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."

Table 6.1
Malware Terminology (1/4)

Name	Description
Advanced persistent threat	Cybercrime directed at business and political targets, using a wide variety of intrusion technologies and malware, applied persistently and effectively to specific targets over an extended period, often attributed to state-sponsored organizations.
Adware	Advertising that is integrated into software. It can result in pop-up ads or redirection of a browser to a commercial site.
Attack Kit	Set of tools for generating new malware automatically using a variety of supplied propagation and payload mechanisms.
Auto-rooter	Malicious hacker tools used to break into new machines remotely.
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.

Table 6.1
Malware Terminology (2/5)

Name	Description
Downloaders	Code that installs other items on a machine that is under attack. It is normally included in the malware code first inserted on to a compromised system to then import a larger malware package.
Drive-by download	An attack using code in a compromised web site that exploits a browser vulnerability to attack a client system when the site is viewed.
Exploits	Code specific to a single vulnerability or set of vulnerabilities.
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.

Table 6.1
Malware Terminology (1/3)

(e.g., Adobe® Flash®, Shockwave®, Java applets, VBScripts, ActiveX).

Name	Description
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.

Table 6.1
Malware Terminology (1/3)

Name	Description
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.

6.1.1 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**
- **Stealthing/hiding its presence on the system**

6.1.2 Attack Kits

- Initially the development and deployment of malware required considerable technical skill by software authors
  - The development of virus-creation toolkits in the early 1990s and then more general attack kits in the 2000s greatly assisted in the development and deployment of malware
- Toolkits are often known as "crimeware"
  - Include a variety of propagation mechanisms and payload modules that even novices can deploy
  - Variants that can be generated by attackers using these toolkits creates a significant problem for those defending systems against them
- Examples are:
  - Zeus
  - Angler

6.1.3 Attack Sources

- Another significant malware development is the change from attackers being individuals often motivated to demonstrate their technical competence to their peers to more organized and dangerous attack sources such as:

- This has significantly changed the resources available and motivation behind the rise of malware and has led to development of a large underground economy involving the sale of attack kits, access to compromised hosts, and to stolen information

6.2 Advanced Persistent Threats (APTs)

- Well-resourced, persistent application of a wide variety of intrusion technologies and malware to selected targets (usually business or political)
- Typically attributed to state-sponsored organizations and criminal enterprises
- Differ from other types of attack by their careful target selection and stealthy intrusion efforts over extended periods
- High profile attacks include Aurora, RSA, APT1, and Stuxnet

APT Characteristics

Advanced

- Used by the attackers of a wide variety of intrusion technologies and malware including the development of custom malware if required
- The individual components may not necessarily be technically advanced but are carefully selected to suit the chosen target

Persistent

- Determined application of the attacks over an extended period against the chosen target in order to maximize the chance of success
- A variety of attacks may be progressively applied until the target is compromised

Threats

- Threats to the selected targets as a result of the organized, capable, and well-funded attackers intent to compromise the specifically chosen targets
- The active involvement of people in the process greatly raises the threat level from that due to automated attacks tools, and also the likelihood of successful attacks

APT Attacks

- Aim:
  - Varies from theft of intellectual property or security and infrastructure related data to the physical disruption of infrastructure
- Techniques used:
  - Social engineering
  - Spear-phishing email
  - Drive-by-downloads from selected compromised websites likely to be visited by personnel in the target organization
- Intent:
  - To infect the target with sophisticated malware with multiple propagation mechanisms and payloads
  - Once they have gained initial access to systems in the target organization a further range of attack tools are used to maintain and extend their access

6.3 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

6.3.2 Macro and Scripting Viruses

- NISTIR 7298 defines a macro virus as:

  "a virus that attaches itself to documents and uses the macro programming capabilities of the document's application to execute and propagate"
- Macro viruses infect scripting code used to support active content in a variety of user document types
- Are threatening for a number of reasons:
  - Is platform independent
  - Infect documents, not executable portions of code
  - Are easily spread
  - Because they infect user documents rather than system programs, traditional file system access controls are of limited use in preventing their spread, since users are expected to modify them
  - Are much easier to write or to modify than traditional executable viruses

```
macro Document_Open
    disable Macro menu and some macro security features
    if called from a user document
        copy macro code into Normal template file
    else
        copy macro code into user document being opened
    end if
    if registry key "Melissa" not present
        if Outlook is email client
           for first 50 addresses in address book
                send email to that address
                with currently infected document attached
           end for
        end if
        create registry key "Melissa"
    end if
    if minute in hour equals day of month
        insert text into document being opened
    end if
end macro
```

Figure 6.1 Melissa Macro Virus Pseudocode

6.3.3 Virus Classifications

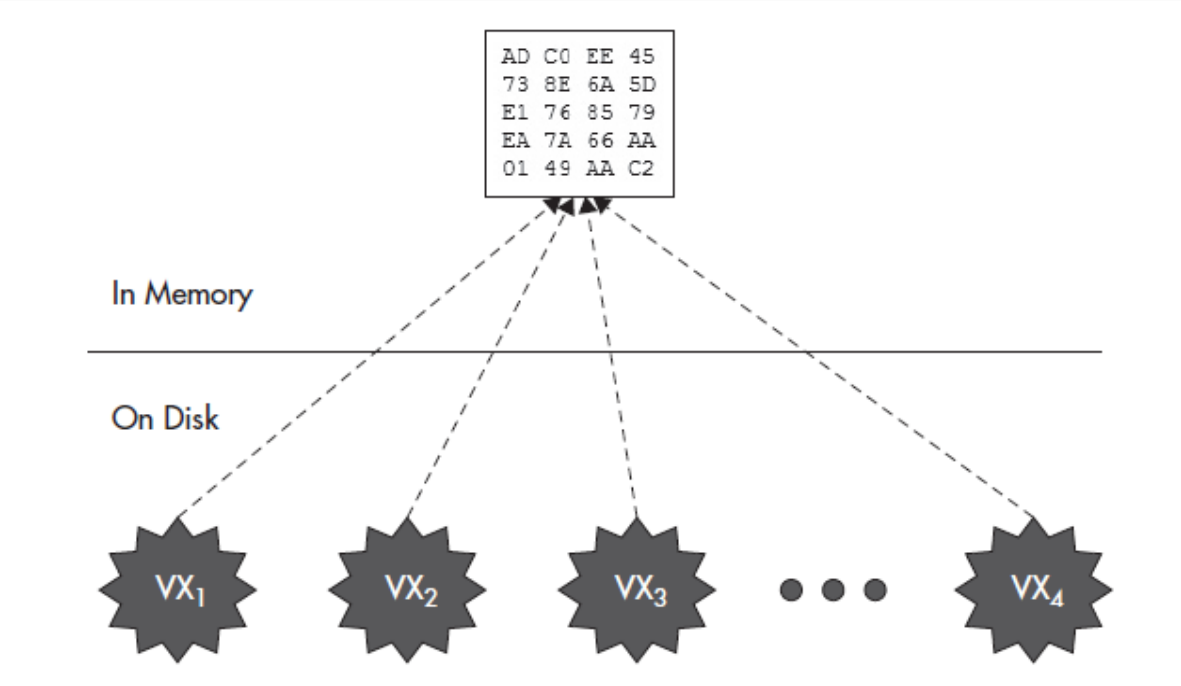
Classification by target

- Boot sector infector
  - Infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus
- File infector
  - Infects files that the operating system or shell considers to be executable
- Macro virus
  - Infects files with macro or scripting code that is interpreted by an application
- Multipartite virus
  - Infects files in multiple ways

