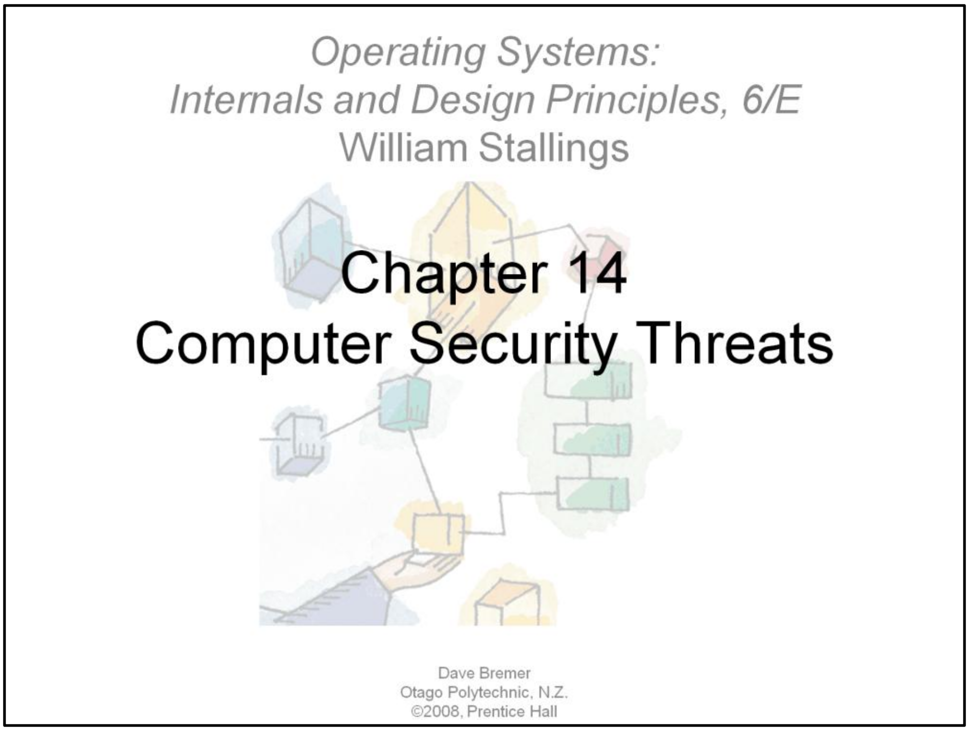
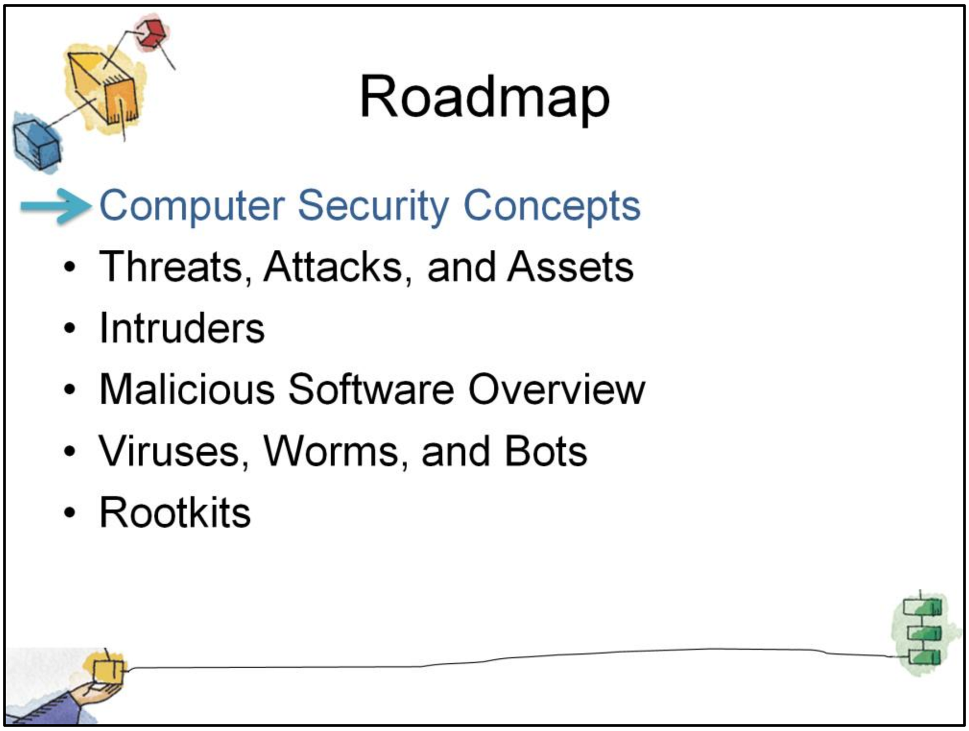


Operating Systems:
Internals and Design Principles, 6/E
William Stallings

Chapter 14
Computer Security Threats

Dave Bremer
Otago Polytechnic, N.Z.

These slides are intended to help a teacher develop a presentation. This PowerPoint covers the entire chapter and includes too many slides for a single delivery. Professors are encouraged to adapt this presentation in ways which are best suited for their students and environment.



Roadmap

→ Computer Security Concepts

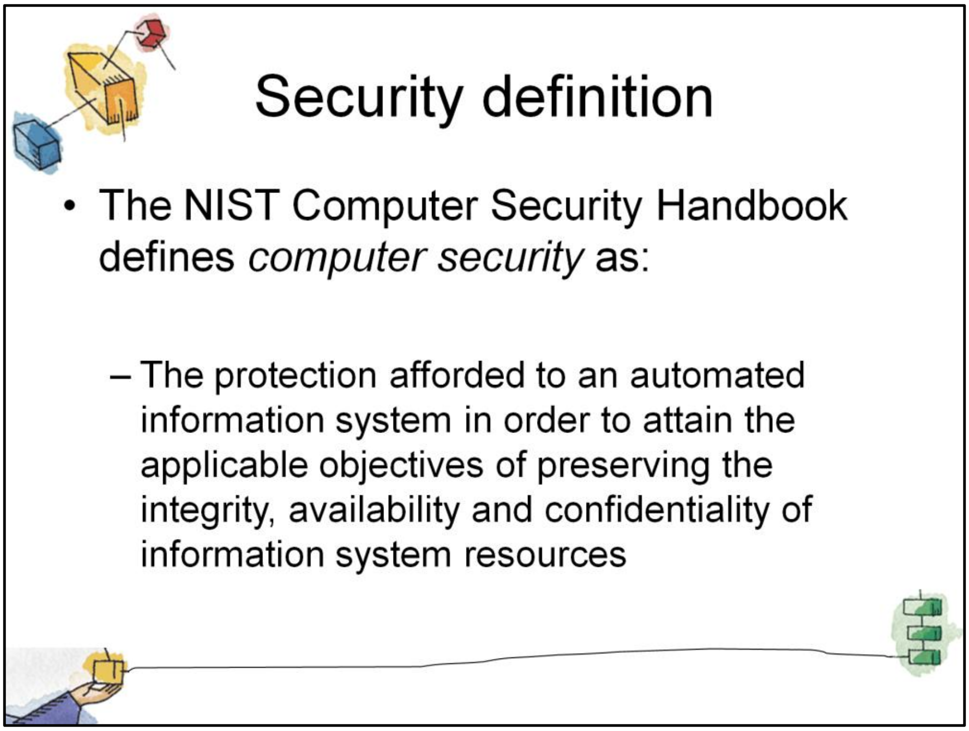
- Threats, Attacks, and Assets
- Intruders
- Malicious Software Overview
- Viruses, Worms, and Bots
- Rootkits

Chapter 14 begins with an overview of computer security concept.

Then the chapter provides a survey of the threats to computer security.

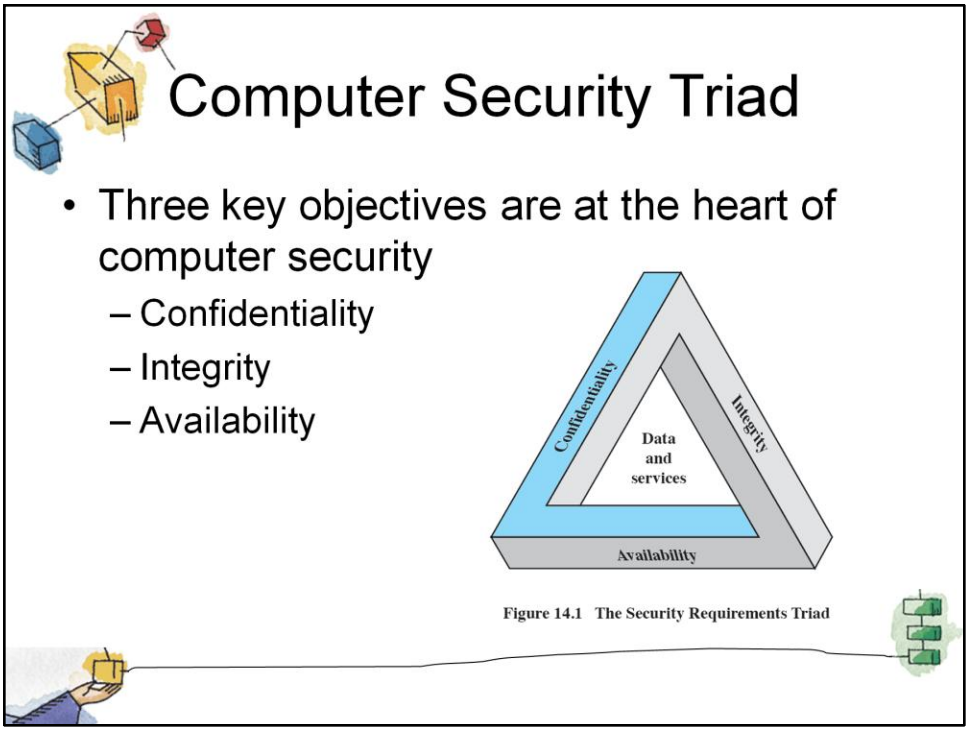
he bulk of the chapter is devoted to four major threats:

- viruses,
- worms,
- bots, and
- rootkits.



Security definition

- The NIST Computer Security Handbook defines *computer security* as:
  - The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability and confidentiality of information system resources



Computer Security Triad

- Three key objectives are at the heart of computer security
  - Confidentiality
  - Integrity
  - Availability

Confidentiality
Integrity
Availability
Data
and
services

Figure 14.1 The Security Requirements Triad

Confidentiality: Covering two related concepts:

— Data confidentiality: Assures that private or confidential information is not made available or disclosed to unauthorized individuals

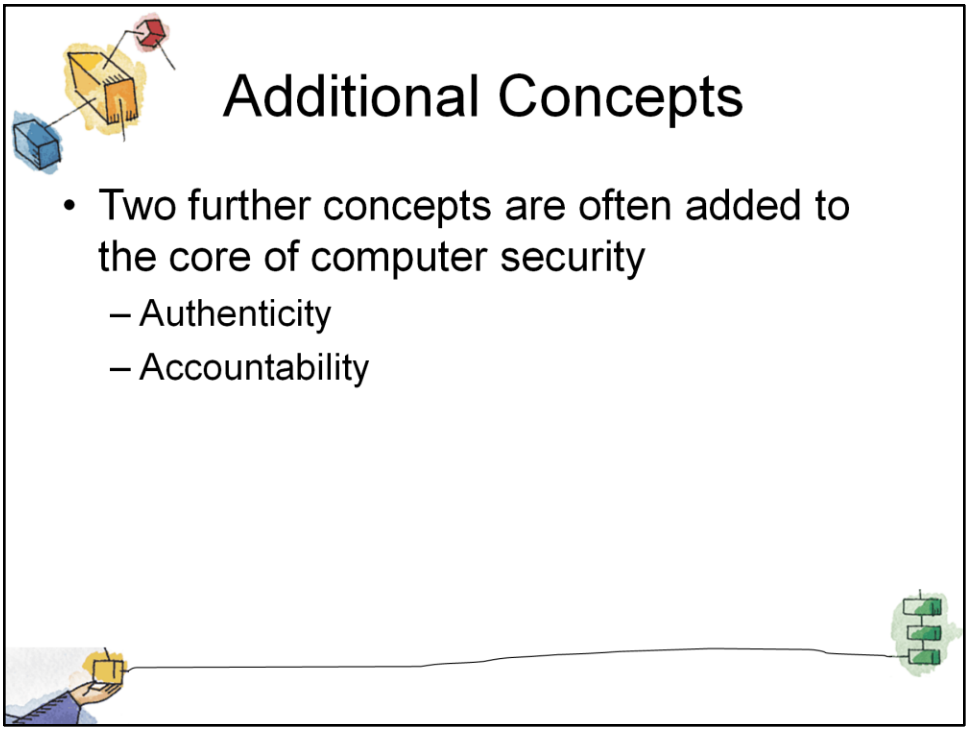
—Privacy: Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed

Integrity: Also covers two related concepts:

