Types: Ransomware, Cryptomalware
  - **Ransomware**
    - Prevents a user's endpoint device from properly and fully functioning until a fee is paid
    - Some pretends to come from a law enforcement agency
    - Others pretend to come from a software vendor and displays a fictitious warning that a license has expired

Figure 3-1 Blocker ransomware message

Figure 3-2 Ransomware computer infection

4

Imprison

- **Primary Action- Imprison:**
  - **Cryptomalware**
    - Malware that imprisons users and encrypts all files on the device so that none of them can be opened
    - The cost for the key to unlock the cryptomalware increases every few hours or days
    - New variants of encrypt all files on any network or attached device connected to that computer

- Cost to unblock: Small enough that general user pays
- Number game

Launch

- **Primary Action- Launch:**
  - Malware that infects a computer to launch attacks on other computers
  - Examples: A virus, worm, and bot

- **Virus**
  - Replicates itself by modifying other computer programs and inserting its own code - Needs a host program
    - First unloads a payload to perform a malicious action
    - Next, replicates itself by inserting its code into another file (on the same computer)
  - **Two types**: A file-based virus and a fileless virus
  - **A *file-based virus***
    - *A M*alicious code that is attached to a file that reproduces itself on the same computer without any human intervention
    - **An *armored file-based virus***
      - *G*oes to great lengths to avoid detection
      - Techniques include *split infection* and *mutation*

Viruses

Appender infection

- Virus place their code at the end of the files they infect, adjusting the file's entry point to cause their code to be executed before that of the original file.
  - The virus appends itself to end of a file
  - Moves first three bytes of original file to virus code
  - Replaces them with a jump instruction pointing to the virus code
  - Easily detected by virus scanners

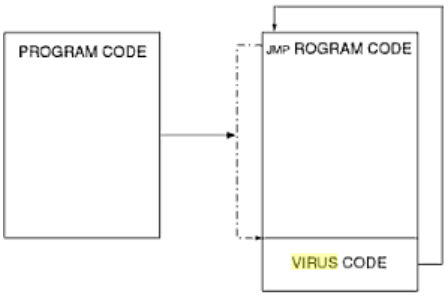


Figure 2-1 Appender infection

7

Viruses

Swiss cheese infection

- Viruses inject themselves into executable code instead of at the end
- Overwritten original code is transferred and stored inside virus code
- Host code executes properly after the infection
- Virus code is “scrambled” to make it more difficult to detect

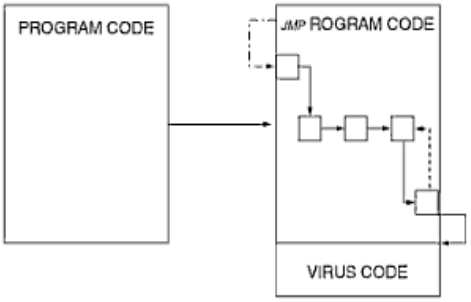


Figure 2-2 Swiss cheese infection

8

8

Viruses

Split infection

- Virus splits into several parts
- Parts placed at random positions in host program
- Head of virus code starts at beginning of file
- Gives control to next piece of virus code
- The parts may contain unnecessary "garbage" to mask their true purpose

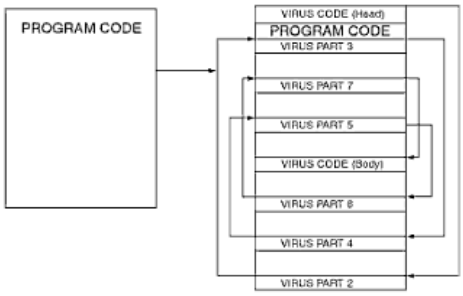


Figure 2-3 Split infection

9

Launch

- **Primary Action- Launch:**
  - **Fileless virus**
    - Does not attach itself to a file
    - Takes advantage of native services and processes that are part of the OS to avoid detection and carry out its attacks
    - The code is loaded directly in the computer's random access memory (RAM)
  - **Advantages of a fileless virus over a file-based virus:**
    - *Easy to infect*
    - *Extensive control*
    - *Persistent*
    - *Difficult to detect*
    - *Difficult to defend against*

