

Computer Security: Principles and Practice

Fourth Edition

By: William Stallings and Lawrie Brown

Chapter 6

Malicious Software

Malware

NIST 800-83 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.”

Classification of Malware

Classified into two broad categories:

Based first on how it spreads or propagates to reach the desired targets

Then on the actions or payloads it performs once a target is reached

Also classified by:

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)

Types of Malicious Software (Malware)

Propagation mechanisms include:

- **Infection of existing content by viruses that is subsequently spread to other systems**
- **Exploit of software vulnerabilities by worms or drive-by-downloads to allow the malware to replicate**
- **Social engineering attacks that convince users to bypass security mechanisms to install Trojans or to respond to phishing attacks**



Payload actions performed by malware once it reaches a target system can include:

- **Corruption of system or data files**
- **Theft of service/make the system a zombie agent of attack as part of a botnet**
- **Theft of information from the system/keylogging**
- **Stealth/hiding its presence on the system**

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

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

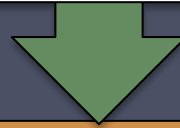
Virus Phases

Dormant phase

Virus is idle

Will eventually be activated by some event

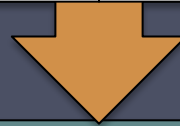
Not all viruses have this stage



Triggering phase

Virus is activated to perform the function for which it was intended

Can be caused by a variety of system events

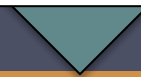


Propagation phase

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



Execution phase

Function is performed

May be harmless or damaging

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

Worm Replication

Electronic mail or instant messenger facility

File sharing

Remote execution capability

Remote file access or transfer capability

Remote login capability

Recent Worm Attacks

Melissa	1998	E-mail worm First to include virus, worm and Trojan in one package
Code Red	July 2001	Exploited Microsoft IIS bug Probes random IP addresses Consumes significant Internet capacity when active
Code Red II	August 2001	Also targeted Microsoft IIS Installs a backdoor for access
Nimda	September 2001	Had worm, virus and mobile code characteristics Spread using e-mail, Windows shares, Web servers, Web clients, backdoors
SQL Slammer	Early 2003	Exploited a buffer overflow vulnerability in SQL server compact and spread rapidly
Sobig.F	Late 2003	Exploited open proxy servers to turn infected machines into spam engines
Mydoom	2004	Mass-mailing e-mail worm Installed a backdoor in infected machines
Warezov	2006	Creates executables in system directories Sends itself as an e-mail attachment Can disable security related products
Conficker (Downadup)	November 2008	Exploits a Windows buffer overflow vulnerability Most widespread infection since SQL Slammer
Stuxnet	2010	Restricted rate of spread to reduce chance of detection Targeted industrial control systems