—Data integrity: Assures that information and programs are changed only in a specified and authorized manner

—System integrity: Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system

Availability: Assures that systems work promptly and service is not denied to authorized users



Additional Concepts

- Two further concepts are often added to the core of computer security
  - Authenticity
  - Accountability

Authenticity:

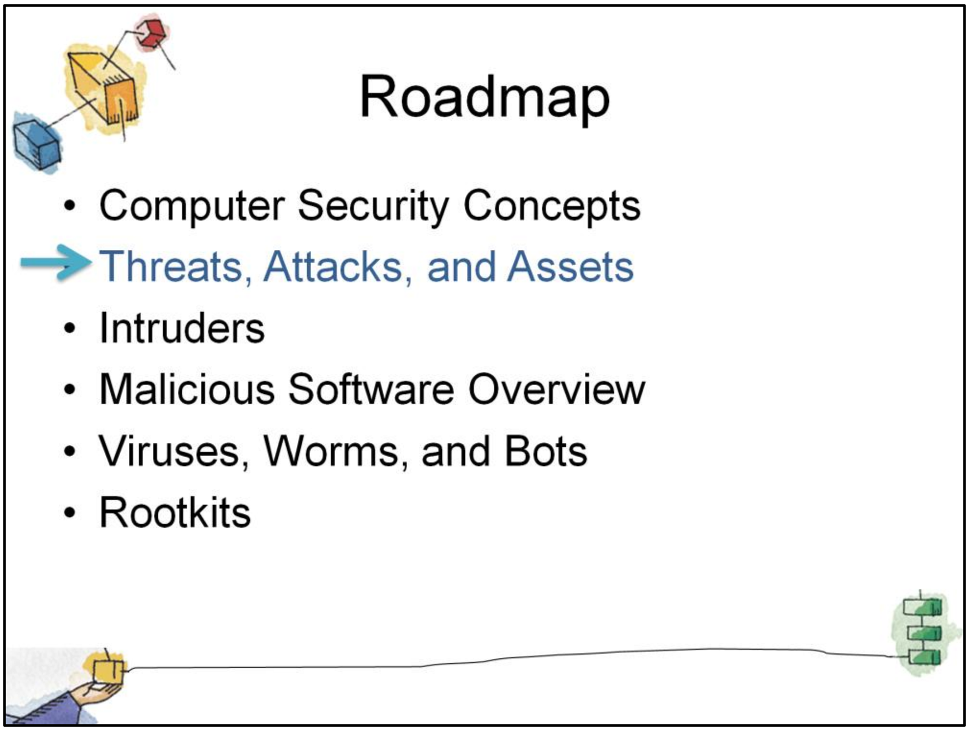
• The property of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, a message, or message originator.

•This means verifying that users are who they say they are and that each input arriving at the system came from a trusted source.

Accountability:

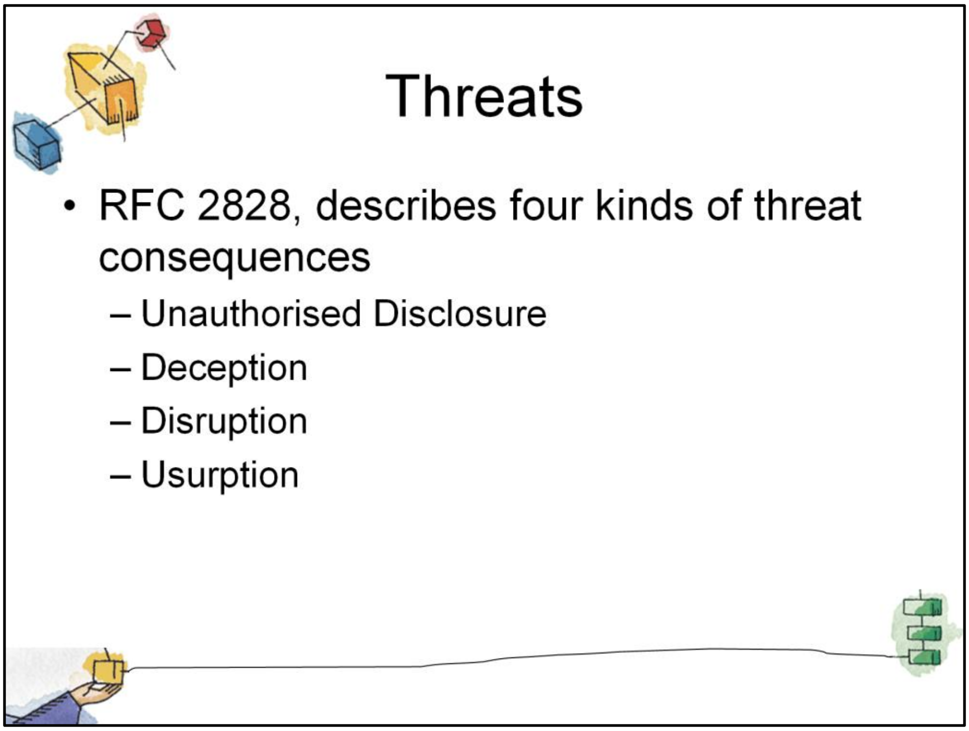
• The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity.

• This supports nonrepudiation, deterrence, fault isolation, intrusion detection and prevention, and after-action recovery and legal action.



Roadmap

- Computer Security Concepts
- → Threats, Attacks, and Assets
- Intruders
- Malicious Software Overview
- Viruses, Worms, and Bots
- Rootkits



Threats

- RFC 2828, describes four kinds of threat consequences
  - Unauthorised Disclosure
  - Deception
  - Disruption
  - Usurption

Unauthorized Disclosure

• A circumstance or event whereby an entity gains access to data for which the entity is not authorized.

Deception

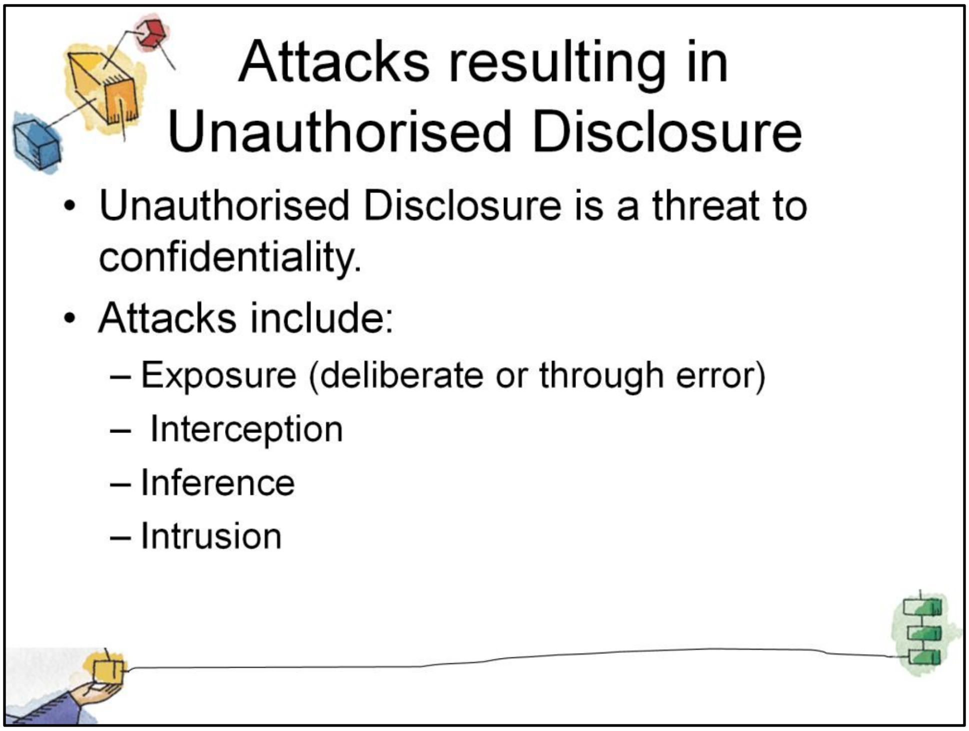
• A circumstance or event that may result in an authorized entity receiving false data and believing it to be true.

Disruption

• A circumstance or event that interrupts or prevents the correct operation of system services and functions.

Usurpation

• A circumstance or event that results in control of system services or functions by an unauthorized entity



Attacks resulting in Unauthorised Disclosure

- Unauthorised Disclosure is a threat to confidentiality.
- Attacks include:
  - Exposure (deliberate or through error)
  - Interception
  - Inference
  - Intrusion

Exposure:

- This can be deliberate, as when an insider intentionally releases sensitive information, such as credit card numbers, to an outsider.
- Can also be the result of a human, hardware, or software error,which results in an entity gaining unauthorized knowledge of sensitive data.

Interception:

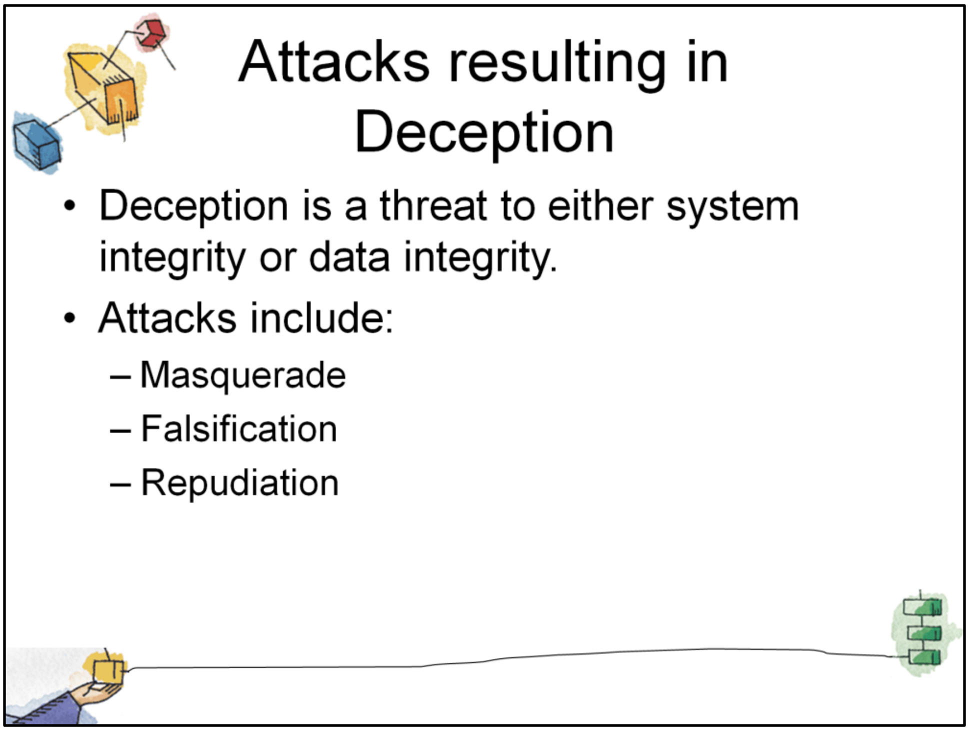
- On a shared local area network (LAN), such as a wireless LAN or a broadcast Ethernet, any device attached to the LAN can receive a copy of packets intended for another device.
- On the Internet, a determined hacker can gain access to e-mail traffic and other data transfers.

Inference:

- An adversary is able to gain information from observing the pattern of traffic on a network, such as the amount of traffic between particular pairs of hosts on the network.
- Another example is the inference of detailed information from a database by a user who has only limited access

Intrusion:

- An adversary gaining unauthorized access to sensitive data by overcoming the system’s access control protections.



Attacks resulting in Deception

- Deception is a threat to either system integrity or data integrity.
- Attacks include:
  – Masquerade
  – Falsification
  – Repudiation

Masquerade:

• An attempt by an unauthorized user to gain access to a system by posing as an authorized user; this could happen if the unauthorized user has learned another user’s logon ID and password.