Classification by concealment strategy

- Encrypted virus
  - A portion of the virus creates a random encryption key and encrypts the remainder of the virus
- Stealth virus
  - A form of virus explicitly designed to hide itself from detection by anti-virus software
- Polymorphic virus
  - A virus that mutates with every infection
- Metamorphic virus
  - A virus that mutates and rewrites itself completely at each iteration and may change behavior as well as appearance

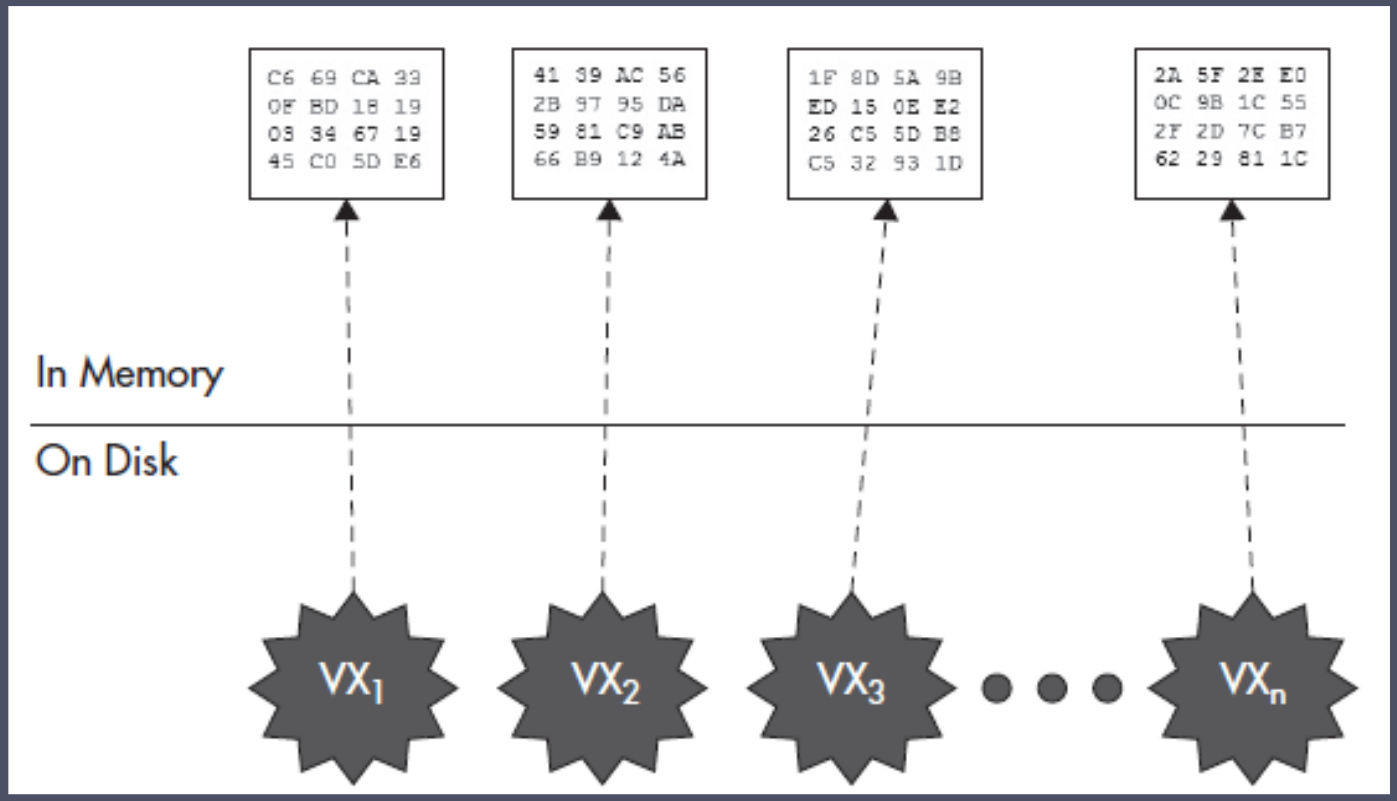
Polymorphic virus



- Polymorphic malware infections differ on disk but are the same in memory because the decrypted code in all infections is the same.

Malware, Rootkits & Botnets A Beginner's Guide.pdf, P-135 (162/385)