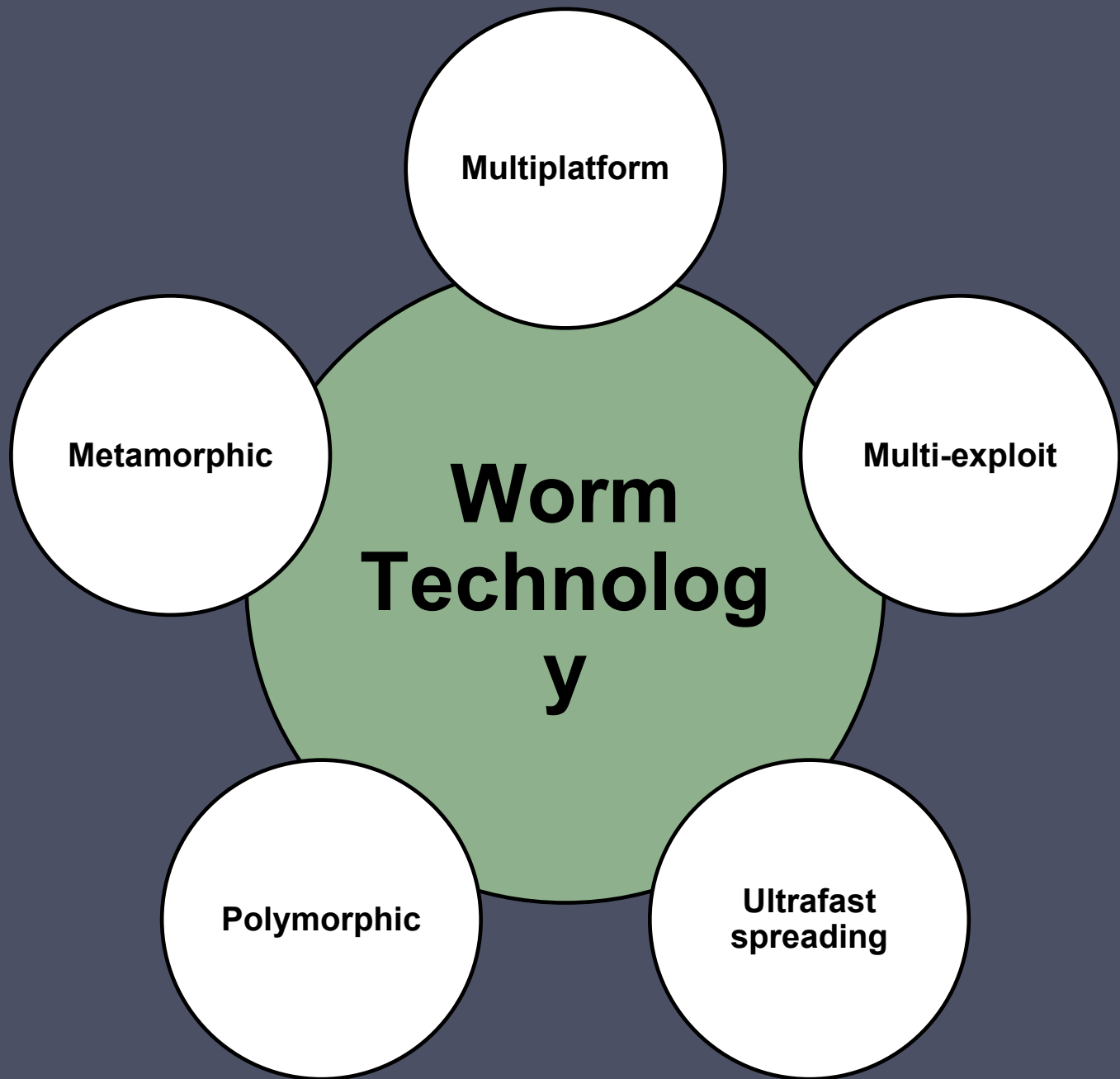
WannaCry

Ransomware attack in May 2017 that spread extremely fast over a period of hours to days, infecting hundreds of thousands of systems belonging to both public and private organizations in more than 150 countries

It spread as a worm by aggressively scanning both local and random remote networks, attempting to exploit a vulnerability in the Server Message Block protocol file sharing service on unpatched Windows systems

This rapid spread was only slowed by the accidental activation of a “kill-switch” domain by a UK security researcher

Once installed on infected systems, it also encrypted files, demanding a ransom payment to recover them



Mobile Code

- NIST SP 800-28 defines mobile code as
 - “programs that can be shipped unchanged to a heterogeneous collection of platforms and executed with identical semantics”
- Transmitted from a remote system to a local system and then executed on the local system
- Often acts as a mechanism for a virus, worm, or Trojan horse
- Takes advantage of vulnerabilities to perform its own exploits
- Popular vehicles include:
 - Java applets
 - ActiveX
 - JavaScript
 - VBScript
- Most common ways of using mobile code for malicious operations on local system are:
 - Cross-site scripting
 - Interactive and dynamic Web sites

Mobile Phone Worms

- First discovery was Cabir worm in 2004
- Then 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

Drive-By-Downloads

Exploits browser and plugin vulnerabilities so when the user views a webpage controlled by the attacker, it contains code that exploits the bug to download and install malware on the system without the user's knowledge or consent

In most cases the malware does not actively propagate as a worm does

Spreads when users visit the malicious Web page

Watering-Hole Attacks

- A variant of drive-by-download used in highly targeted attacks
- The attacker researches their intended victims to identify websites they are likely to visit, then scans these sites to identify those with vulnerabilities that allow their compromise
- They then wait for one of their intended victims to visit one of the compromised sites
- Attack code may even be written so that it will only infect systems belonging to the target organization and take no action for other visitors to the site
- This greatly increases the likelihood of the site compromise remaining undetected