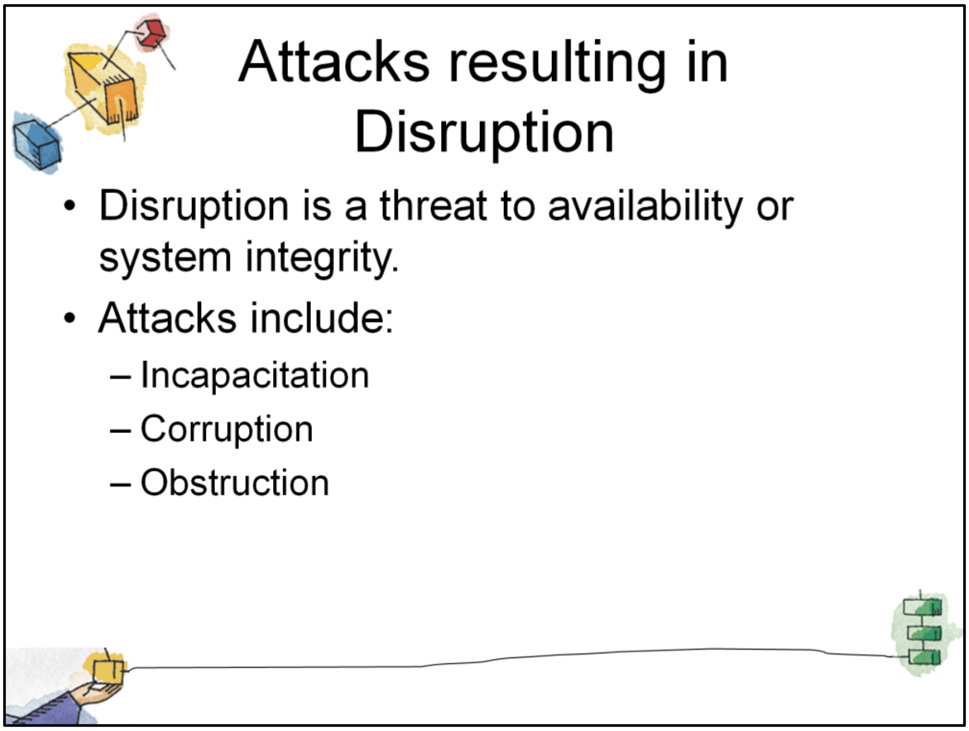
• Another example is malicious logic, such as a Trojan horse, that appears to perform a useful or desirable function but actually gains unauthorized access to system resources or tricks a user into executing other malicious logic.

Falsification:

•This refers to the altering or replacing of valid data or the introduction of false data into a file or database. For example, a student my alter his or her grades on a school database.

Repudiation:

• A user either denies sending data or a user denies receiving or possessing the data.



Attacks resulting in Disruption

- Disruption is a threat to availability or system integrity.
- Attacks include:
  - Incapacitation
  - Corruption
  - Obstruction

Incapacitation:

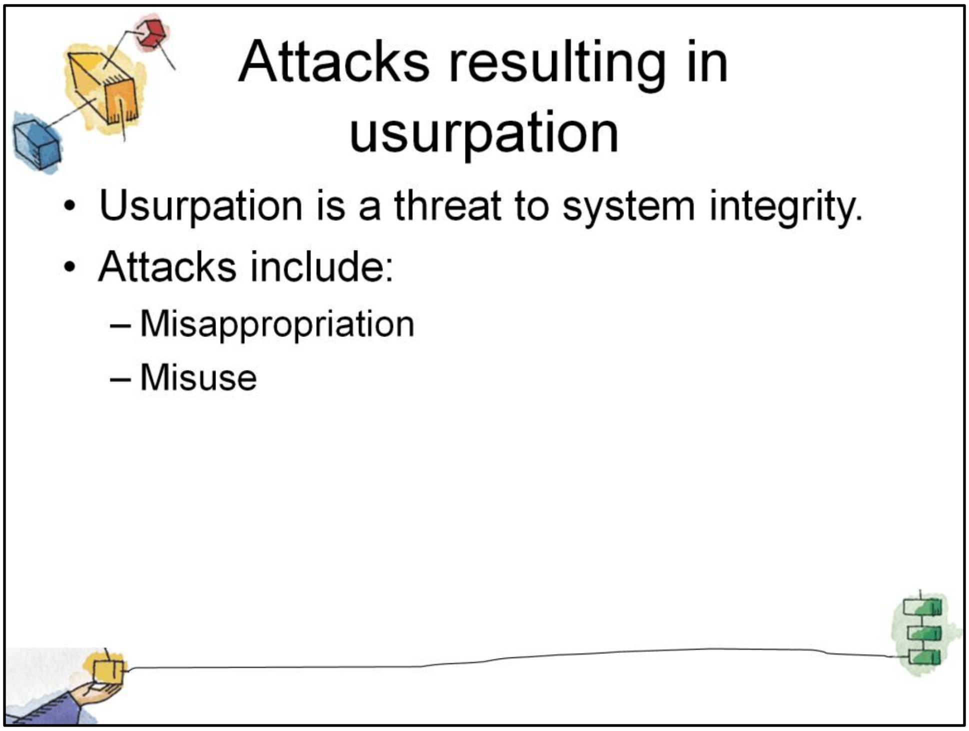
- This is an attack on system availability.
- This could occur as a result of physical destruction of or damage to system hardware.
- Often malicious software, such as Trojan horses, viruses, or worms, could operate in such a way as to disable a system or some of its services.

Corruption:

- This is an attack on system integrity.
- Malicious software in this context could operate in such a way that system resources or services function in an unintended manner.
- Or a user could gain unauthorized access to asystem and modify some of its functions. An example of the latter is a user
- placing backdoor logic in the system to provide subsequent access to a system and its resources by other than the usual procedure.

Obstruction:

- One way to obstruct system operation is to interfere with communications by disabling communication links or altering communication control information.
- Another way is to overload the system by placing excess burden on communication traffic or processing resources.



Attacks resulting in usurpation

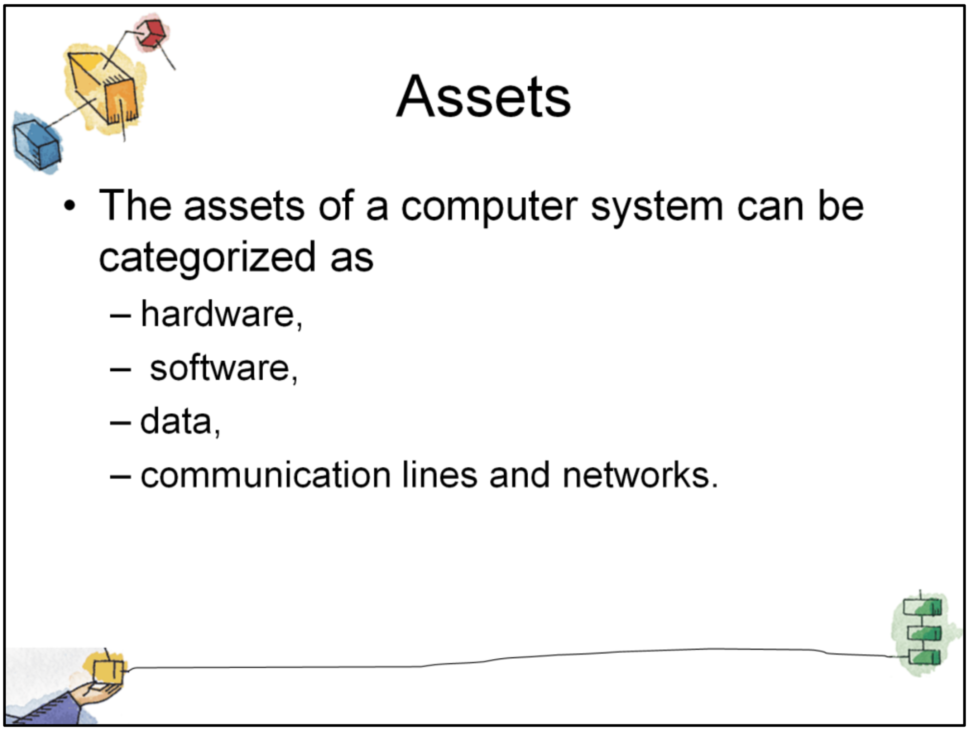
- Usurpation is a threat to system integrity.
- Attacks include:
  - Misappropriation
  - Misuse

Misappropriation:

- This can include theft of service.
- An example is an a distributed denial of service attack, when malicious software is installed on a number of hosts to be used as platforms to launch traffic at a target host.
- In this case, the malicious software makes unauthorized use of processor and operating system resources.

Misuse:

- Misuse can occur either by means of malicious logic or a hacker that has gained unauthorized access to a system.
- In either case, security functions can be disabled or thwarted.



Assets

- The assets of a computer system can be categorized as
  - hardware,
  - software,
  - data,
  - communication lines and networks.

Scope of System Security

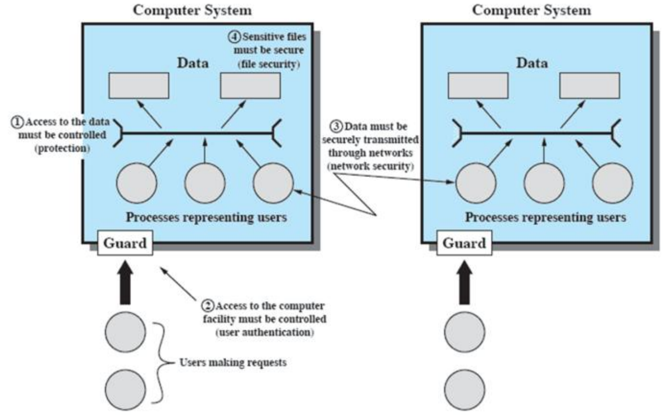
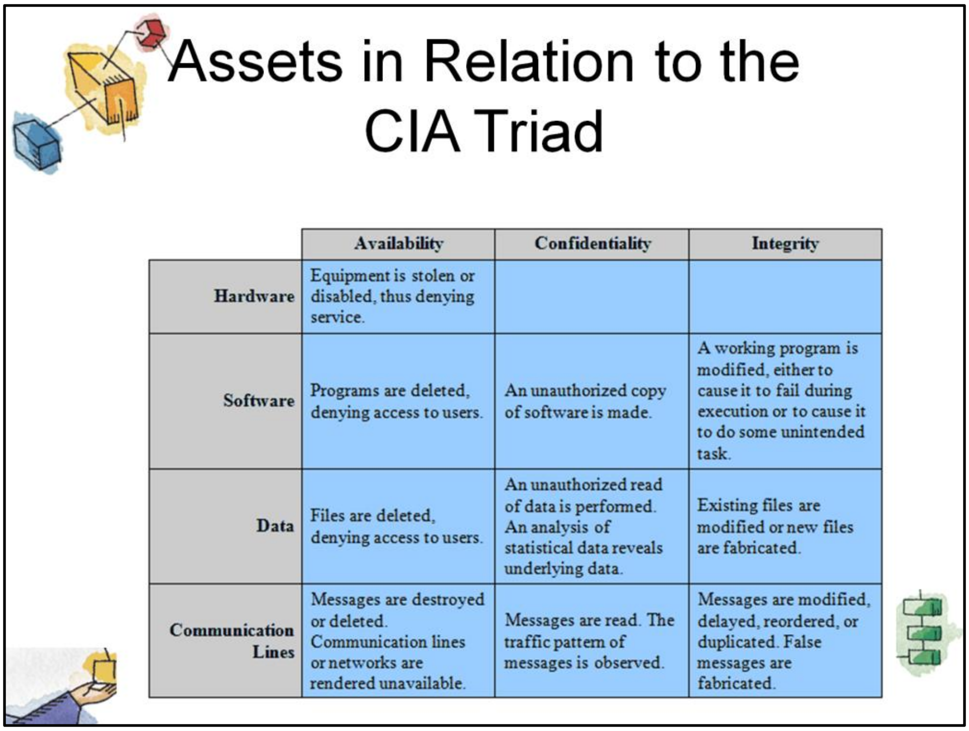

Computer System
④Sensitive files must be secure (file security)
Data
①Access to the data must be controlled (protection)
③Data must be securely transmitted through networks (network security)
Processes representing users
Guard
②Access to the computer facility must be controlled (user authentication)
Users making requests
Computer System
Data
Processes representing users
Guard

Figure 14.2 Scope of System Security



Assets in Relation to the CIA Triad

	Availability	Confidentiality	Integrity
Hardware	Equipment is stolen or disabled, thus denying service.		
Software	Programs are deleted, denying access to users.	An unauthorized copy of software is made.	A working program is modified, either to cause it to fail during execution or to cause it to do some unintended task.
Data	Files are deleted, denying access to users.	An unauthorized read of data is performed. An analysis of statistical data reveals underlying data.	Existing files are modified or new files are fabricated.
Communication Lines	Messages are destroyed or deleted. Communication lines or networks are rendered unavailable.	Messages are read. The traffic pattern of messages is observed.	Messages are modified, delayed, reordered, or duplicated. False messages are fabricated.

This table relates the assets to to the concepts of integrity, confidentiality, and availability introduced earlier.

Hardware

- A major threat to computer system hardware is the threat to availability.
- Hardware is the most vulnerable to attack and the least susceptible to automated controls.
- Threats include accidental and deliberate damage to equipment as well as theft.