Viruses

- **Mutation** – Some viruses can mutate or change
  - **Oligomorphic virus**
    - Changes its internal code to one of a set of number of predefined mutations whenever executed
  - **Polymorphic virus**
    - Completely changes from its original form when executed
  - **Metamorphic virus**
    - Rewrite its own code and appear different each time it is executed

11 11

Oligomorphic Malware	• Oligo- few or little • Morph - Transform / mutate adopt	• Changes its internal code to a predefined mutation whenever executed • Has limited number of mutations, so changes back to its original state
Polymorphic malware	Poly - Many	• Completely changes from its original form whenever it is executed • Usually accompanied by the malware containing scrambled code, that is unscrambled when the malware is activated
Metamorphic malware	Meta: indicate a concept which is an abstraction from another concept,	• Can rewrite its own code and thus appears different each time it is executed. • Does this by creating a logical equivalent of the code whenever it is run

12 12

Viruses

- **Viruses perform two actions:**
  - Unloads a payload to perform a malicious action
  - Reproduces itself by inserting its code into another file on the same computer
- **Impact**
  - Cause a computer to repeatedly crash
  - Erase files from or reformat hard drive
  - Turn off computer's security settings

Launch

- **Worm**
  - A malicious program that uses a computer network to replicate (sometimes called a network virus)
  - Designed to enter a computer through the network and then take advantage of a vulnerability in an application or an OS on the host computer
  - Searches for other computers with same vulnerabilities
  - Sends copies of itself to other network devices
  - Today's worms can leave behind a payload on the systems they infect and cause harm, much like a virus
- Worms may:
  - Consume resources **or**
  - Leave behind a payload to harm infected systems
- Examples of worm actions
  - Deleting computer files
  - Allowing remote control of a computer by an attacker

Action	Virus	Worm
What does it do?	Inserts malicious code into a program or data file	Exploits a vulnerability in an application or operating system
How does it spread to other computers?	User transfers infected files to other devices	Uses a network to travel from one computer to another
Does it infect a file?	Yes	No
Does there need to be user action for it to spread?	Yes	No

- Virus self replicate on the local computer, worms self replicate between computers
  - If a virus infects computer A, there will be multiple files on computer A that are infected, but computer B, C might not be infected
  - If a worm infects computer A, there will be single infection on computer A, but computer B, C may also be infected

15 15

Launch

- **Bot**
  - Allows the infected computer to be placed under the remote control of an attacker for the purpose of launching attacks
  - The infected robot computer is known as a **bot** or *zombie*
  - When hundreds, thousands, or even millions of bot computers are gathered into a logical computer network, they create a ***botnet*** under the control of a ***bot herder***
  - Infected bot computers receive instructions through a **command and control** (C&C) structure from the bot herders

16 16

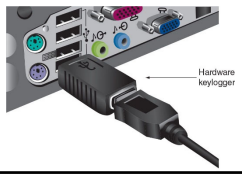
https://searchsecurity.techtarget.com/definition/worm

17

Snoop

- **Primary Action- Snoop**
  - Types: Spyware and keyloggers
- **Spyware**
  - A tracking software that is deployed without the consent or control of the user
- **Keylogger**
  - Silently captures and stores each keystroke that a user types on the computer's keyboard
  - The threat actor can then search the captured text for any useful information such as passwords, credit card numbers, or personal information