Malvertising

Places malware on websites without actually compromising them

The attacker pays for advertisements that are highly likely to be placed on their intended target websites and incorporate malware in them

Using these malicious ads, attackers can infect visitors to sites displaying them

The malware code may be dynamically generated to either reduce the chance of detection or to only infect specific systems

Has grown rapidly in recent years because they are easy to place on desired websites with few questions asked and are hard to track

Attackers can place these ads for as little as a few hours, when they expect their intended victims could be browsing the targeted websites, greatly reducing their visibility

Clickjacking

- Also known as a user-interface (UI) redress attack
- Using a similar technique, keystrokes can also be hijacked
 - A user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker
- Vulnerability used by an attacker to collect an infected user's clicks
 - The attacker can force the user to do a variety of things from adjusting the user's computer settings to unwittingly sending the user to Web sites that might have malicious code
 - By taking advantage of Adobe Flash or JavaScript an attacker could even place a button under or over a legitimate button making it difficult for users to detect
 - A typical attack uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the top level page
 - The attacker is hijacking clicks meant for one page and routing them to another page

🔖 Chapter 6 Malicious Software

- 🔖 6.1 Types of Malicious Software (Malware)
- 🔖 6.2 Advanced Persistent Threat
- 🔖 6.3 Propagation—Infected Content—Viruses
- 🔖 6.4 Propagation—Vulnerability Exploit—Worms
- 🔖 6.5 Propagation—Social Engineering—Spam E-mail, Trojans
- 🔖 6.6 Payload—System Corruption
- 🔖 6.7 Payload—Attack Agent—Zombie, Bots
- 🔖 6.8 Payload—Information Theft—Keyloggers, Phishing, Spyware
- 🔖 6.9 Payload—Stealth—Backdoors, Rootkits
- 🔖 6.10 Countermeasures

Social Engineering

- “Tricking” users to assist in the compromise of their own systems

Unsolicited bulk
e-mail

Significant carrier of
malware

Used for phishing
attacks

Program or utility
containing harmful
hidden code

Used to accomplish
functions that the
attacker could not
accomplish directly

First appeared in
2004 (Skuller)

Target is the
smartphone

Payload

System Corruption

Chernobyl virus

- First seen in 1998
- Example of a destructive parasitic memory-resident Windows 95 and 98 virus
- Infects executable files when they are opened and when a trigger date is reached, the virus deletes data on the infected system by overwriting the first megabyte of the hard drive with zeroes, resulting in massive corruption of the entire file system

Klez

- Mass mailing worm infecting Windows 95 to XP systems
- First seen in October 2001
- Spreads by e-mailing copies of itself to addresses found in the address book and in files on the system
- It can stop and delete some anti-virus programs running on the system
- On trigger date causes files on the hard drive to become empty

Ransomware

- Encrypts the user's data and demands payment in order to access the key needed to recover the information
- PC Cyborg Trojan (1989)
- Mid-2006 a number of worms and Trojans appeared that used public-key cryptography with increasingly larger key sizes to encrypt data
- The user needed to pay a ransom, or to make a purchase from certain sites, in order to receive the key to decrypt this data

Ransomware

- WannaCry
 - Infected a large number of systems in many countries in May 2017
 - When installed on infected systems, it encrypted a large number of files and then demanded a ransom payment in Bitcoins to recover them
 - Recovery of this information was generally only possible if the organization had good backups and an appropriate incident response and disaster recovery plan
 - Targets widened beyond personal computer systems to include mobile devices and Linux servers
 - Tactics such as threatening to publish sensitive personal information, or to permanently destroy the encryption key after a short period of time, are sometimes used to increase the pressure on the victim to pay up