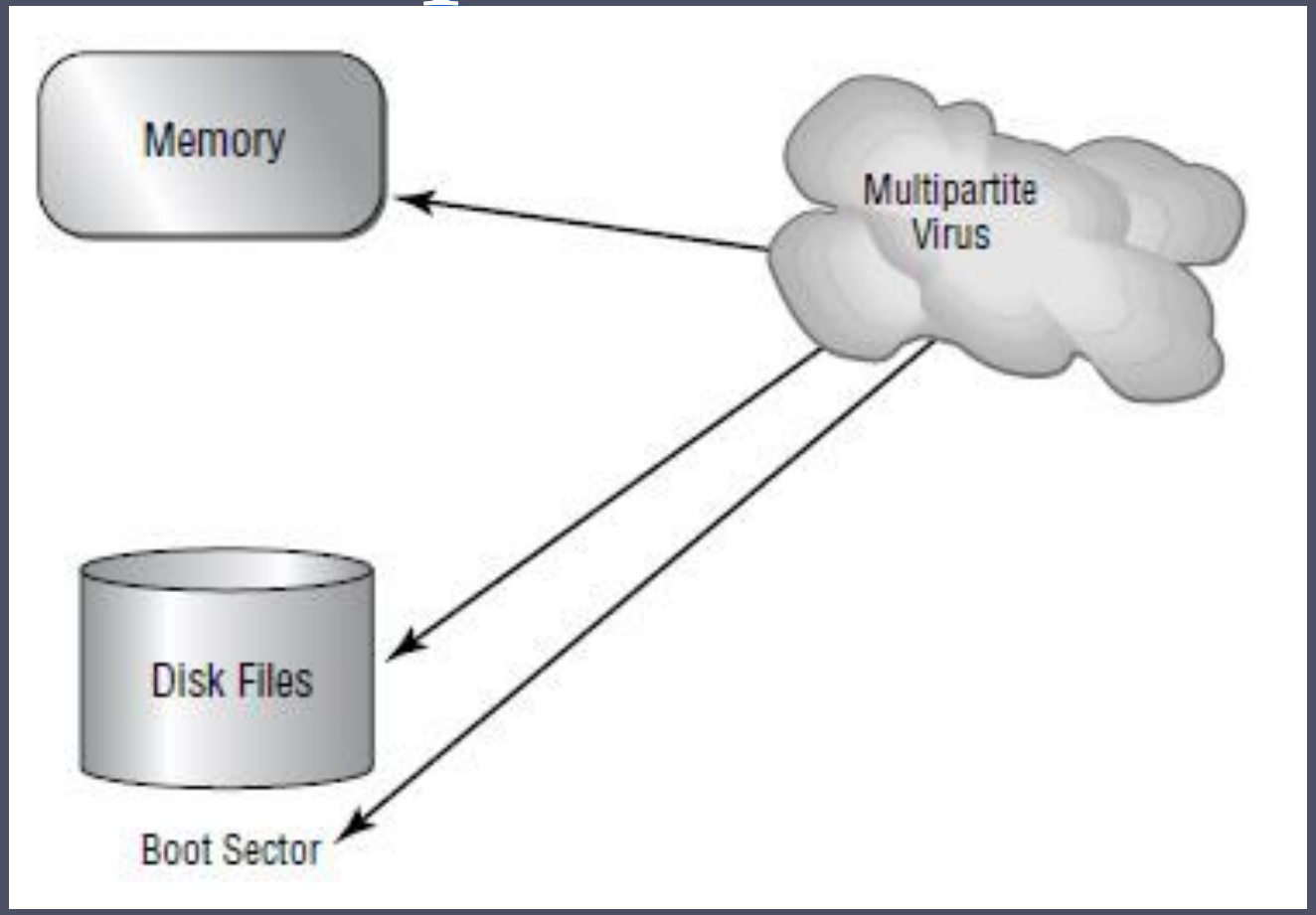
Metamorphic virus



- Metamorphic malware infections differ on disk and in memory.

Malware, Rootkits & Botnets A Beginner's Guide.pdf, P-135 (162/385)

Multipartite Virus



- A multipartite virus commencing an attack on a system

CompTIA Security+ Study Guide, P-85 (133/679)

6.4 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	• Worm e-mails a copy of itself to other systems • Sends itself as an attachment via an instant message service
File sharing	• Creates a copy of itself or infects a file as a virus on removable media
Remote execution capability	• Worm executes a copy of itself on another system
Remote file access or transfer capability	• Worm uses a remote file access or transfer service to copy itself from one system to the other
Remote login capability	• Worm logs onto a remote system as a user and then uses commands to copy itself from one system to the other

6.4.1 Target Discovery

- Scanning (or fingerprinting)
  - First function in the propagation phase for a network worm
  - Searches for other systems to infect
- Random
  - Each compromised host probes random addresses in the IP address space using a different seed
  - This produces a high volume of Internet traffic which may cause generalized disruption even before the actual attack is launched
- Hit-list
  - The attacker first compiles a long list of potential vulnerable machines
  - Once the list is compiled the attacker begins infecting machines on the list
  - Each infected machine is provided with a portion of the list to scan
  - This results in a very short scanning period which may make it difficult to detect that infection is taking place
- Topological
  - This method uses information contained on an infected victim machine to find more hosts to scan
- Local subnet
  - If a host can be infected behind a firewall that host then looks for targets in its own local network
  - The host uses the subnet address structure to find other hosts that would otherwise be protected by the firewall

6.4.2 Worm Propagation Model

$$\frac{dI(t)}{dt} = \beta I(t)\, S(t)$$

where

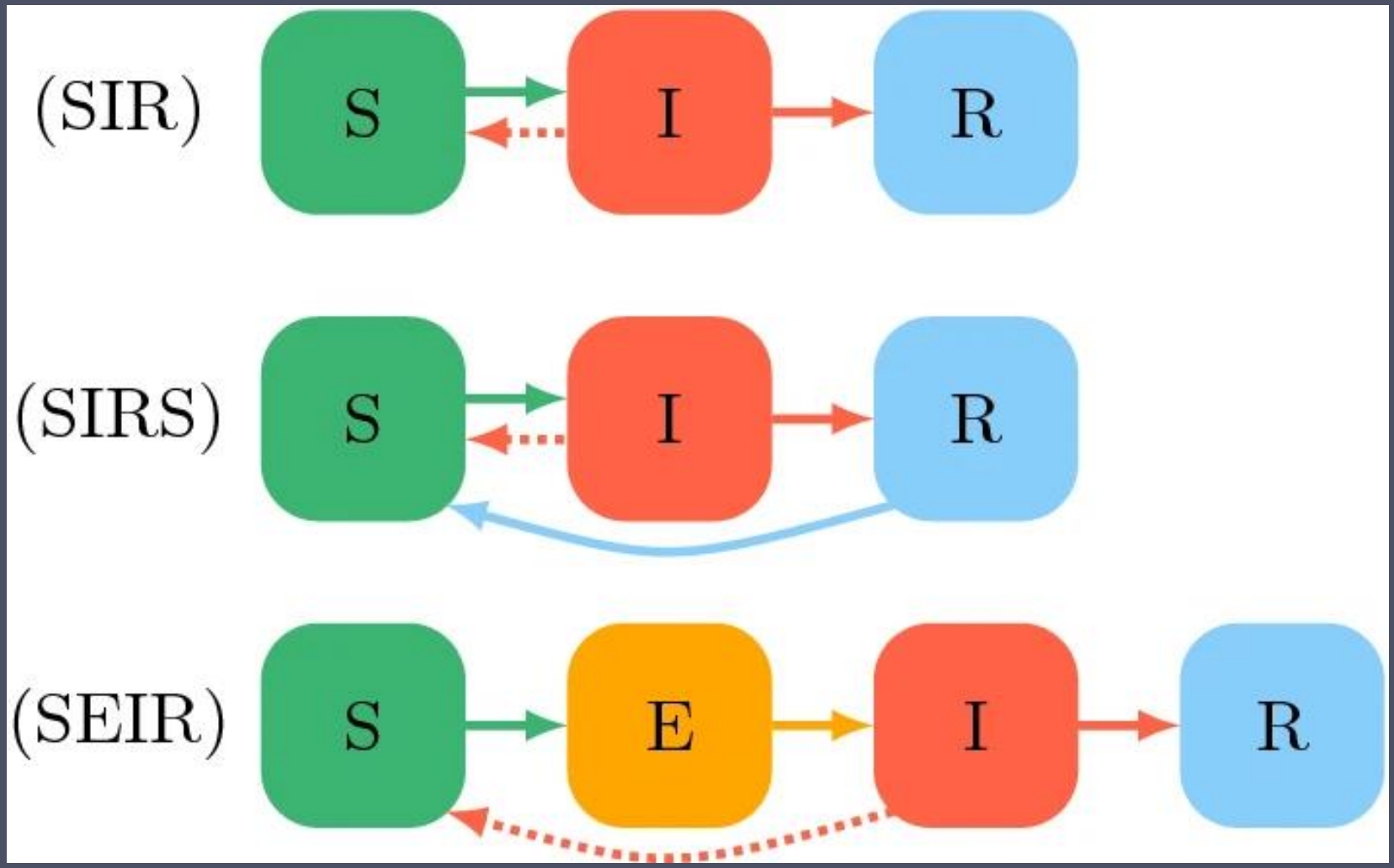
$I(t)$ = number of individuals infected as of time t