Software

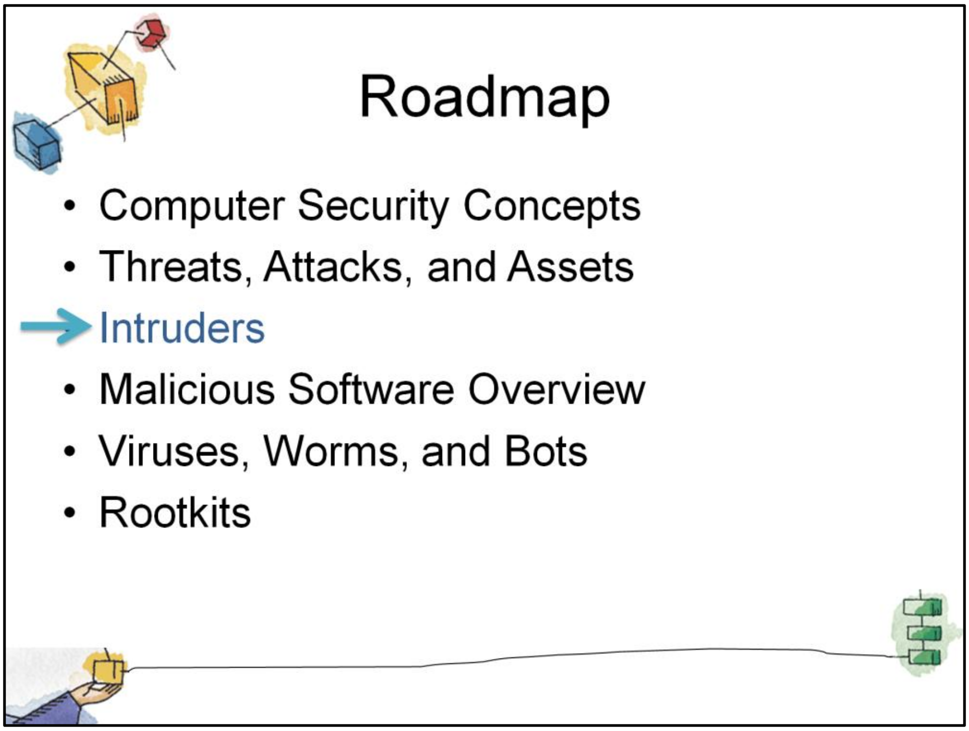
- Includes the operating system, utilities, and application programs.
- A key threat to software is an attack on availability through deletion, alteration or damage to render it useless, or worse, malicious

Data

- Involves files and other forms of data controlled by individuals, groups, and business organizations.
- Security concerns with respect to data are broad, encompassing availability, secrecy, and integrity. In the case of availability, the concern is with the destruction of data files, which can occur either accidentally or maliciously.

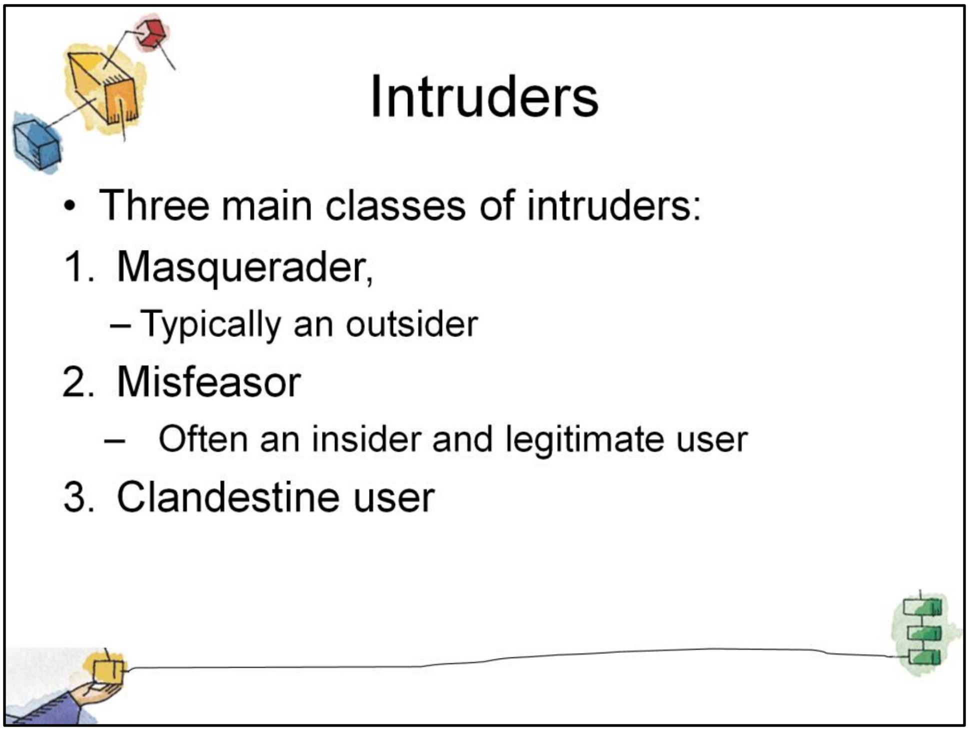
Communication Lines and Networks

- Network security attacks can be classified as **passive attacks** and **active attacks**.
- A passive attack attempts to learn or make use of information from the system but does not affect system resources.
- An active attack attempts to alter system resources or affect their operation.



Roadmap

- Computer Security Concepts
- Threats, Attacks, and Assets
- → Intruders
- Malicious Software Overview
- Viruses, Worms, and Bots
- Rootkits



Intruders

- Three main classes of intruders:

1. Masquerader,
   – Typically an outsider
2. Misfeasor
   – Often an insider and legitimate user
3. Clandestine user

Masquerader:

• An individual who is not authorized to use the computer and who penetrates a system’s access controls to exploit a legitimate user’s account

Misfeasor:

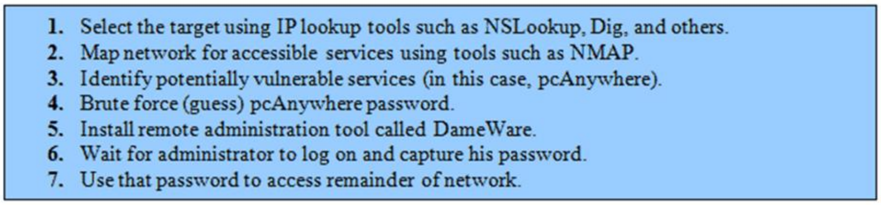
• A legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuses his or her privileges

Clandestine user:

• An individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection

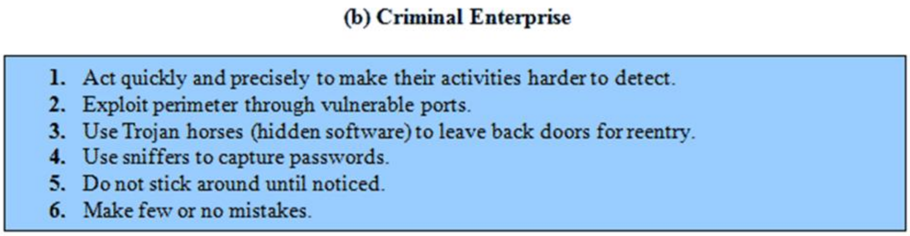
Intruder Behavior: Hackers

(a) Hacker



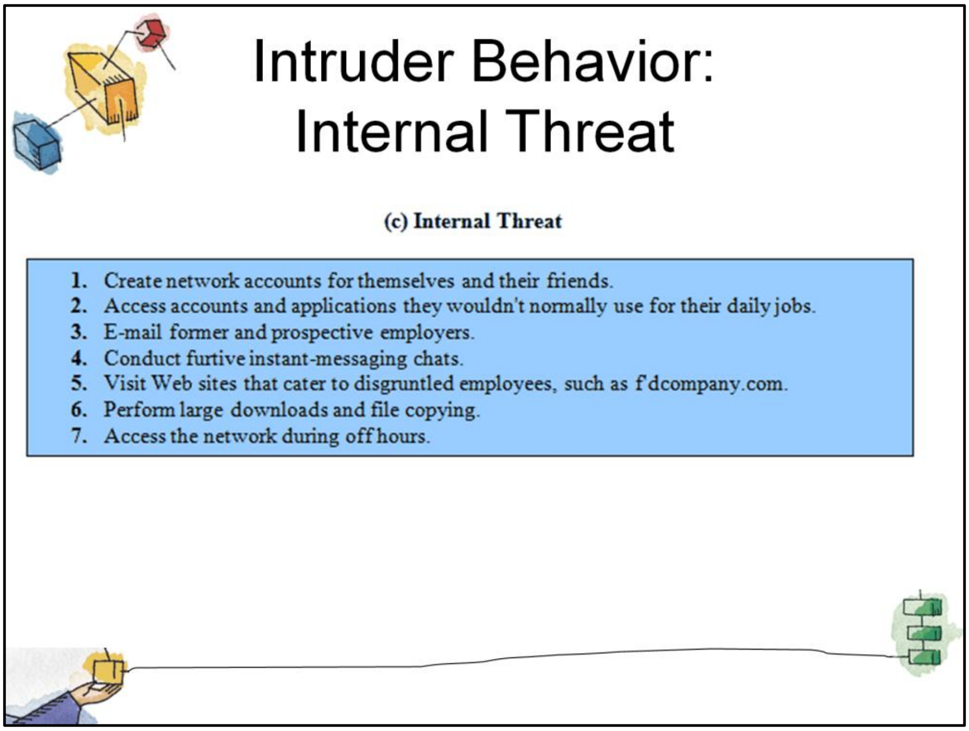
1. Select the target using IP lookup tools such as NSLookup, Dig, and others.
2. Map network for accessible services using tools such as NMAP.
3. Identify potentially vulnerable services (in this case, pcAnywhere).
4. Brute force (guess) pcAnywhere password.
5. Install remote administration tool called DameWare.
6. Wait for administrator to log on and capture his password.
7. Use that password to access remainder of network.

Intruder Behavior: Criminal Enterprise



(b) Criminal Enterprise

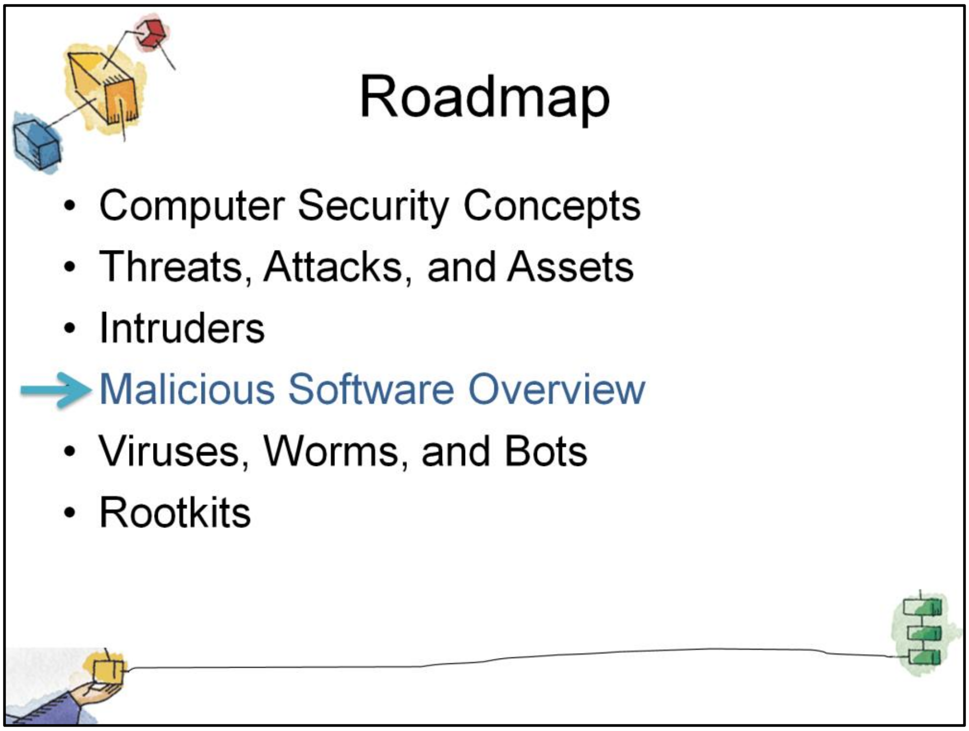
1. Act quickly and precisely to make their activities harder to detect.
2. Exploit perimeter through vulnerable ports.
3. Use Trojan horses (hidden software) to leave back doors for reentry.
4. Use sniffers to capture passwords.
5. Do not stick around until noticed.
6. Make few or no mistakes.