Payload

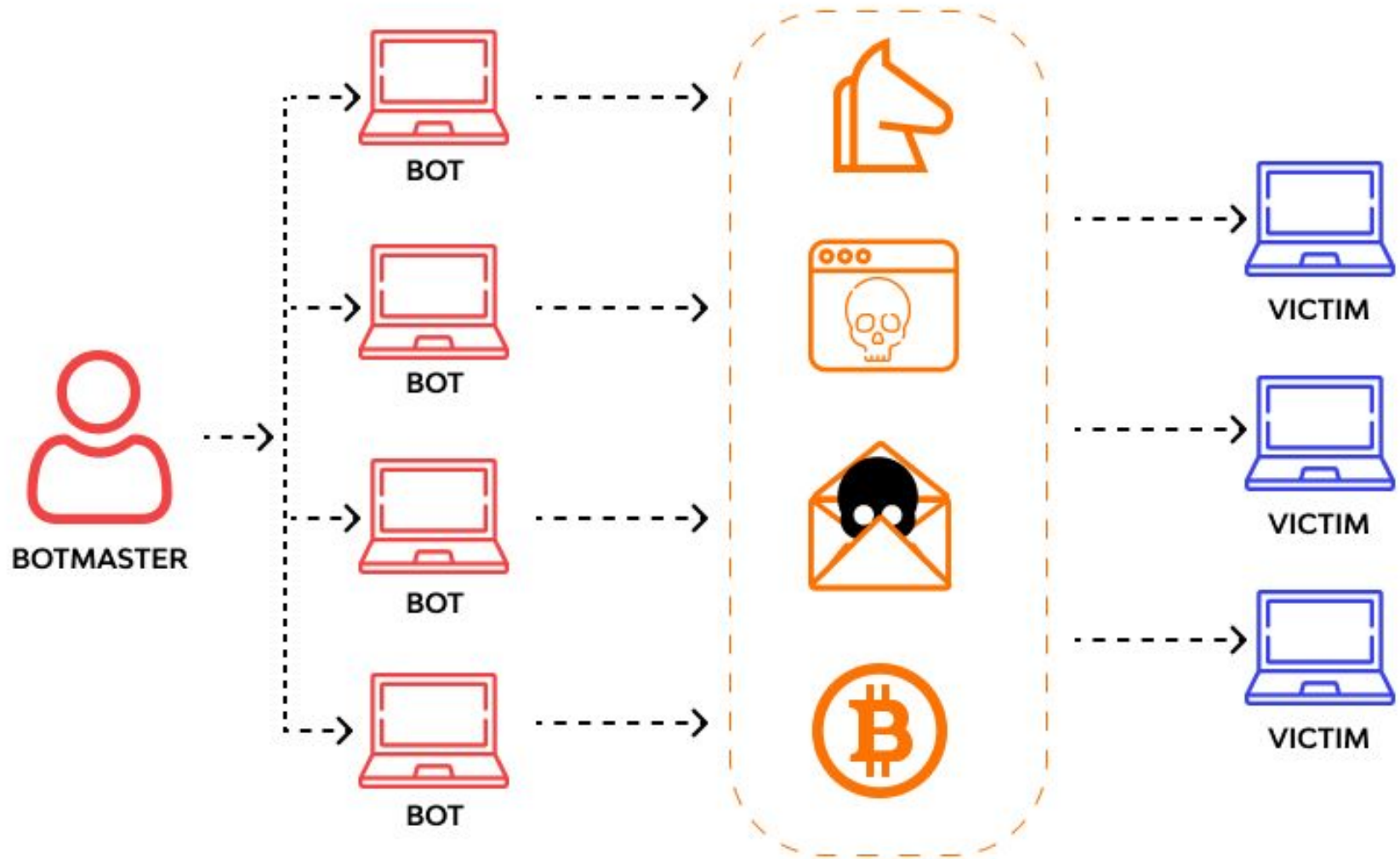
System Corruption

- Real-world damage
 - Causes damage to physical equipment
 - Chernobyl virus rewrites BIOS code
 - Stuxnet worm
 - Targets specific industrial control system software
 - There are concerns about using sophisticated targeted malware for industrial sabotage
- Logic bomb
 - Code embedded in the malware that is set to “explode” when certain conditions are met