$S(t)$ = number of susceptible individuals (susceptible to infection but not yet infected) at time t

β = infection rate

N = size of the population, $N = I(t) + S(t)$

SIR Epidemics Modeling



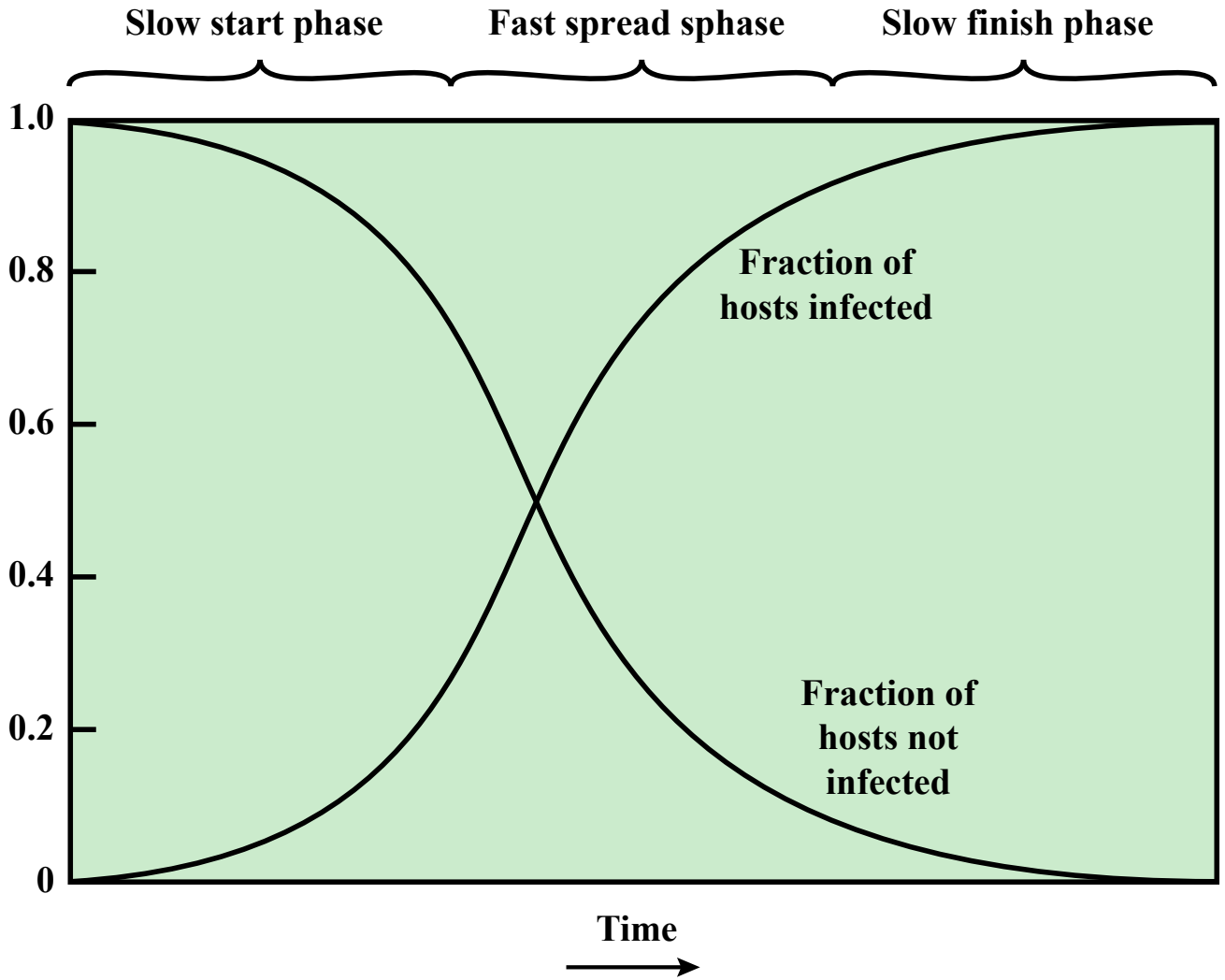


Figure 6.2 Worm Propagation Model

6.4.3 Morris Worm

- Earliest significant worm infection
- Released by Robert Morris in 1988
- Designed to spread on UNIX systems
  - Attempted to crack local password file to use login/password to logon to other systems
  - Exploited a bug in the finger protocol which reports the whereabouts of a remote user
  - Exploited a trapdoor in the debug option of the remote process that receives and sends mail
- Successful attacks achieved communication with the operating system command interpreter
  - Sent interpreter a bootstrap program to copy worm

6.4.4 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

Recent Worm Attacks

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.
Duqu	Late 2011	Code similar to Stuxnet Aim we cyber-espionage
Flame	2012	Middle-Eastern countries