Intruder Behavior: Internal Threat

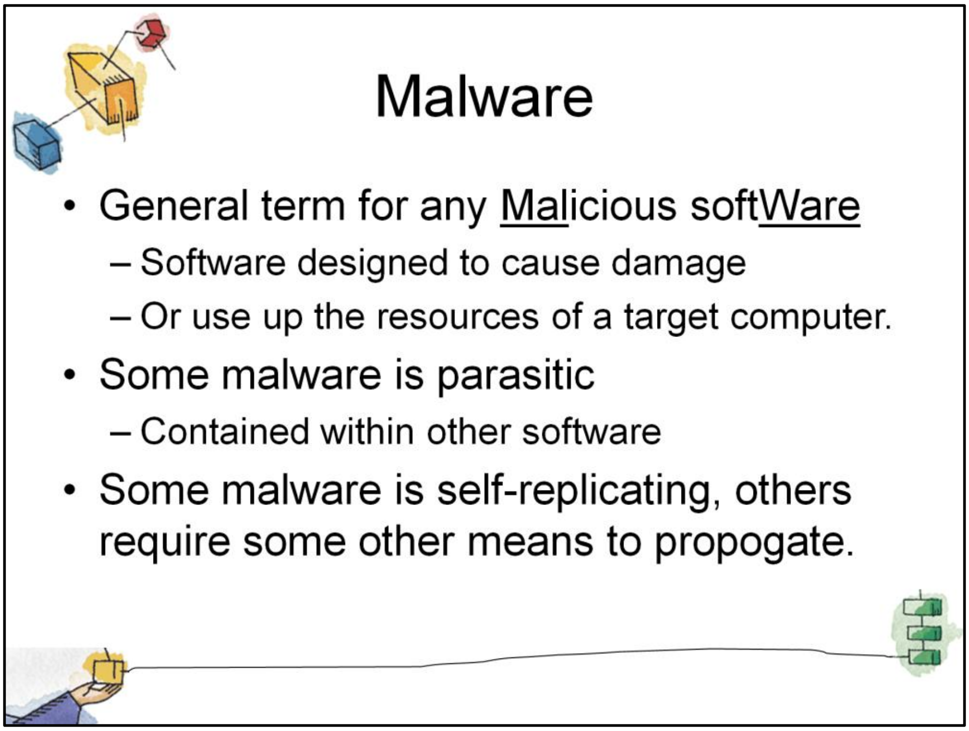
(c) Internal Threat

1. Create network accounts for themselves and their friends.
2. Access accounts and applications they wouldn't normally use for their daily jobs.
3. E-mail former and prospective employers.
4. Conduct furtive instant-messaging chats.
5. Visit Web sites that cater to disgruntled employees, such as f'dcompany.com.
6. Perform large downloads and file copying.
7. Access the network during off hours.



Roadmap

- Computer Security Concepts
- Threats, Attacks, and Assets
- Intruders
- → Malicious Software Overview
- Viruses, Worms, and Bots
- Rootkits



Malware

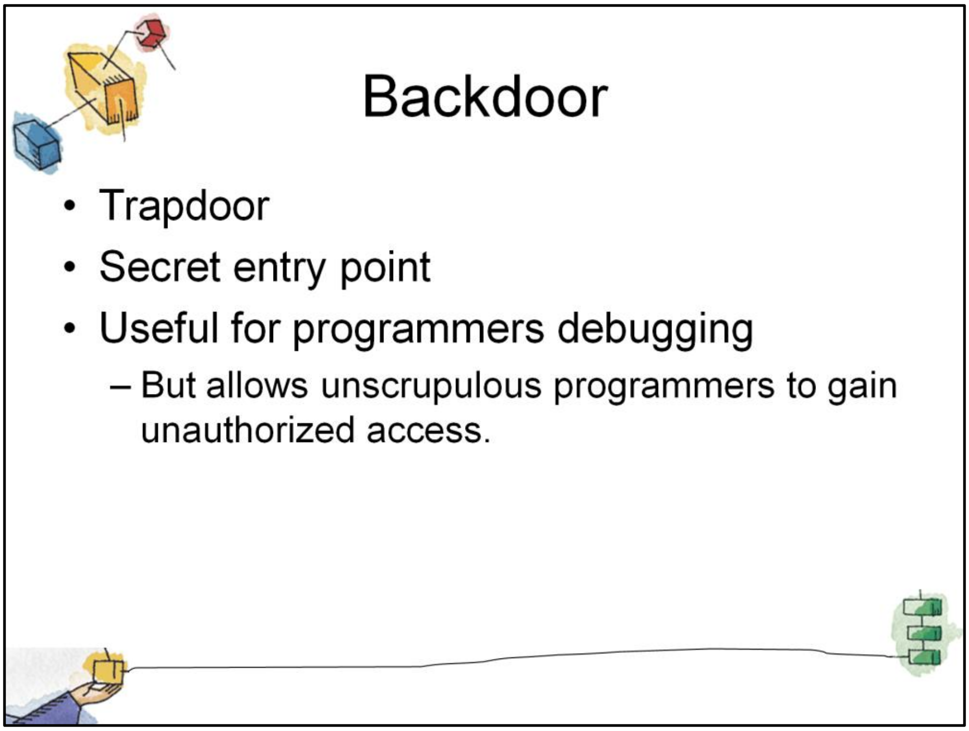
- General term for any Malicious softWare
  - Software designed to cause damage
  - Or use up the resources of a target computer.
- Some malware is parasitic
  - Contained within other software
- Some malware is self-replicating, others require some other means to propogate.

Perhaps the most sophisticated types of threats to computer systems are presented by programs that exploit vulnerabilities in computing systems.

- Such threats are referred to as malicious software, or malware.

Malware is software designed to cause damage to or use up the resources of a target computer.

- It is frequently concealed within or masquerades as legitimate software.
- In some cases, it spreads itself to other computers via e-mail or infected floppy disks.



Backdoor

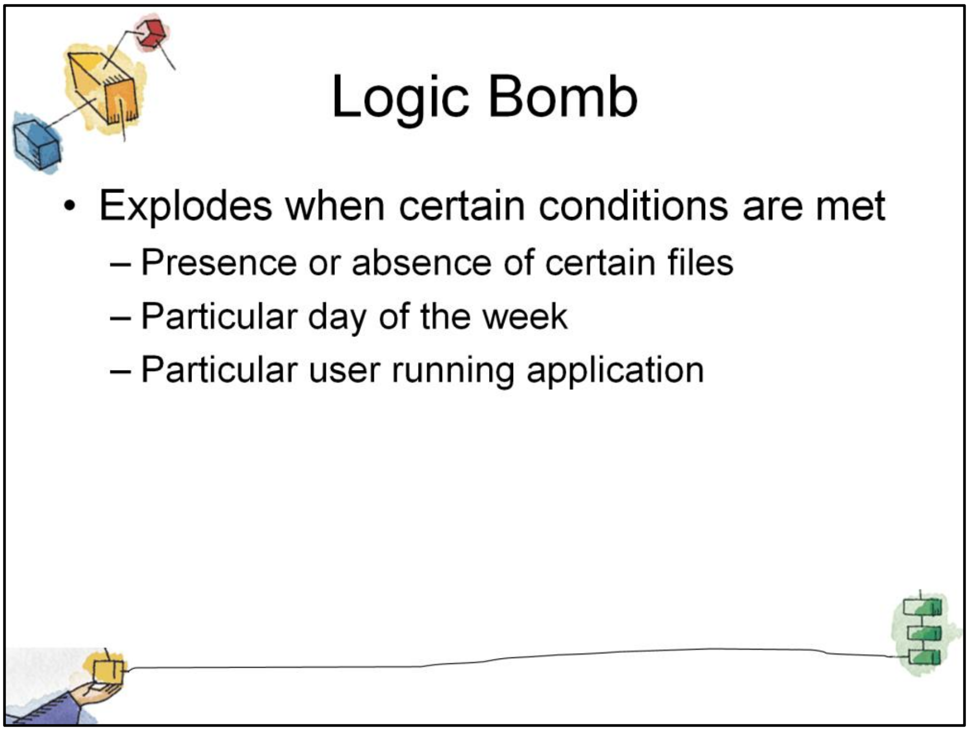
- Trapdoor
- Secret entry point
- Useful for programmers debugging
  - But allows unscrupulous programmers to gain unauthorized access.

A backdoor, also known as a trapdoor, is a secret entry point into a program that allows someone who is aware of the backdoor to gain access without going through the usual security access procedures.

Programmers have used backdoors legitimately for many years to debug and test programs;

- Called a maintenance hook.
- Usually is done when the programmer is developing an application that has an authentication procedure, or a long setup, requiring the user to enter many different values to run the application.

Backdoors become threats when unscrupulous programmers use them to gain unauthorized access.



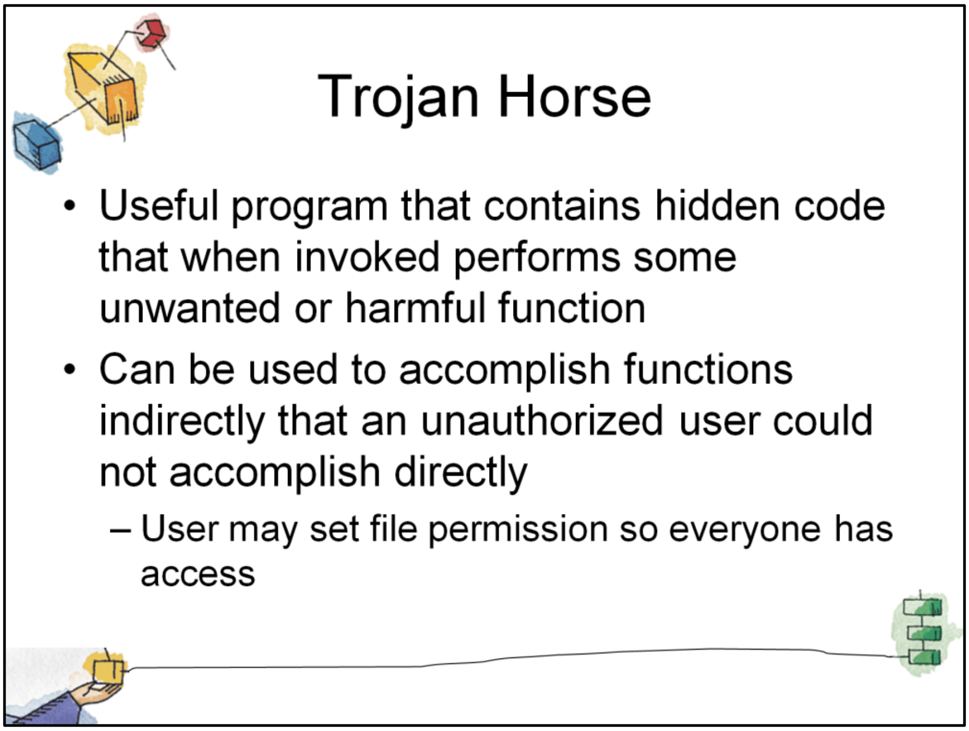
Logic Bomb

- Explodes when certain conditions are met
  - Presence or absence of certain files
  - Particular day of the week
  - Particular user running application

One of the oldest types of program threat, predating viruses and worms, is the logic bomb.

The logic bomb is code embedded in some legitimate program that is set to “explode” when certain conditions are met.

- Once triggered, a bomb may alter or delete data or entire files, cause a machine halt, or do some other damage.

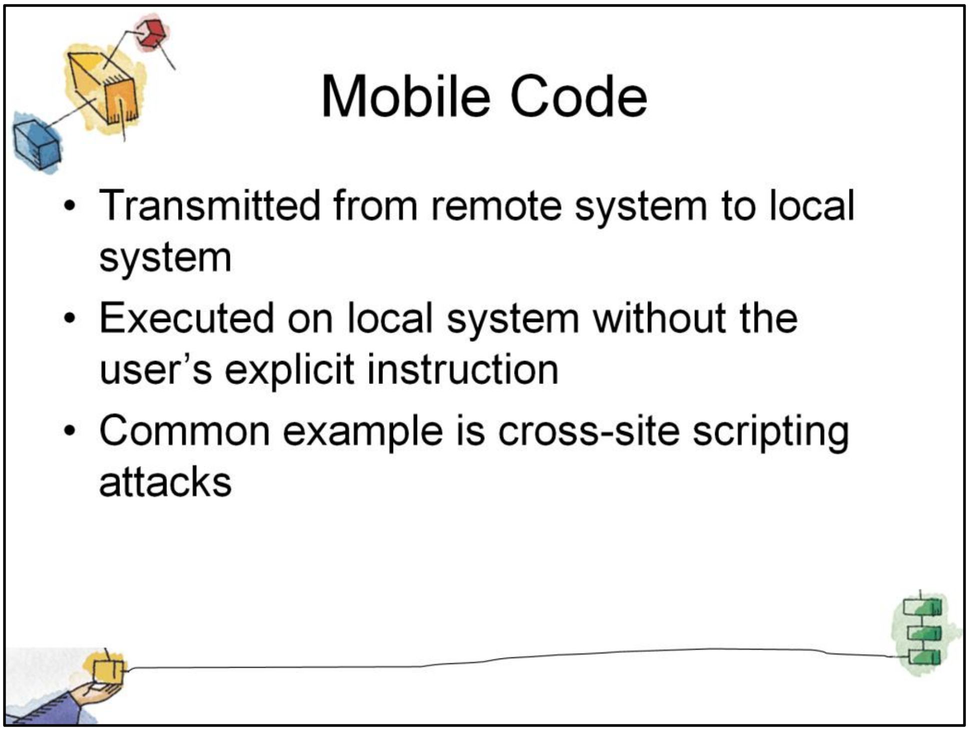


Trojan Horse

- Useful program that contains hidden code that when invoked performs some unwanted or harmful function
- Can be used to accomplish functions indirectly that an unauthorized user could not accomplish directly
  - User may set file permission so everyone has access

A Trojan horse is a useful, or apparently useful, program or command procedure containing hidden code that, when invoked, performs some unwanted or harmful function.