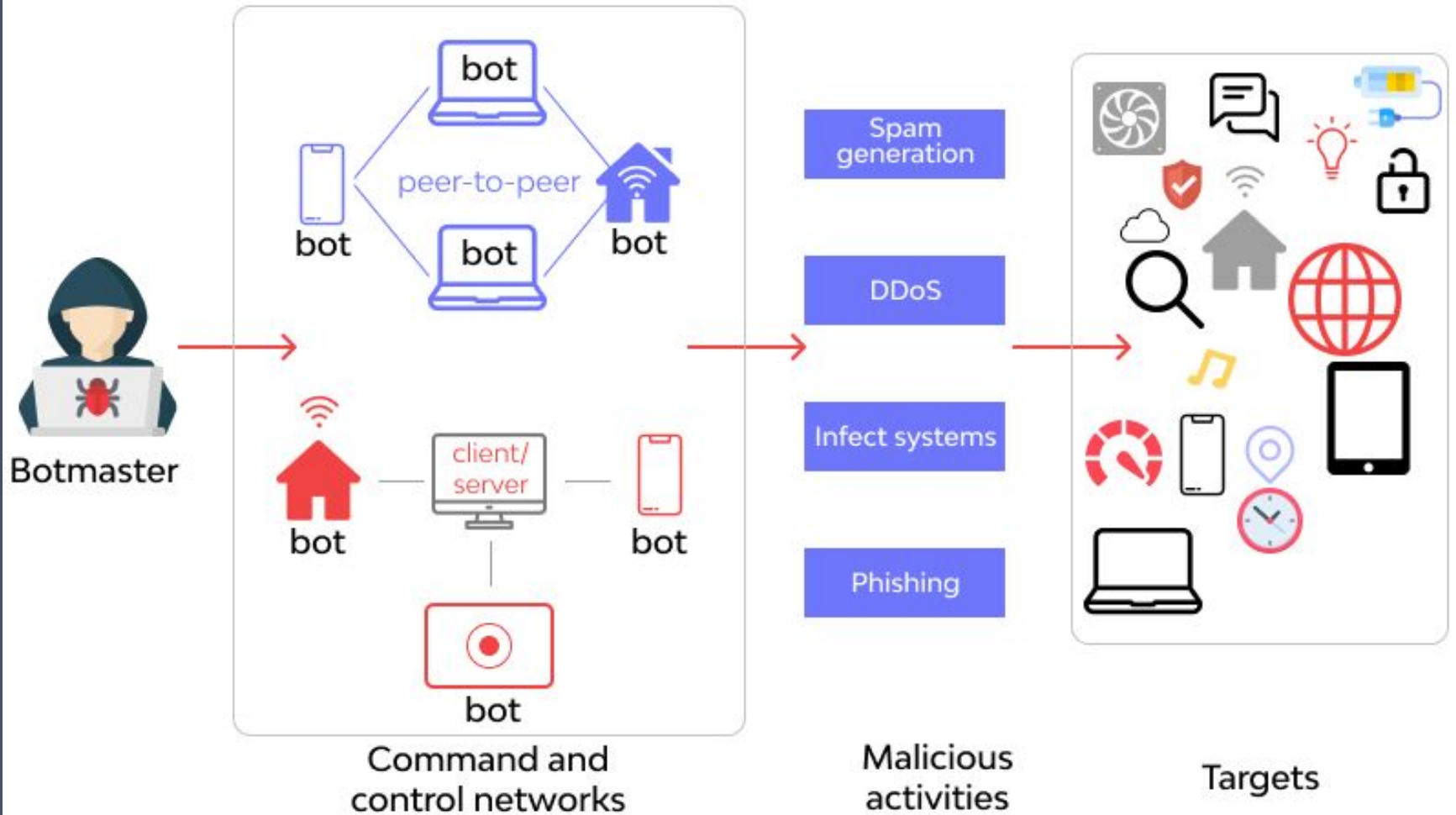
Payload – Attack Agents

Bots

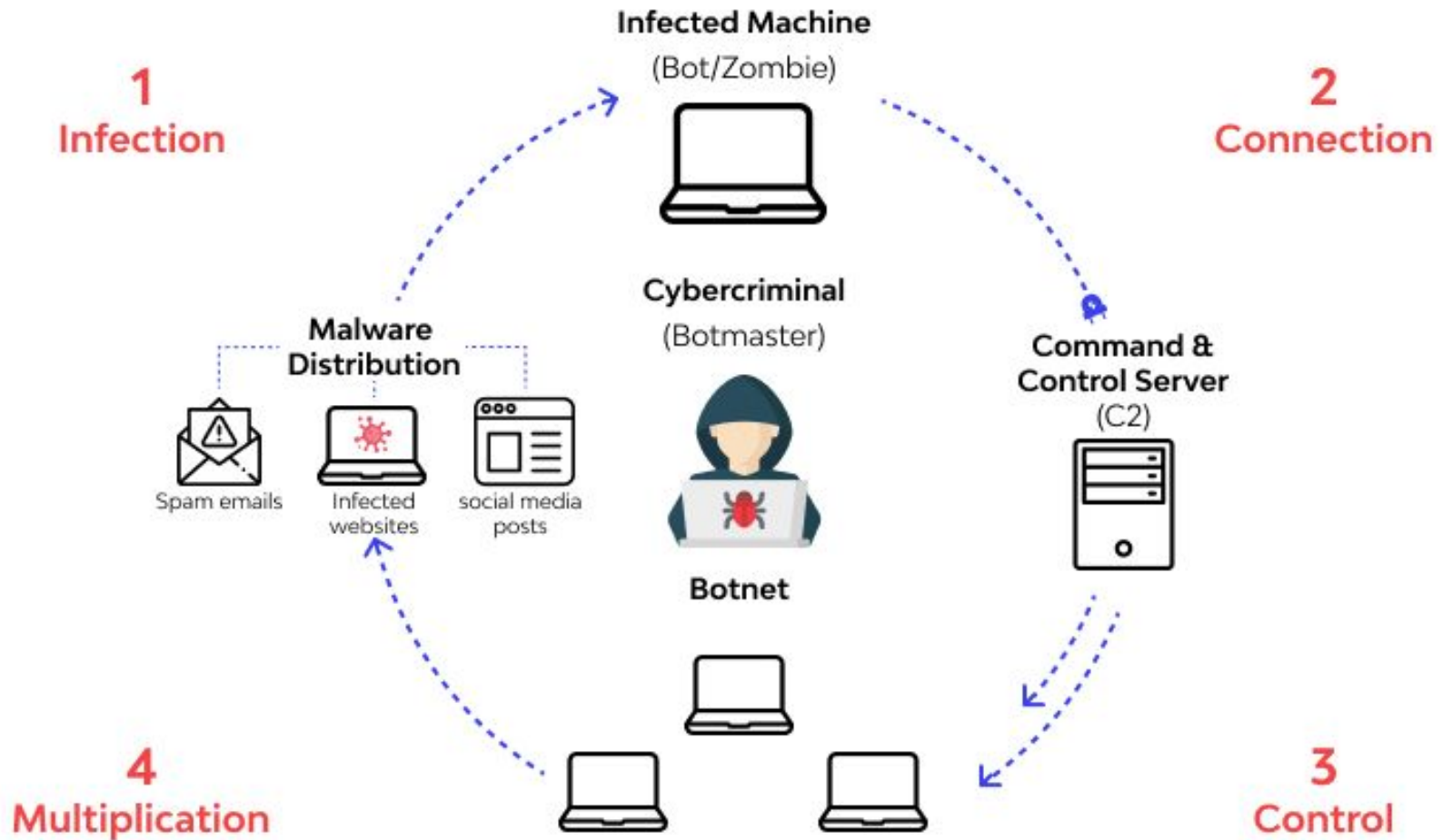
- Takes over another Internet attached computer and uses that computer to launch or manage attacks
- *Botnet* - collection of bots capable of acting in a coordinated manner
- Uses:
 - Distributed denial-of-service (DDoS) attacks
 - Spamming
 - Sniffing traffic
 - Keylogging
 - Spreading new malware
 - Installing advertisement add-ons and browser helper objects (BHOs)
 - Attacking IRC chat networks
 - Manipulating online polls/games



Botnet command and control architecture



How a Botnet works



Remote Control Facility

- Distinguishes a bot from a worm
 - Worm propagates itself and activates itself
 - Bot is initially controlled from some central facility
- Typical means of implementing the remote control facility is on an 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 such as HTTP
 - Distributed control mechanisms use peer-to-peer protocols to avoid a single point of failure