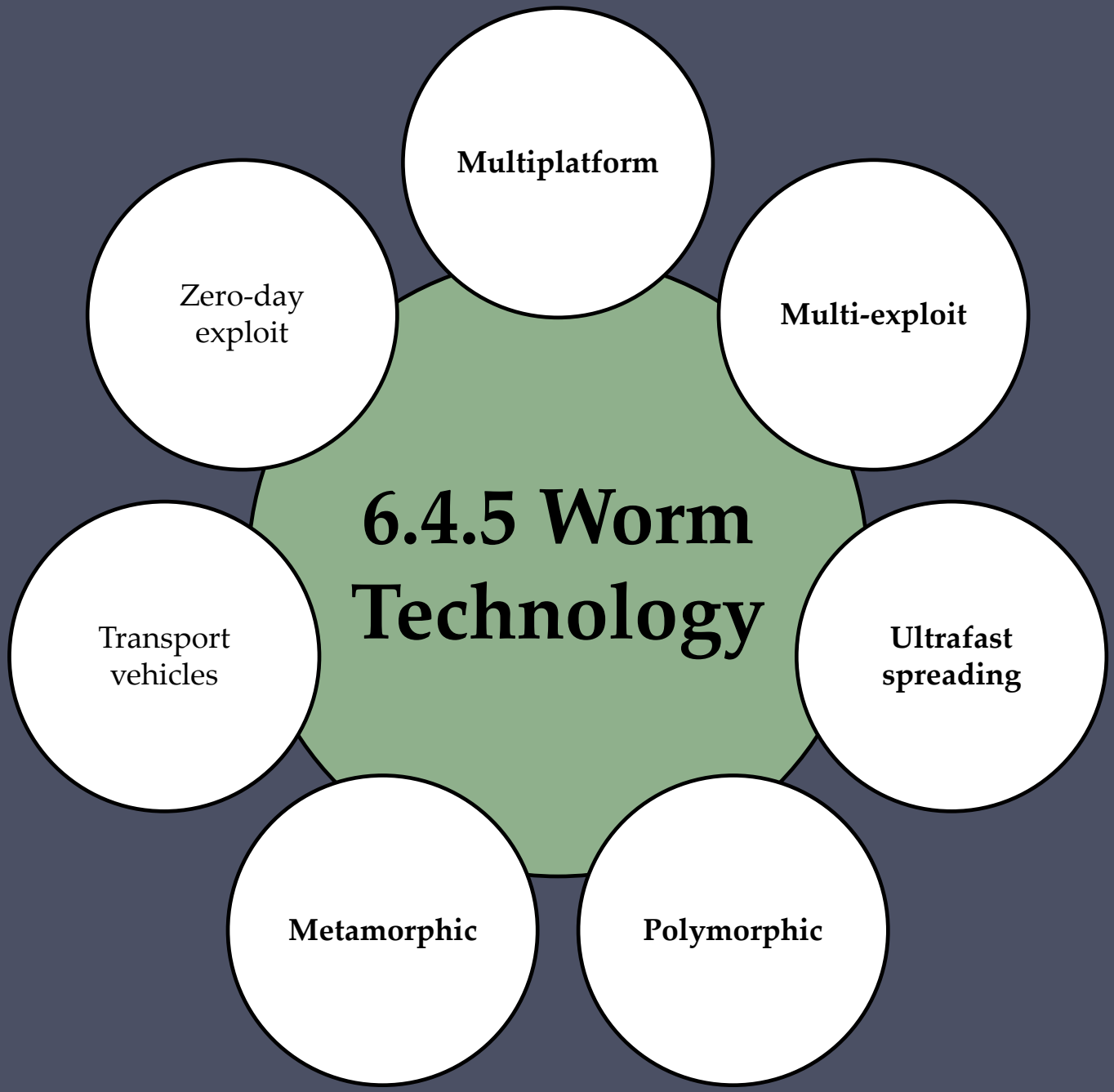
WannaCry

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 SMB 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


6.4.5 Worm Technology
Multiplatform
Multi-exploit
Ultrafast spreading
Polymorphic
Metamorphic
Transport vehicles
Zero-day exploit

6.4.6 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
  - E-mail attachments
  - Downloads from untrusted sites or of untrusted software

6.4.7 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

6.4.8 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

6.4.9 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

6.5 Social Engineering

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

Spam

- Unsolicited bulk e-mail
- Significant carrier of malware
- Used for phishing attacks

Trojan horse

- Program or utility containing harmful hidden code
- Used to accomplish functions that the attacker could not accomplish directly

Mobile phone Trojans

- First appeared in 2004 (Skuller)
- Target is the smartphone

6.6 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 incresasingly 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

6.6.2 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

6.6.3 Logic bomb

- Code embedded in the malware that is set to "explode" when certain conditions are met

6.7 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

6.7 .2 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

6.8Payload – 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

6.9 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

6.9.2 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

6.9.3 Rootkit Classification Characteristics (1/5)

Persistent	**Memory based**	**User mode**
Kernel mode	**Virtual machine based**	**External mode**

Rootkit Classification Characteristics (2/5)

- 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.

Rootkit Classification Characteristics (3/5)

- 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 don't include entries identifying the files associated with the rootkit.

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.

Rootkit Classification Characteristics (4/5)

Virtual machine based: This type of rootkit installs a lightweight virtual machine monitor, and 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.

Rootkit Classification Characteristics (5/5)

- This classification shows a continuing arms race between rootkit authors, who exploit ever more stealthy mechanisms to hide their code, and those who develop mechanisms to harden systems against such subversion, or to detect when it has occurred.
- Much of this advance is associated with finding "layer-below" forms of attack. The early rootkits worked in user mode, modifying utility programs and libraries in order to hide their presence.
- The changes they made could be detected by code in the kernel, as this operated in the layer below the user. Later-generation rootkits used more stealthy techniques, as we discuss next.

6.9.3 Kernel Mode Rootkits (1/4)

- The next generation of rootkits moved down a layer, making changes inside the kernel and co-existing with the operating systems code, in order to make their detection much harder. Any "anti-virus" program would now be subject to the same "low-level" modifications that the rootkit uses to hide its presence. However, methods were developed to detect these changes.
- Programs operating at the user level interact with the kernel through system calls. Thus, system calls are a primary target of kernel-level rootkits to achieve concealment. As an example of how rootkits operate, we look at the implementation of system calls in Linux.

Kernel Mode Rootkits (2/4)

- In Linux, each system call is assigned a unique *syscall number* . When a user-mode process executes a system call, the process refers to the system call by this number.
- The kernel maintains a system call table with one entry per system call routine; each entry contains a pointer to the corresponding routine. The syscall number serves as an index into the system call table.