Trojan horse programs can be used to accomplish functions indirectly that an unauthorized user could not accomplish directly.

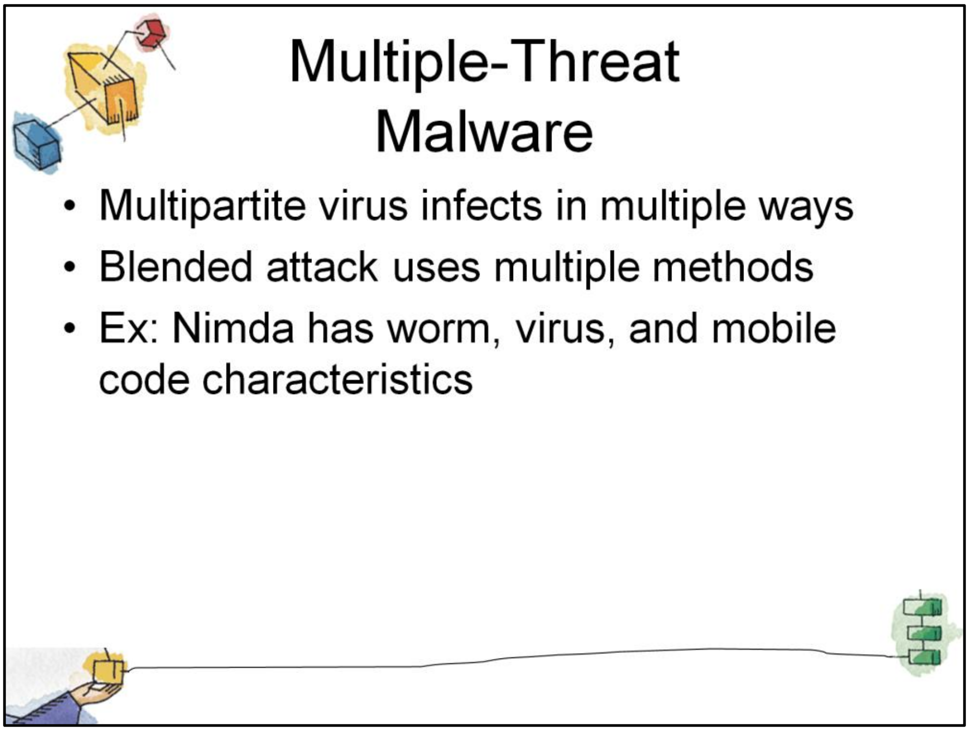


Mobile Code

- Transmitted from remote system to local system
- Executed on local system without the user’s explicit instruction
- Common example is cross-site scripting attacks

Mobile code refers to programs (e.g., script, macro, or other portable instruction) that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics.

Mobile code is transmitted from a remote system to a local system and then executed on the local system without the user’s explicit instruction.



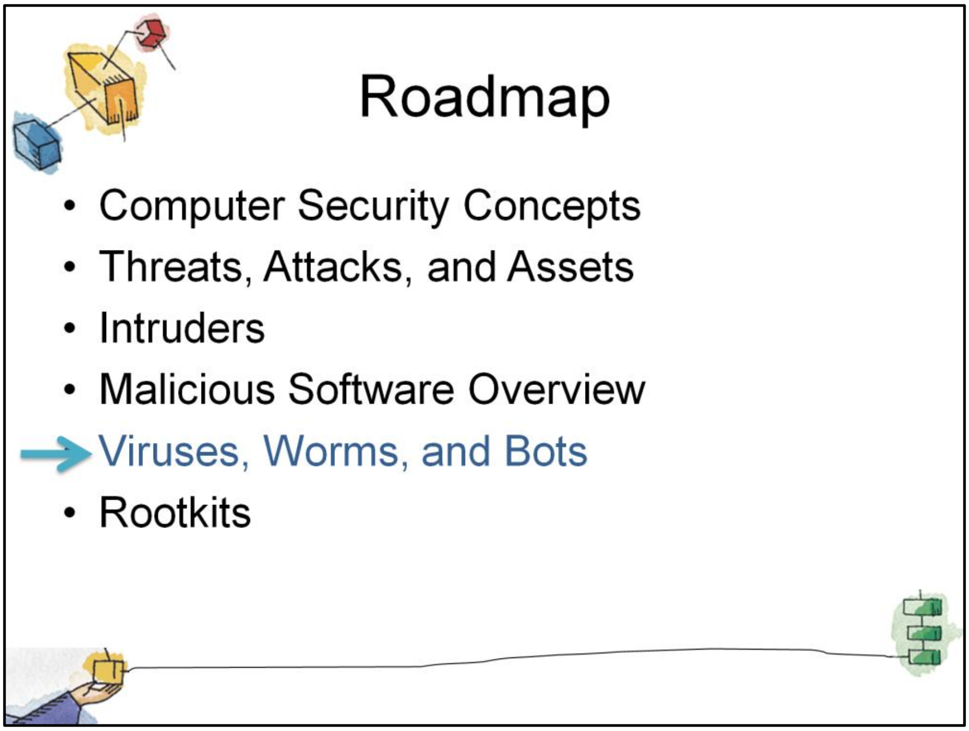
Multiple-Threat Malware

- Multipartite virus infects in multiple ways
- Blended attack uses multiple methods
- Ex: Nimda has worm, virus, and mobile code characteristics

A multipartite virus infects in multiple ways.

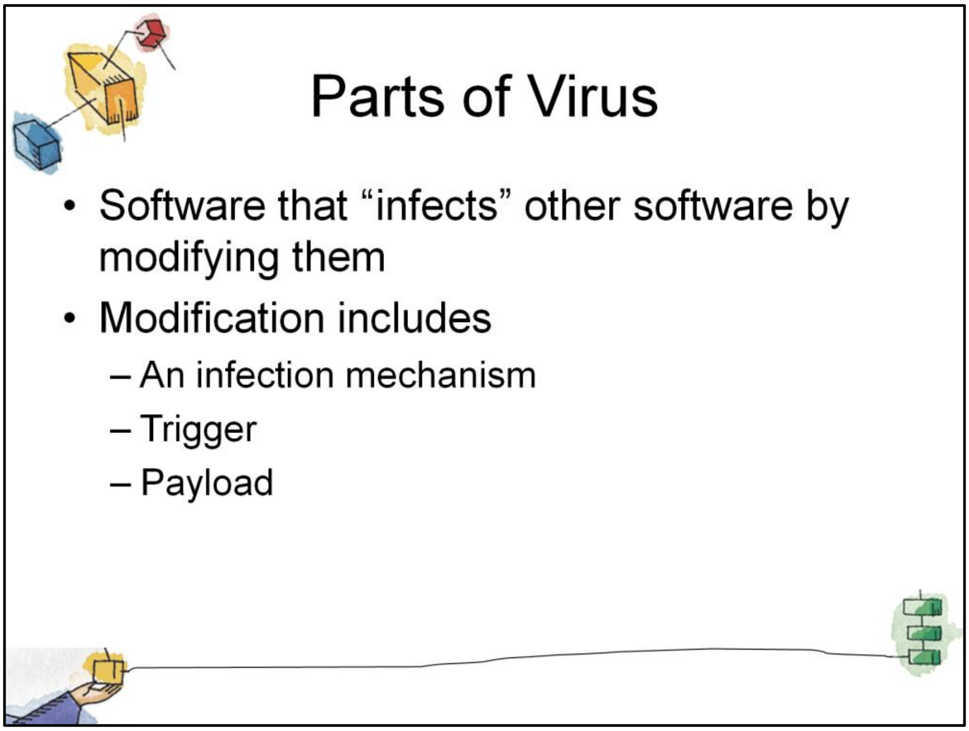
Typically, the multipartite virus is capable of infecting multiple types of files, so that virus eradication must deal with all of the possible sites of infection.

A blended attack uses multiple methods of infection or transmission, to maximize the speed of contagion and the severity of the attack.



Roadmap

- Computer Security Concepts
- Threats, Attacks, and Assets
- Intruders
- Malicious Software Overview
- → Viruses, Worms, and Bots
- Rootkits



Parts of Virus

- Software that “infects” other software by modifying them
- Modification includes
  - An infection mechanism
  - Trigger
  - Payload

A computer virus is a piece of software that can “infect” other programs by modifying them;

- The modification includes injecting the original program with a routine to make copies of the virus program, which can then go on to infect other programs.

A computer virus carries in its instructional code the recipe for making perfect copies of itself.

- The typical virus becomes embedded in a program on a computer.
- Then, whenever the infected computer comes into contact with an uninfected piece of software, a fresh copy of the virus passes into the new program.

A computer virus has three parts:

Infection mechanism:

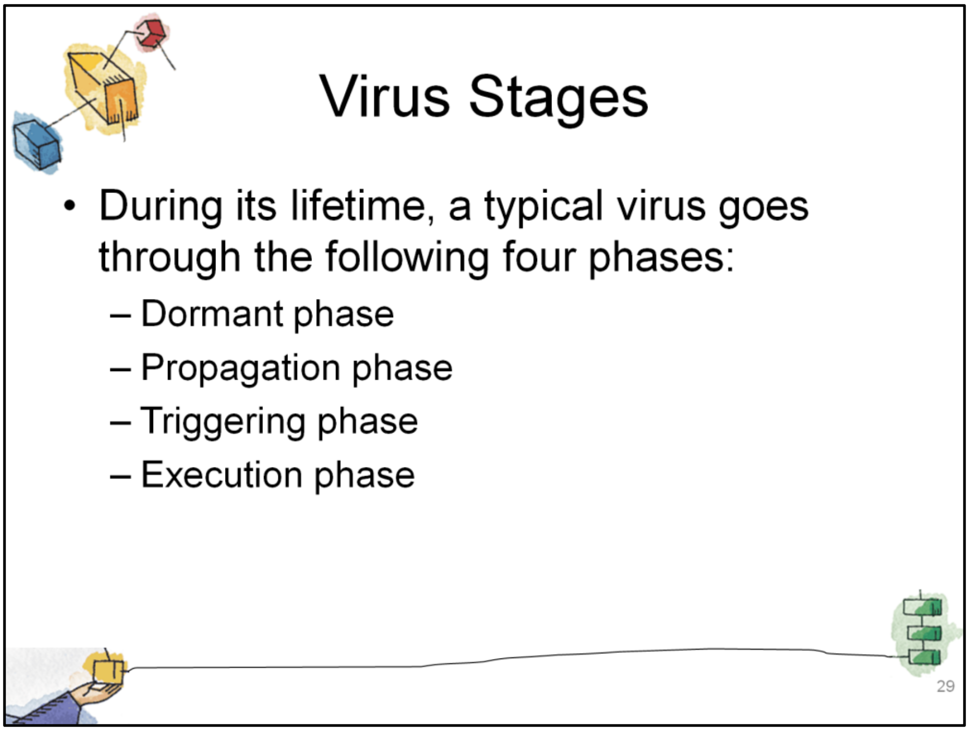
- The means by which a virus spreads, enabling it to replicate.
- The mechanism is also referred to as the infection vector.

Trigger:

- The event or condition that determines when the payload is activated or delivered.

Payload:

- What the virus does, besides spreading.
- The payload may involve damage or may involve benign but noticeable activity.



Virus Stages

- During its lifetime, a typical virus goes through the following four phases:
  - Dormant phase
  - Propagation phase
  - Triggering phase
  - Execution phase

During its lifetime, a typical virus goes through the following four phases:

Dormant phase:

- The virus is idle.
- The virus will eventually be activated by some event, such as a date, the presence of another program or file, or the capacity of the disk exceeding some limit.
- Not all viruses have this stage.

Propagation phase:

- The virus places an identical copy of itself into other programs or into certain system areas on the disk.
- Each infected program will now contain a clone of the virus, which will itself enter a propagation phase.