Payload – Information Theft

Keyloggers and Spyware

Keylogger

- Captures keystrokes to allow attacker to monitor sensitive information
- Typically uses some form of filtering mechanism that only returns information close to keywords (“login”, “password”)

Spyware

- Subverts the compromised machine to allow monitoring of a wide range of activity on the system
 - Monitoring history and content of browsing activity
 - Redirecting certain Web page requests to fake sites
 - Dynamically modifying data exchanged between the browser and certain Web sites of interest

Payload – Information Theft

Phishing

- Exploits social engineering to leverage the user's trust by masquerading as communication from a trusted source
 - Include a URL in a spam e-mail that links to a fake Web site that mimics the login page of a banking, gaming, or similar site
 - Suggests that urgent action is required by the user to authenticate their account
 - Attacker exploits the account using the captured credentials
- Spear-phishing
 - Recipients are carefully researched by the attacker
 - E-mail is crafted to specifically suit its recipient, often quoting a range of information to convince them of its authenticity

Payload – Stealthing

Backdoor

- Also known as a *trapdoor*
- Secret entry point into a program allowing the attacker to gain access and bypass the security access procedures
- *Maintenance hook* is a backdoor used by Programmers to debug and test programs
- Difficult to implement operating system controls for backdoors in applications

Payload - Stealthing Rootkit

- 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

Rootkit Classification

Characteristics

Persistent

**Memory
based**

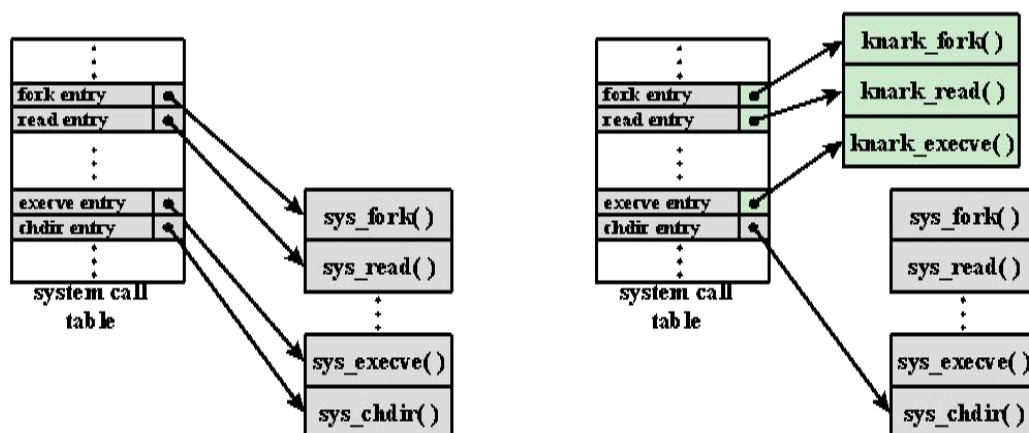
User mode

Kernel mode

**Virtual
machine
based**

**External
mode**

- **Persistent:** Activates each time the system boots. The rootkit must store code in a persistent store, such as the registry or file system, and configure a method by which the code executes without user intervention. This means it is easier to detect, as the copy in persistent storage can potentially be scanned.
- **Memory based:** Has no persistent code and therefore cannot survive a reboot. However, because it is only in memory, it can be harder to detect.
- **User mode:** Intercepts calls to APIs (application program interfaces) and modifies returned results. For example, when an application performs a directory listing, the return results do not include entries identifying the files associated with the rootkit.
- **Kernel mode:** Can intercept calls to native APIs in kernel mode.⁴ The rootkit can also hide the presence of a malware process by removing it from the kernel's list of active processes.
- **Virtual machine based:** This type of rootkit installs a lightweight virtual machine monitor, then runs the operating system in a virtual machine above it. The rootkit can then transparently intercept and modify states and events occurring in the virtualized system.
- **External mode:** The malware is located outside the normal operation mode of the targeted system, in BIOS or system management mode, where it can directly access hardware.