A keylogger can be a software program or a small hardware device



18

Table 3-4 Technologies Used by Spyware

Technology	Description	Impact
Automatic download software	Downloads and installs software without the user's interaction	Could install unauthorized applications
Passive tracking technologies	Gathers information about user activities without installing any software	Could collect private information such as websites a user has visited
System modifying software	Modifies or changes user configurations, such as the web browser home page or search page, default media player, or lower-level system functions	Changes configurations to settings that the user did not approve
Tracking software	Monitors user behavior or gathers information about the user, sometimes including personally identifiable or other sensitive information	Could collect personal information that can be shared widely or stolen, resulting in fraud or identity theft

19

Deceive

- **Primary Action- Deceive**
  - Attempts to deceive the user and hide its true intentions
  - Examples include potentially unwanted programs (PUPs), Trojans, and remote access Trojans (RATs)

- **Potentially Unwanted Program (PUP)**
  - Software that the user does not want on their computer
  - Examples of PUPs:
    - Advertising that obstructs content or interferes with web browsing
    - Pop-up windows
    - Pop-under windows
    - Browser (Search Engine) hijacking
    - Home page hijacking, etc

20

Deceive

- **Primary Action- Deceive**
  - **Trojan**
    - An executable program that masquerades as performing a benign activity
    - Also does something malicious
  - **Remote Access Trojan (RAT)**
    - Has the basic functionality of a Trojan
    - Also gives the threat agent unauthorized remote access to the victim's computer by using specially configured communication protocols
    - Creates an opening to the victim's computer allowing the threat agent unrestricted access

21

Evade

- **Primary Action- Evade**
  - Attempts to help malware or attacks evade detection
  - Includes backdoor, logic bomb, and rootkit
- **Backdoor**
  - Gives access to a computer, program, or service that circumvents any normal security protections
- **Logic bomb**
  - Computer code that is typically added to a legitimate program but lies dormant and evades detection until a specific logical event triggers it
- **Rootkits**
  - A malware that hide its presence and the presence of other malware on the computer
    - It does this by accessing "lower layers" of the OS to make alterations

22

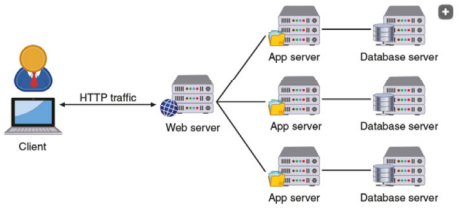
Application Attacks

- **Application Attacks**
  - Look for vulnerabilities in applications or manipulate applications in order to compromise them
  - Common targets of attackers using application attacks are Internet web server

- **Web server**
  - Provides services that are implemented as “web applications” through software applications running on the server

Figure 3-8

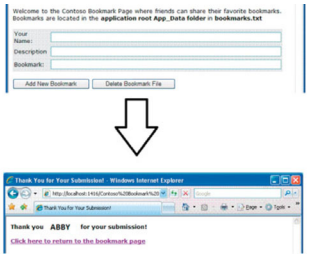
Web server application infrastructure



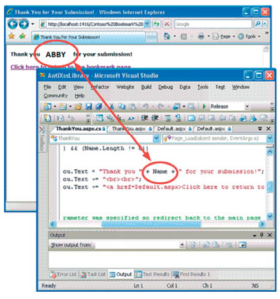
23

Scripting

- **Cross-site scripting** (**XSS**) attack
  - A website that accepts user input without validating it and uses that input in a response can be exploited
  - An attacker can take advantage in an XSS attack by tricking a valid website into feeding a malicious script to another user’s web browser



Input used in response



24

Injection

- Attacks called **injections** introduce new input to exploit a vulnerability

- **SQL injection**
  - Inserts statements to manipulate a database server
  - Targets SQL servers by introducing malicious commands into them
    - Crafted SQL statements as user input, information from the database can be extracted or the existing can be manipulated

```
SELECT fieldlist FROM table WHERE field = 'whatever' or 'a'='a'
```

25

Request Forgery

- **Request forgery**
  - *A* request that has been fabricated
  - **Types:** Cross-site request forgery (CSFR) and a server-site request forgery (SSRF)