Triggering phase:

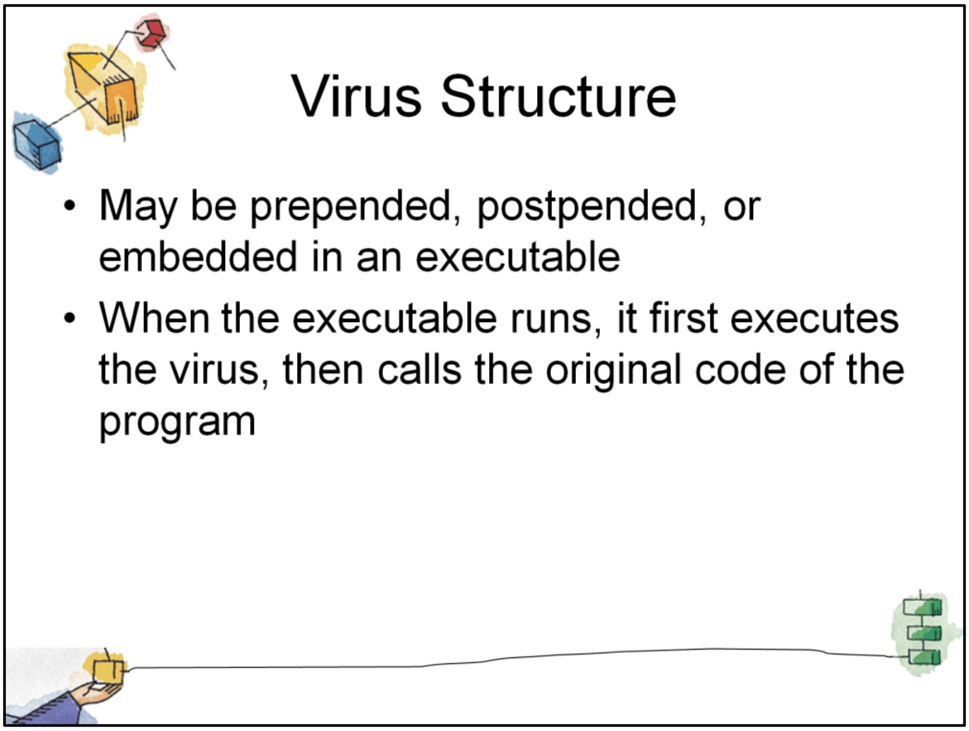
- The virus is activated to perform the function for which it was intended.
- As with the dormant phase, the triggering phase can be caused by a variety of system events, including a count of the number of times that this copy of the virus has made copies of itself.

Execution phase:

- The function is performed.
- The function may be harmless, such as a message on the screen, or damaging, such as the destruction of programs and data files.

Most viruses carry out their work in a manner that is specific to a particular operating system and, in some cases, specific to a particular hardware platform.

- Thus, they are designed to take advantage of the details and weaknesses of particular systems.

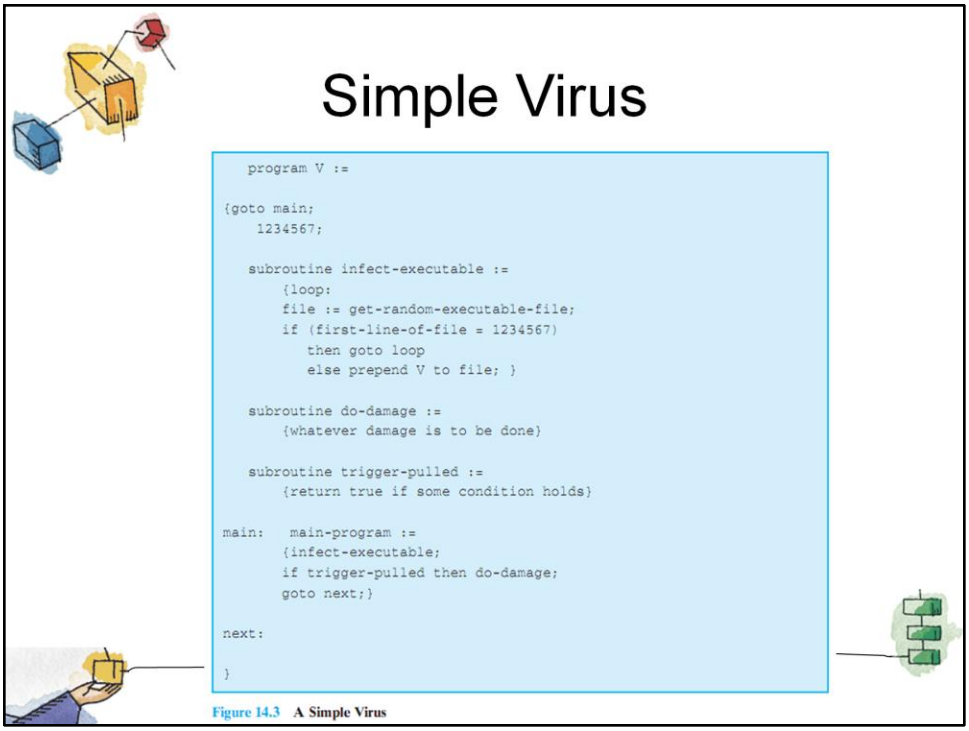


Virus Structure

- May be prepended, postpended, or embedded in an executable
- When the executable runs, it first executes the virus, then calls the original code of the program

A virus can be prepended or postpended to an executable program, or it can be embedded in some other fashion.

The key to its operation is that the infected program, when invoked, will first execute the virus code and then execute the original code of the program.



Simple Virus

```
    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:

}
```

Figure 14.3 A Simple Virus

A very general depiction of virus structure is shown here.

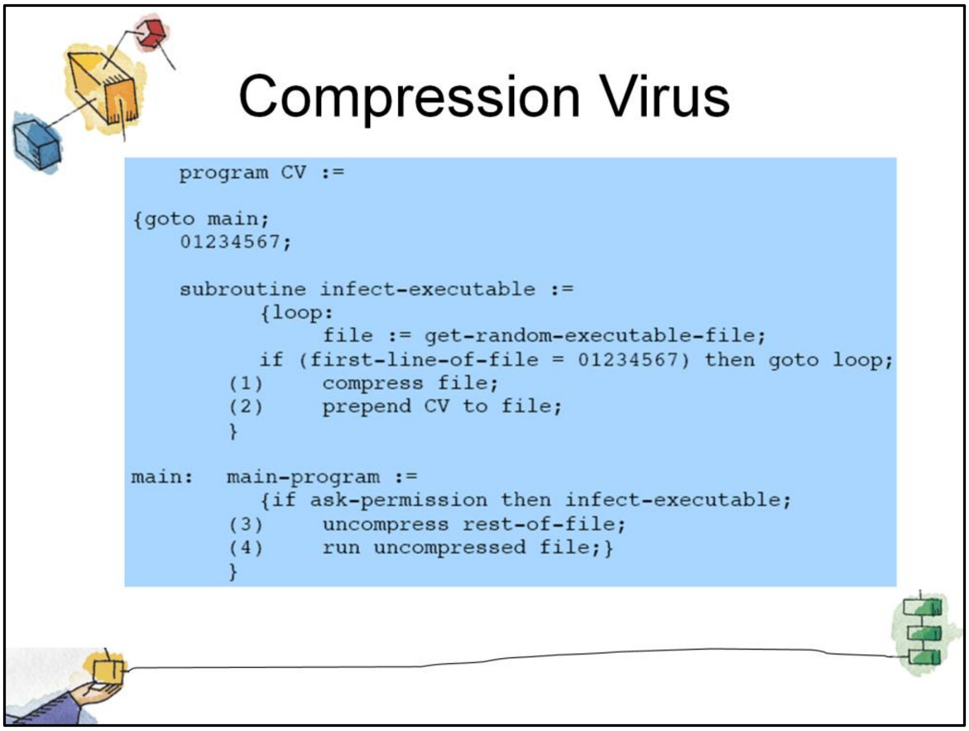
In this case, the virus code, V, is prepended to infected programs,

- it is assumed that the entry point to the program, when invoked, is the first line of the program.

The infected program begins with the virus code and works as follows.

- The first line of code is a jump to the main virus program.
- The second line is a special marker that is used by the virus to determine whether or not a potential victim program has already been infected with this virus.
- When the program is invoked, control is immediately transferred to the main virus program.
- The virus program may first seek out uninfected executable files and infect them.
- Next, the virus may perform some action, usually detrimental to the system.
    - This action could be performed every time the program is invoked, or it could be a logic bomb that triggers only under certain conditions.
- Finally, the virus transfers control to the original program.

If the infection phase of the program is reasonably rapid, a user is unlikely to notice any difference between the execution of an infected and an uninfected program.



Compression Virus

```
    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;}
       }
```

A virus such as the one just described is easily detected because an infected version of a program is longer than the corresponding uninfected one.

A way to thwart such a simple means of detecting a virus is to compress the executable file so that both the infected and uninfected versions are of identical length.

This figure shows in general terms the logic required.

We assume that program P1 is infected with the virus CV.

- When this program is invoked, control passes to its virus, which performs the following steps:

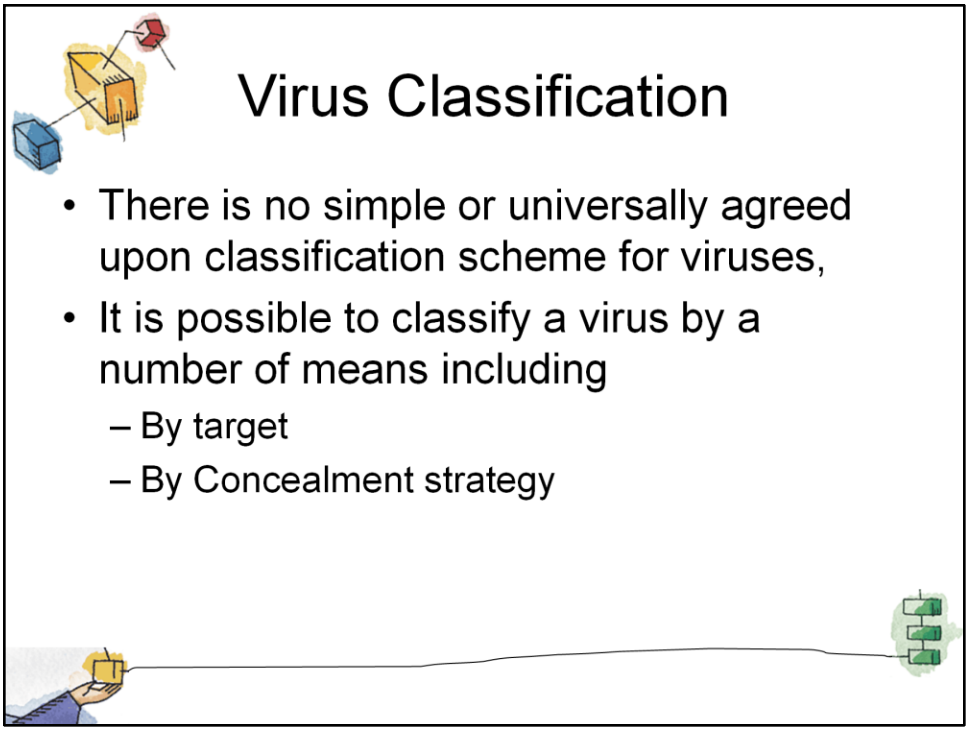
1. For each uninfected file P2 that is found, the virus first compresses that file to produce P' 2, which is shorter than the original program by the size of the virus.

2. A copy of the virus is prepended to the compressed program.

3. The compressed version of the original infected program, P9 1, is uncompressed.

4. The uncompressed original program is executed.

- In this example, the virus does nothing other than propagate, it could include a logic bomb.



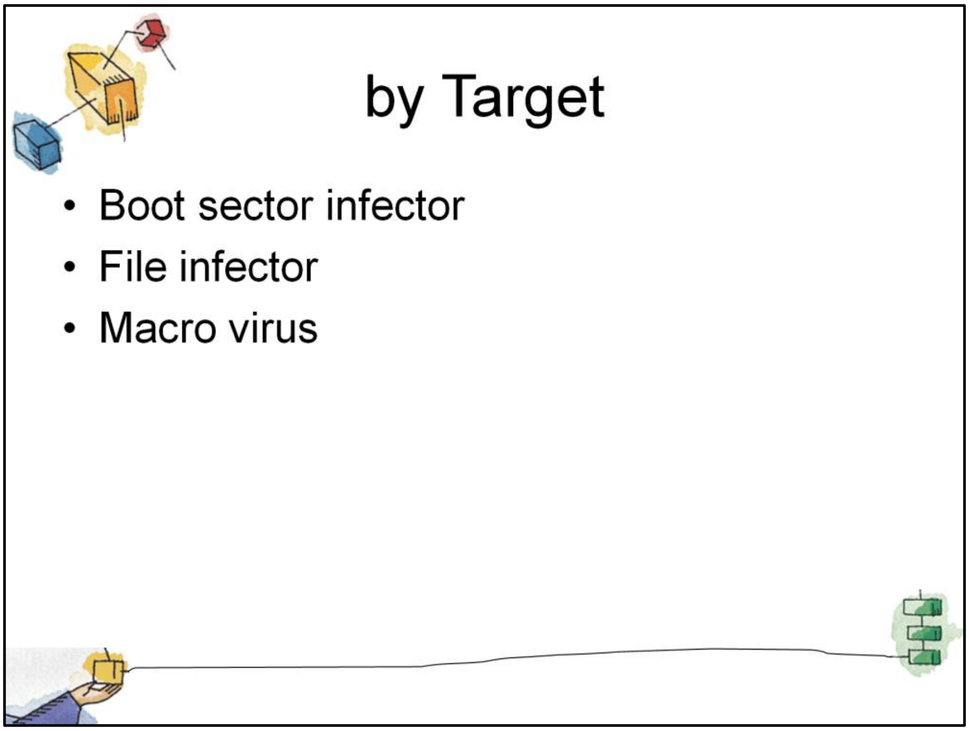
Virus Classification

- There is no simple or universally agreed upon classification scheme for viruses,
- It is possible to classify a virus by a number of means including
  - By target
  - By Concealment strategy

There is no simple or universally agreed upon classification scheme for viruses.