(a) Normal kernel memory layout

(b) After nkark install

Figure 6.3 System Call Table Modification by Rootkit

- **Modify the system call table:** The attacker modifies selected syscall addresses stored in the system call table. This enables the rootkit to direct a system call away from the legitimate routine to the rootkit's replacement. Figure 6.3 shows how the knark rootkit achieves this.
- **Modify system call table targets:** The attacker overwrites selected legitimate system call routines with malicious code. The system call table is not changed.
- **Redirect the system call table:** The attacker redirects references to the entire system call table to a new table in a new kernel memory location.

Malware Countermeasure Approaches

- Ideal solution to the threat of malware is prevention

Four main elements of prevention:

- Policy
- Awareness
- Vulnerability mitigation
- Threat mitigation

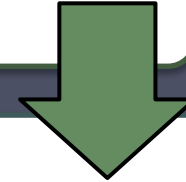
- If prevention fails, technical mechanisms can be used to support the following threat mitigation options:

- Detection
- Identification
- Removal

Generations of Anti-Virus Software

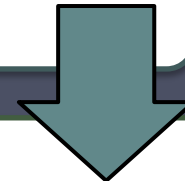
First generation: simple scanners

- Requires a malware signature to identify the malware
- Limited to the detection of known malware



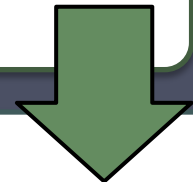
Second generation: heuristic scanners

- Uses heuristic rules to search for probable malware instances
- Another approach is integrity checking



Third generation: activity traps

- Memory-resident programs that identify malware by its actions rather than its structure in an infected program



Fourth generation: full-featured protection

- Packages consisting of a variety of anti-virus techniques used in conjunction
- Include scanning and activity trap components and access control capability

Sandbox Analysis

- Running potentially malicious code in an emulated sandbox or on a virtual machine
- Allows the code to execute in a controlled environment where its behavior can be closely monitored without threatening the security of a real system
- Running potentially malicious software in such environments enables the detection of complex encrypted, polymorphic, or metamorphic malware
- The most difficult design issue with sandbox analysis is to determine how long to run each interpretation

Host-Based Behavior-Blocking Software

- Integrates with the operating system of a host computer and monitors program behavior in real time for malicious action
 - Blocks potentially malicious actions before they have a chance to affect the system
 - Blocks software in real time so it has an advantage over anti-virus detection techniques such as fingerprinting or heuristics

Limitations

- Because malicious code must run on the target machine before all its behaviors can be identified, it can cause harm before it has been detected and blocked

Perimeter Scanning Approaches

- Anti-virus software typically included in e-mail and Web proxy services running on an organization's firewall and IDS
- May also be included in the traffic analysis component of an IDS
- May include intrusion prevention measures, blocking the flow of any suspicious traffic
- Approach is limited to scanning malware

Located at the border between the enterprise network and the Internet

One technique is to look for incoming traffic to unused local IP addresses

Located at the egress point of individual LANs as well as at the border between the enterprise network and the Internet

Monitors outgoing traffic for signs of scanning or other suspicious behavior

Two types of monitoring software

Summary

- Types of malicious software (malware)
 - Broad classification of malware
 - Attack kits
 - Attack sources
- Advanced persistent threat
- Propagation-vulnerability exploit-worms
 - Target discovery
 - Worm propagation model
 - The Morris Worm
 - Brief history of worm attacks
 - State of worm technology
 - Mobile code
 - Mobile phone worms
 - Client-side vulnerabilities
 - Drive-by-downloads
 - Clickjacking
- Payload-stealth-backdoors, rootkits
 - Backdoor
- Propagation-social engineering-span E-mail, Trojans
 - Spam E-mail
 - Trojan horses
 - Mobile phone Trojans
- Payload-system corruption
 - Data destruction
 - Real-world damage
 - Logic bomb
- Payload-attack agent-zombie, bots
 - Uses of bots
 - Remote control facility
- Payload-information theft-keyloggers, phishing, spyware
 - Credential theft, keyloggers, and spyware
 - Phishing and identity theft
 - Reconnaissance, espionage, and data exfiltration
- Countermeasures
 - Malware countermeasure approaches
 - Host-based scanners
 - Signature-based anti-virus
 - Perimeter scanning approaches