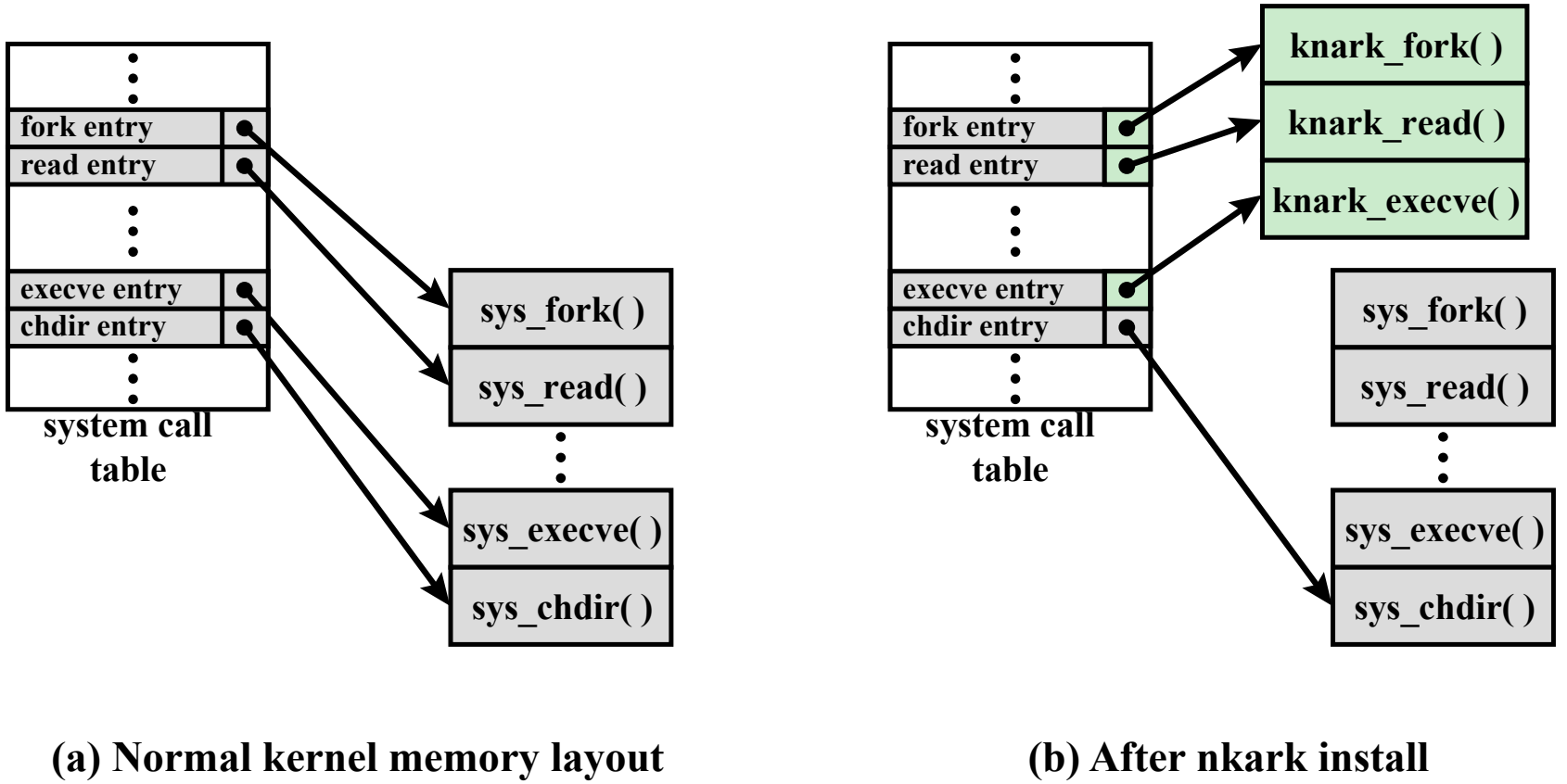
Kernel Mode Rootkits (3/4)

[LEVI06] lists three techniques that can be used to change system calls:

- **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 6.5 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.

Kernel Mode Rootkits (4/4)