- **Cross-Site Request Forgery (CSRF)**
  - Takes advantage of an authentication "token" that a website sends to a user's web browser
  - If a user is currently authenticated on a website and is then tricked into loading another webpage, the new page inherits the identity and privileges of the victim
    - May then perform an undesired function on the attacker's behalf

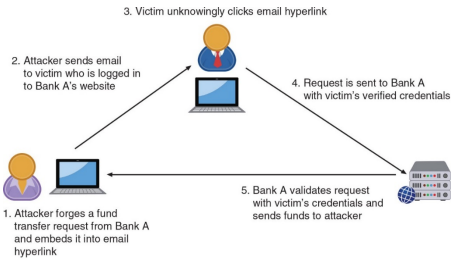


Figure 3-11 Cross-site request forgery

26

Request Forgery

- **Server-Site Request Forgery (SSRF)**
  - An SSRF takes advantage of a trusting relationship between web servers
  - SSRF attacks exploit how a web server processes external information received from another server
  - Some web applications are designed to read information from or write information to a specific URL
  - If an attacker can modify that target URL, they can potentially extract sensitive information from the application or inject untrusted input into it

Attack Name	Attack Target	Purpose of Attack
CSRF	User	Force target to take action for attacker while pretending to be authorized user
SSRF	Web server	Gain access to sensitive data or inject harmful data

27

Replay

- Replay attacks are commonly used against digital identities
  - After intercepting and copying data, the threat actor retransmits selected and edited portions of the copied communications later to impersonate the legitimate user
- Many digital identity replay attacks are between a user and an authentication server

28

Attacks on Software

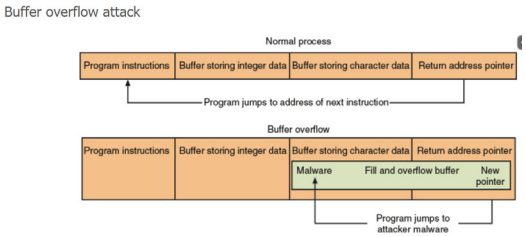
- **Attacks on Software**
  - Directly focused on vulnerabilities in the software applications
- Include:
  - Exploiting memory vulnerabilities
  - Improper exception and error handling
  - External software components

- **Memory Vulnerabilities**
  - **Resource exhaustion** attacks
    - "Deplete" parts of memory and thus interfere with the normal operation of the program in RAM
    - An example : M**emory leak**.
      - An application normally dynamically allocates memory, but due to a programming error, it may not free that memory when finished using it.
      - An attacker can then take advantage of the unexpected program behavior resulting from a low memory condition.

29

Attacks on Software

- **Memory Vulnerabilities (continued)**
  - **Buffer overflow attack**
    - A process attempts to store data in RAM beyond the boundaries of a fixed-length storage buffer
    - This extra data overflows into the adjacent memory locations
    - Because the storage buffer typically contains the "return address" memory location of the software program being executed when another function interrupted the process, an attacker can overflow the buffer with a new address pointing to the attacker's malware code.



30

- **Memory Vulnerabilities (continued)**
  - **Integer overflow attack**
    - An attacker changes the value of a variable to something outside the range that the programmer had intended by using an integer overflow
    - When this integer overflow occurs, the interpreted value then wraps around from the maximum value to the minimum value.
    - For example, an eight-bit signed integer has a maximum value of 127 and a minimum value of −128.
    - If the value 127 is stored in a variable and 1 is added to it, the sum exceeds the maximum value for this integer type and wraps around to become −128.

- **Memory Vulnerabilities (continued)**
  - **Improper Exception Handling**
    - Some attacks are the result of poor coding on the part of software developers
    - Software that allows the user to enter data but has **improper input handling** features does not filter or validate user input to prevent a malicious action

- **Incorrect error handling**:
  - An attacker enters a string of characters that is much longer than expected.
  - Because the software has not been designed for this event, the program could crash or suddenly halt its execution and then display an underlying OS prompt, giving an attacker access to the computer.