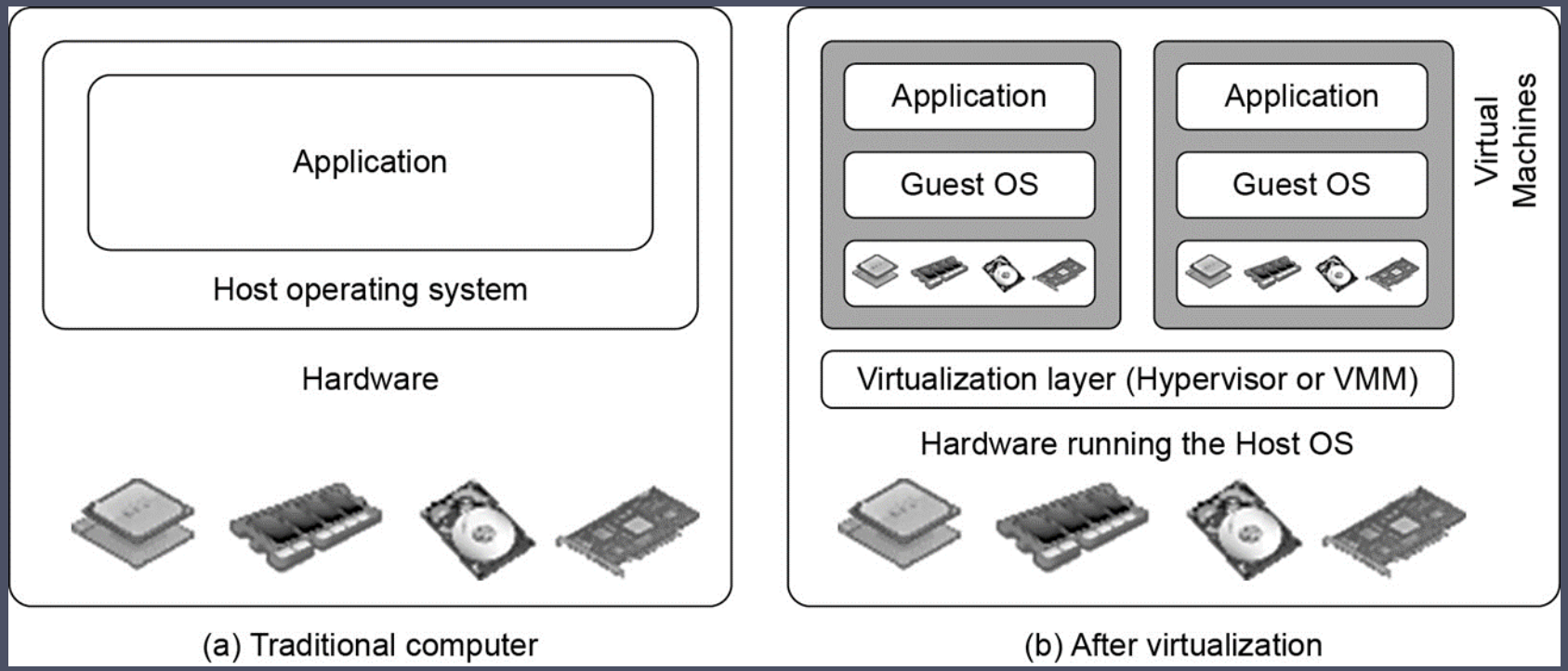
(a) Normal kernel memory layout

(b) After nkark install

System Call Table Modification

Concept of Virtual Machines

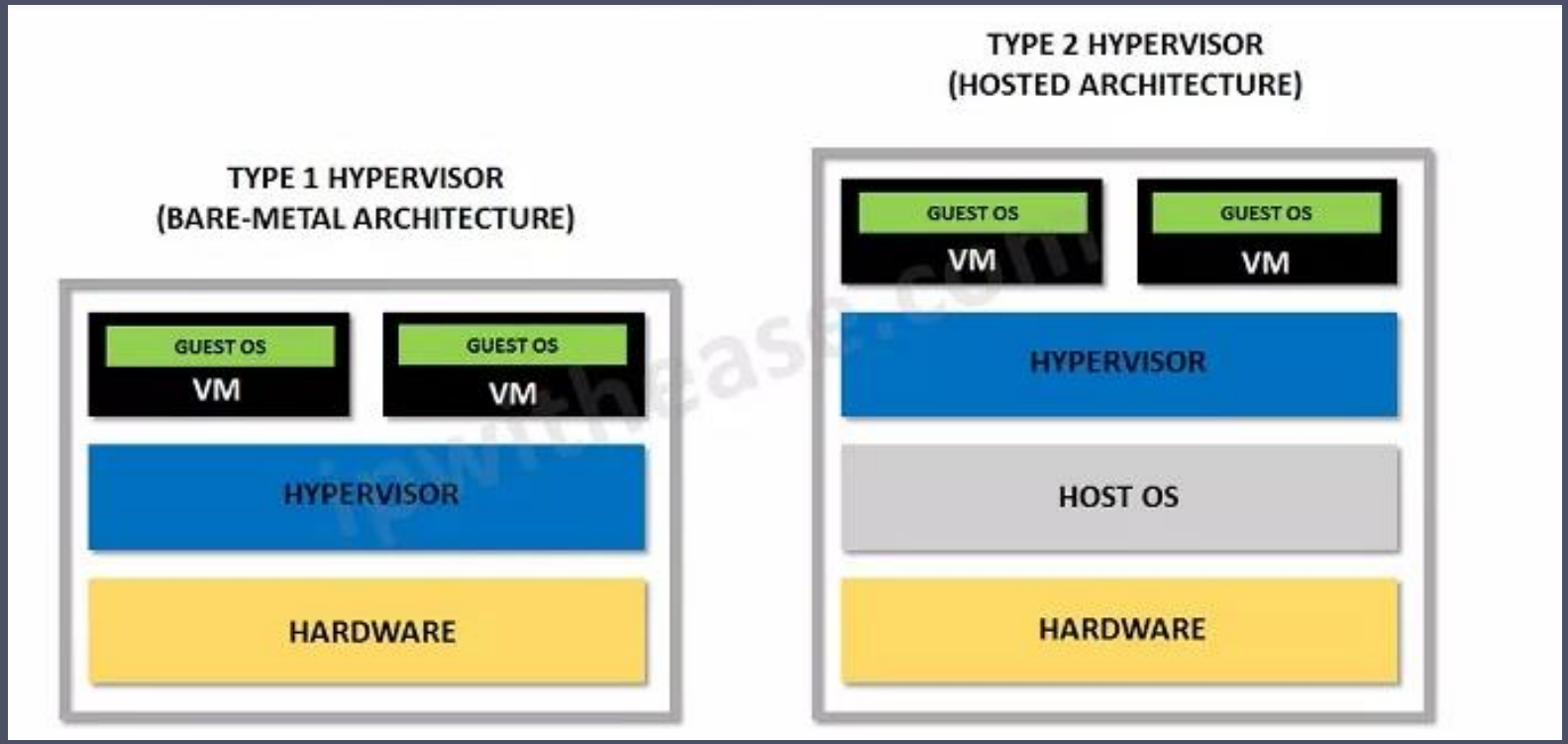
Traditional Computer Vs Virtual machine



Difference between Traditional Computer and Virtual machines

KaiHwang\Chapter_03_Mod_2.ppt

Type-1 vs Type-2 Hypervisors



https://networkinterview.com/type-1-vs-type-2-hypervisors/

Type-1 vs Type-2 Hypervisors: Difference Table

PARAMETER	TYPE 1 HYPERVISOR	TYPE 2 HYPERVISOR
Terminology	Run directly on System Hardware	Run on host Operating System
Booting	Boots before Operating system	Cannot boot until Operating System is up and running
Other names	Native/ Bare metal / Embedded Hypervisor	Host OS Hypervisor
Efficiency	Comparatively better	Inferior
Support	Hardware virtualization	Operating system virtualization
Availability	Comparatively better	Inferior
Performance	High	Low
Security	Comparatively better	Inferior
Usage	In Datacentre	By Lab and IT professionals
Examples	VMware ESXi and Citrix XEN Server	KVM, Virtual Box, VMware Server and Microsoft Virtual PC.

6.9.3Virtual Machine and Other External Rootkits

- The latest generation of rootkits uses code that is entirely invisible to the targeted operating system.
- This can be done using a rogue or compromised virtual machine monitor or hypervisor, often aided by the hardware virtualization support provided in recent processors.
- The rootkit code then runs entirely below the visibility of even kernel code in the targeted operating system, which is now unknowingly running in a virtual machine, and capable of being silently monitored and attacked by the code below.
- Several prototypes of virtualized rootkits were demonstrated in 2006.
  - SubVirt attacked Windows systems running under either Microsoft's Virtual PC or Vmware Workstation hypervisors by modifying the boot process they used.
  - These changes did make it possible to detect the presence of the rootkit.

Virtual Machine and Other External Rootkits

- However, the Blue Pill rootkit was able to subvert a native Windows Vista system by installing a thin hypervisor below it, then seamlessly continuing execution of the Vista system in a virtual machine.
- As it only required the execution of a rogue driver by the Vista kernel, this rootkit could install itself while the targeted system was running, and is much harder to detect.
- This type of rootkit is a particular threat to systems running on modern processors with hardware virtualization support, but where no hypervisor is in use.

Virtual Machine and Other External Rootkits

- Other variants exploit the System Management Mode (SMM) in Intel processors that is used for low-level hardware control, or the BIOS code used when the processor first boots.
- Such code has direct access to attached hardware devices, and is generally invisible to code running outside these special modes.
- To defend against these types of rootkits, the entire boot process must be secure, ensuring that the operating system is loaded and secured against the installation of these types of malicious code.
- This needs to include monitoring the loading of any hypervisor code to ensure it is legitimate

6.10 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

Malware Countermeasure Approaches

- **Generality**: The approach taken should be able to handle a wide variety of attacks.
- **Timeliness**: The approach should respond quickly so as to limit the number of infected programs or systems and the consequent activity.
- **Resiliency**: The approach should be resistant to evasion techniques employed by attackers to hide the presence of their malware.

Malware Countermeasure Approaches

- **Minimal denial-of-service costs:** The approach should result in minimal reduction in capacity or service due to the actions of the countermeasure software, and should not significantly disrupt normal operation.

- **Transparency:** The countermeasure software and devices should not require modification to existing (legacy) OSs, application software, and hardware.
- **Global and local coverage:** The approach should be able to deal with attack sources both from outside and inside the enterprise network.

6.10.1 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

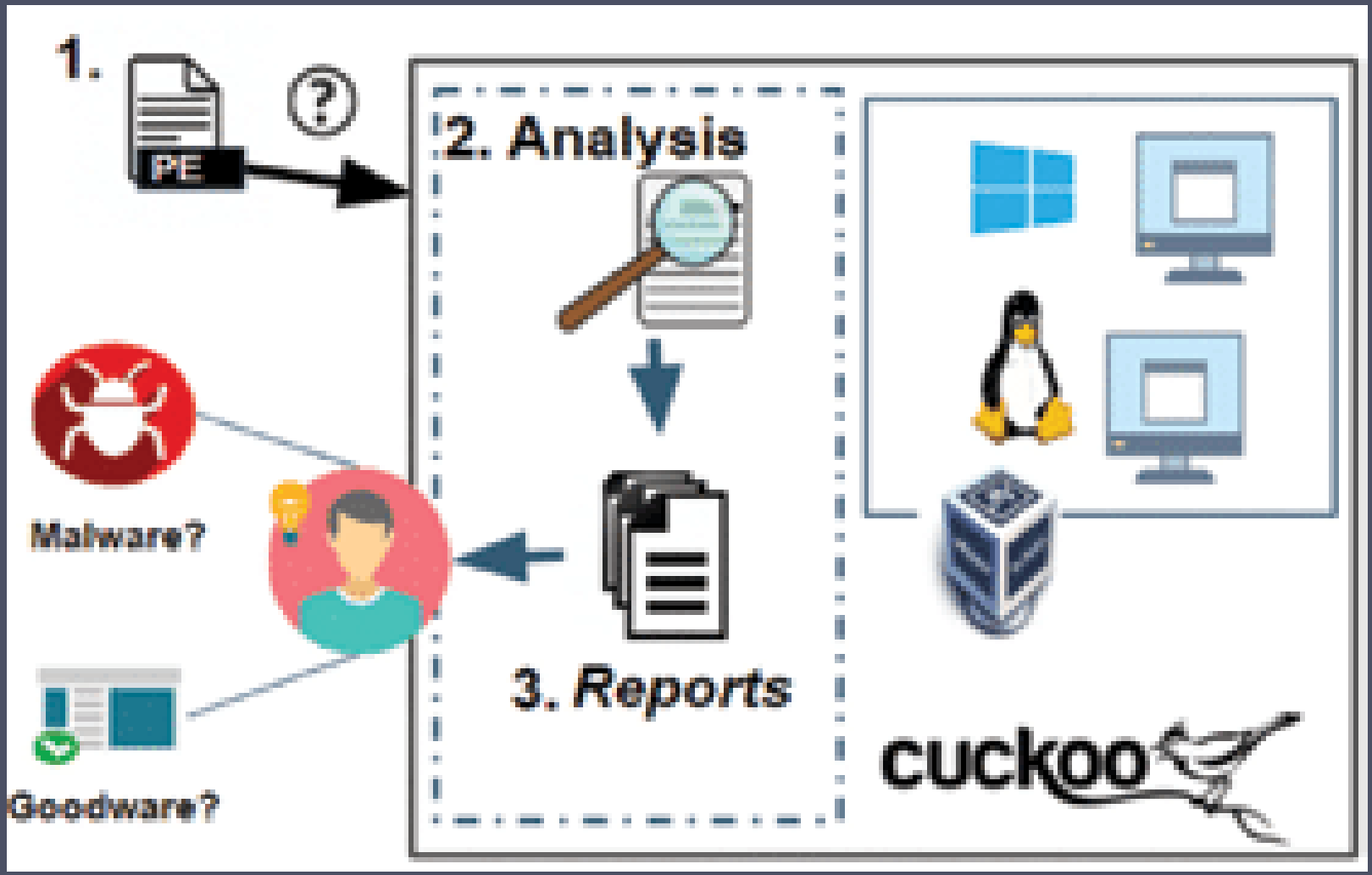
Third generation: activity traps

- 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

6.10.2 Sandbox Analysis



https://www.researchgate.net/figure/MALWARE-ANALYSIS-IN-CUCKOO-SANDBOX_fig5_329496012

6.10.2 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

6.10.3 Host-Based Behavior-Blocking Software

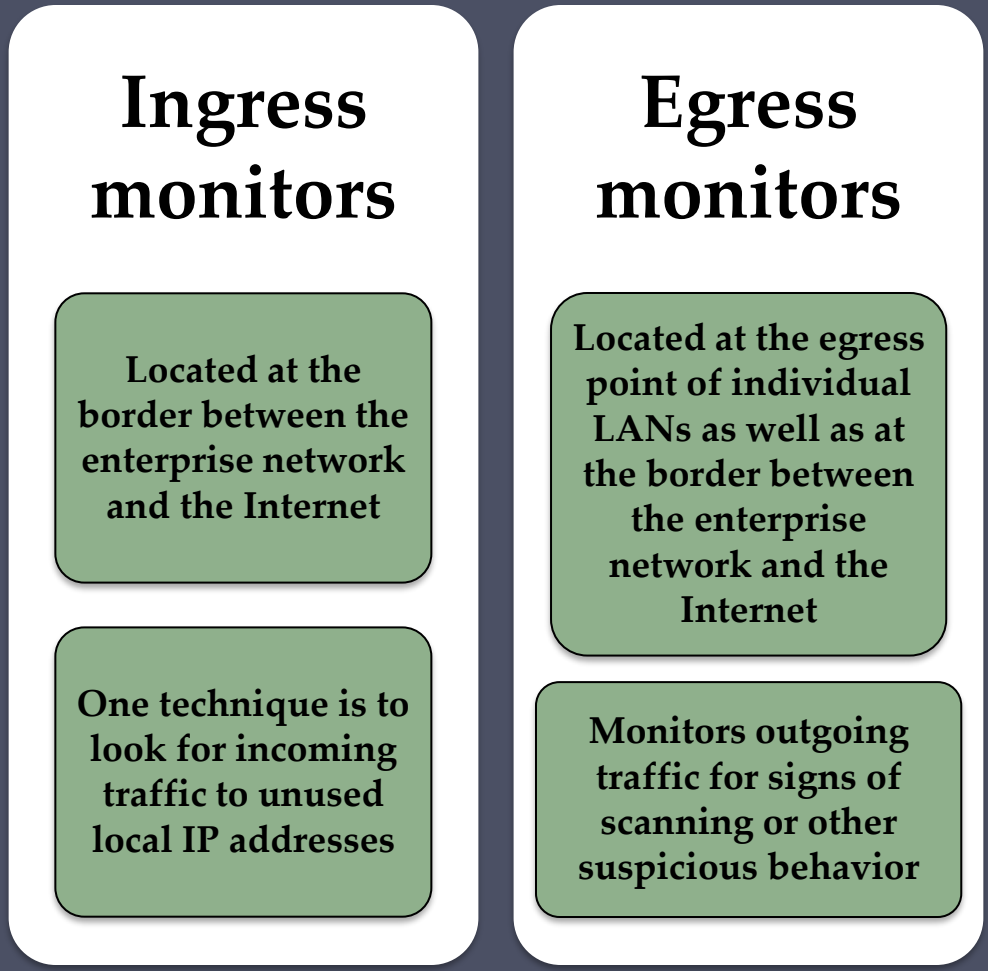
- Integrates with the operating system of a host computer and monitors program behavior in real time for malicious action
  - Blocks potentially malicious actions before they have a chance to affect the system
  - Blocks software in real time so it has an advantage over anti-virus detection techniques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

6.10.3 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 so
  - Distributed Intelligence Gathering Approaches



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-stealthing-backdoors, rootkits
  - Backdoor
  - Rootkit
  - Kernel mode rootkits
  - Virtual machine and other external rootkits
- 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
  - Distributed intelligence gathering approaches