Attacks on Software (3 of 3)

- **Attacks on External Software Components**
  - In addition to attacking the software directly, threat actors also target external software components
  - These include the following:
    - *Application program interface (API)*
    - *Device driver*
    - *Dynamic-link library (DLL)*

  - **Application program interface (API)**
    - A link provided by an OS, web browser, or other platform that allows a developer access to resources at a high level.
    - APIs relieve the developer from the need to write code for specific hardware and software.
    - Provide direct access to data and an entry point to an application's functions, they are attractive targets for attackers looking for vulnerabilities in the API in an **application program interface (API) attack**.

33

- ***Device driver***
  - A software that controls and operates an external hardware device that is connected to a computer.
  - Apecific to both the OS and the hardware device.
  - Threat actors may attempt to alter a device driver for use in an attack (called **device driver manipulation**).
  - **Shimming**
    - Attacker add a small coding library that intercepts calls made by the device and changes the parameters passed between the device and the device driver.
    - This **refactoring** (changing the design of existing code) can be difficult to detect yet serves as a real threat.

34

34

Adversarial Artificial Intelligence Attacks

- Cybersecurity is using artificial intelligence to enhance the detection of malicious behavior and advanced threats
  - However, there are significant vulnerabilities and risks with using these new tools
- Understanding them includes:
  - Knowing what the tools are and what they can do
  - How these tools are used in cybersecurity
  - Knowing their potential risks

35

What Are Artificial Intelligence (AI) and Machine Learning (ML)?

- ***AI***
  - Technology that imitates human abilities
- ***Machine learning* (*ML*)**
  - A recognized subset of AI
  - Defined as "teaching" a technology device to "learn" by itself without the continual instructions of a computer programmer
  - Also involves learning through repeated experience
    - If something attempted does not work, then it determines how it could be changed to make it work

36

Uses in Cybersecurity

- Cybersecurity AI allows organizations to detect, predict, and respond to cyberthreats in real time using ML
- Virtually all email systems use some type of AI to block phishing attacks
- The prime advantages of using AI to combat threats are continual learning and greater speed in response
  - AI can predict and prevent future attacks

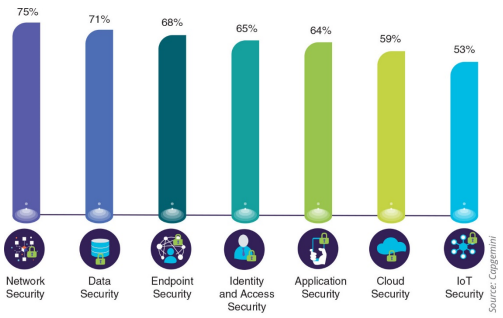


Figure 3-13 How AI cybersecurity is used

37

Risks in Using AI and ML in Cybersecurity

- **Adversarial artificial intelligence**
  - Risks associated with using AI and ML
- **Risk 1: Security of ML algorithms**
  - These could be attacked and compromised, allowing threat actors to alter algorithms to ignore attacks
- **Risk 2: tainted training data for machine learning**
  - Attackers can attempt to alter the training data that is used by ML
  - **Goal:** Produce false negatives to cloak themselves

38

Summary

- The word “endpoint” is commonly used when referring to network-connected hardware devices
- Malware is software that enters a computer system without the user’s knowledge or consent and then performs an unwanted and harmful action
  - Attempt to take away the freedom of users
  - Infects a computer to then launch attacks on other computers
  - “Snoops” or spies on its victims
  - Deceive the user and hide its true intentions
  - Attempts to evade detection
- Another category of attacks specifically targets software applications that are already installed and running on the device
- Artificial intelligence (AI) is technology that imitates human abilities
- Adversarial artificial intelligence