We follow a method to classify viruses along two orthogonal axes:

- the type of target the virus tries to infect and
- the method the virus uses to conceal itself from detection by users and antivirus software.



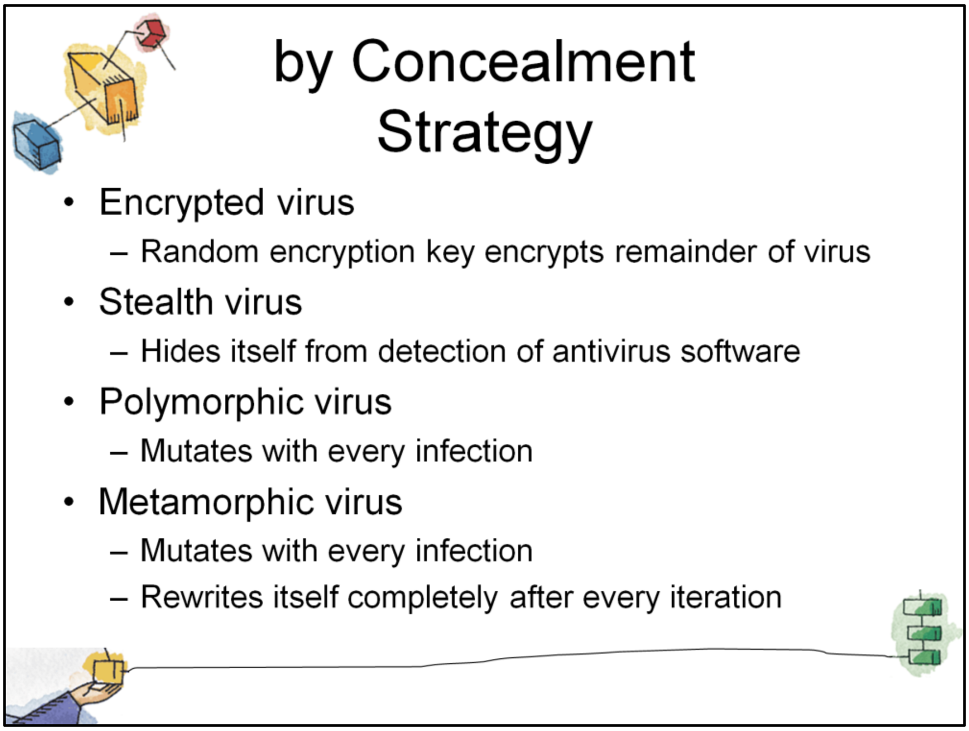
by Target

- Boot sector infector
- File infector
- Macro virus

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 consider to be executable

Macro virus: Infects files with macro code that is interpreted by an application



by Concealment Strategy

- Encrypted virus
  - – Random encryption key encrypts remainder of virus
- Stealth virus
  - – Hides itself from detection of antivirus software
- Polymorphic virus
  - – Mutates with every infection
- Metamorphic virus
  - – Mutates with every infection
  - – Rewrites itself completely after every iteration

A virus classification by concealment strategy includes the following categories:

Encrypted virus:

- A portion of the virus creates a random encryption key and encrypts the remainder of the virus.
- The key is stored with the virus.
- When an infected program is invoked, the virus uses the stored random key to decrypt the virus.
- When the virus replicates, a different random key is selected.
- Because the bulk of the virus is encrypted with a different key for each instance, there is no constant bit pattern to observe.

Stealth virus:

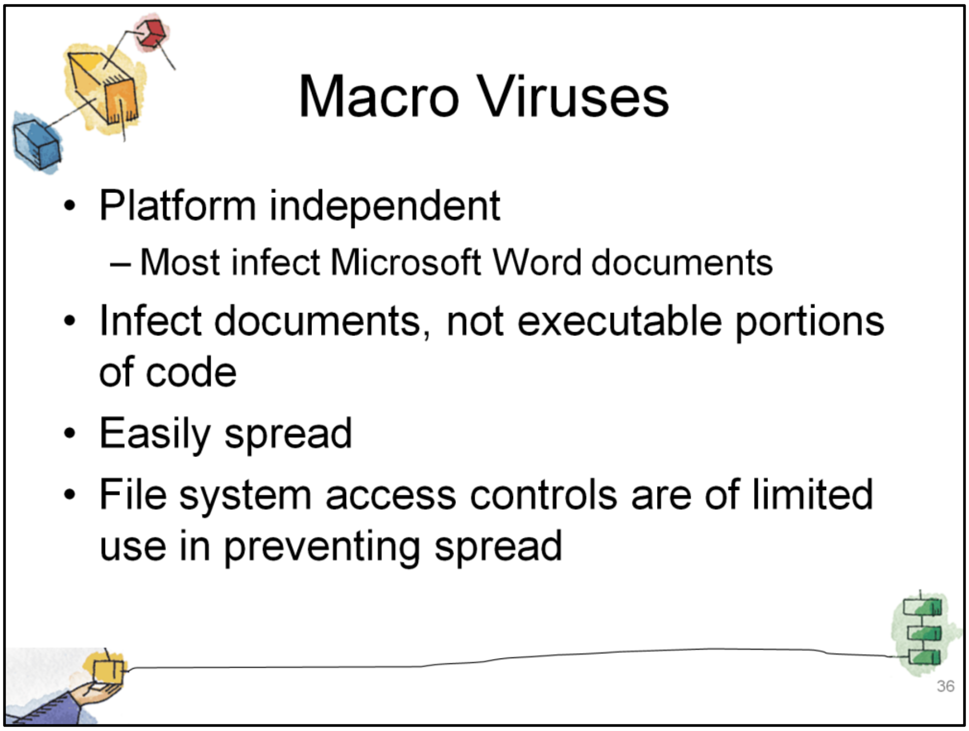
- A form of virus explicitly designed to hide itself from detection by antivirus software.
- Thus, the entire virus, not just a payload, is hidden.

Polymorphic virus:

- A virus that mutates with every infection, making detection by the "signature" of the virus impossible.

Metamorphic virus:

- As with a polymorphic virus, a metamorphic virus mutates with every infection.
- The difference is that a metamorphic virus rewrites itself completely at each iteration, increasing the difficulty of detection.
- Metamorphic viruses may change their behavior as well as their appearance.



Macro Viruses

- Platform independent
  - Most infect Microsoft Word documents
- Infect documents, not executable portions of code
- Easily spread
- File system access controls are of limited use in preventing spread

In the mid-1990s, macro viruses became by far the most prevalent type of virus.

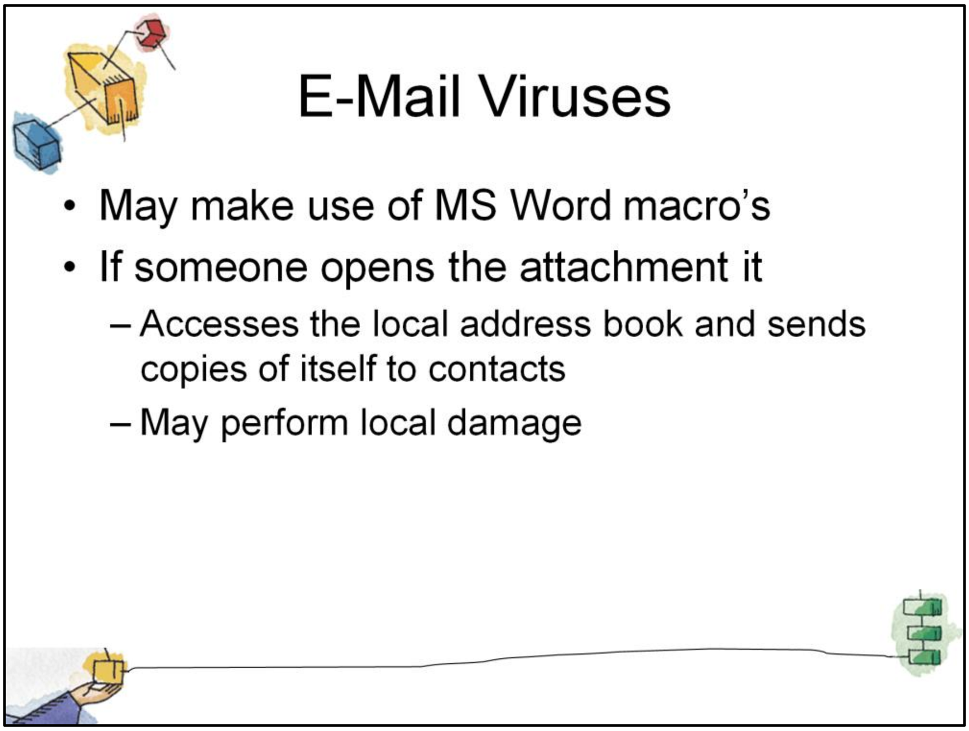
Macro viruses are particularly threatening for a number of reasons:

1. A macro virus is platform independent. Many macro viruses infect Microsoft Word documents or other Microsoft Office documents.
   - Any hardware platform and operating system that supports these applications can be infected.

2. Macro viruses infect documents, not executable portions of code.
   - Most of the information introduced onto a computer system is in the form of a document rather than a program.

3. Macro viruses are easily spread.
   - A very common method is by electronic mail.

4. Because macro viruses infect user documents rather than system programs, traditional file system access controls are of limited use in preventing their spread.



E-Mail Viruses

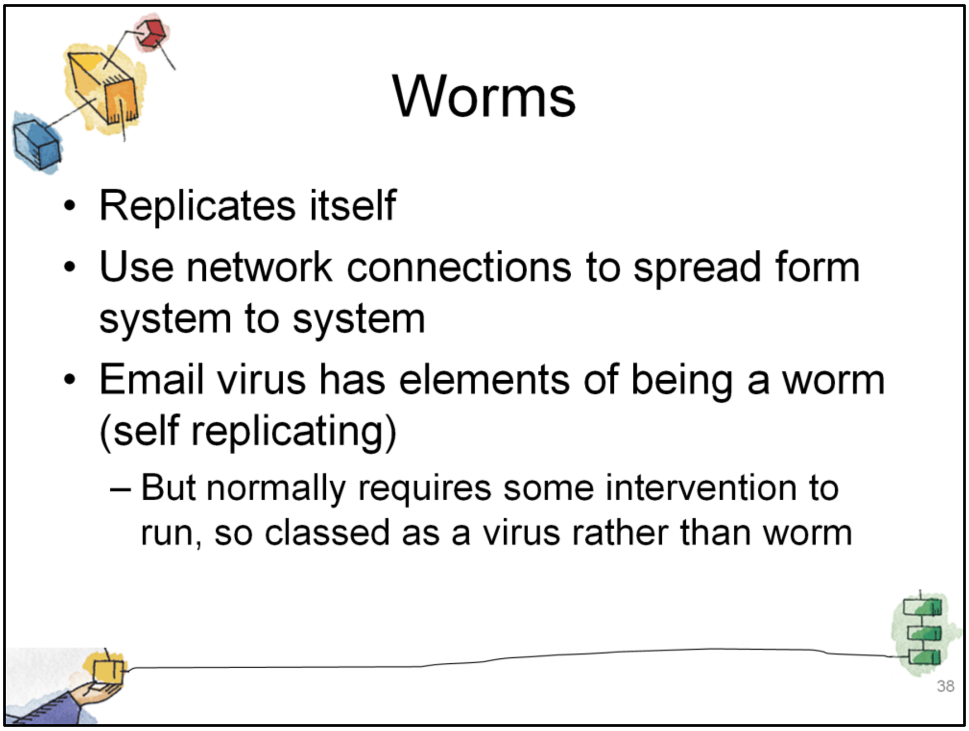
- May make use of MS Word macro's
- If someone opens the attachment it
  - Accesses the local address book and sends copies of itself to contacts
  - May perform local damage

A more recent development in malicious software is the e-mail virus.

The first rapidly spreading e-mail viruses, such as Melissa, made use of a Microsoft Word macro embedded in an attachment.

If the recipient opens the e-mail attachment, the Word macro is activated. Then

1. The e-mail virus sends itself to everyone on the mailing list in the user's e-mail package.
2. The virus does local damage on the user's system.



Worms

- Replicates itself
- Use network connections to spread form system to system
- Email virus has elements of being a worm (self replicating)
  - But normally requires some intervention to run, so classed as a virus rather than worm

A worm is a program that can replicate itself and send copies from computer to computer across network connections.

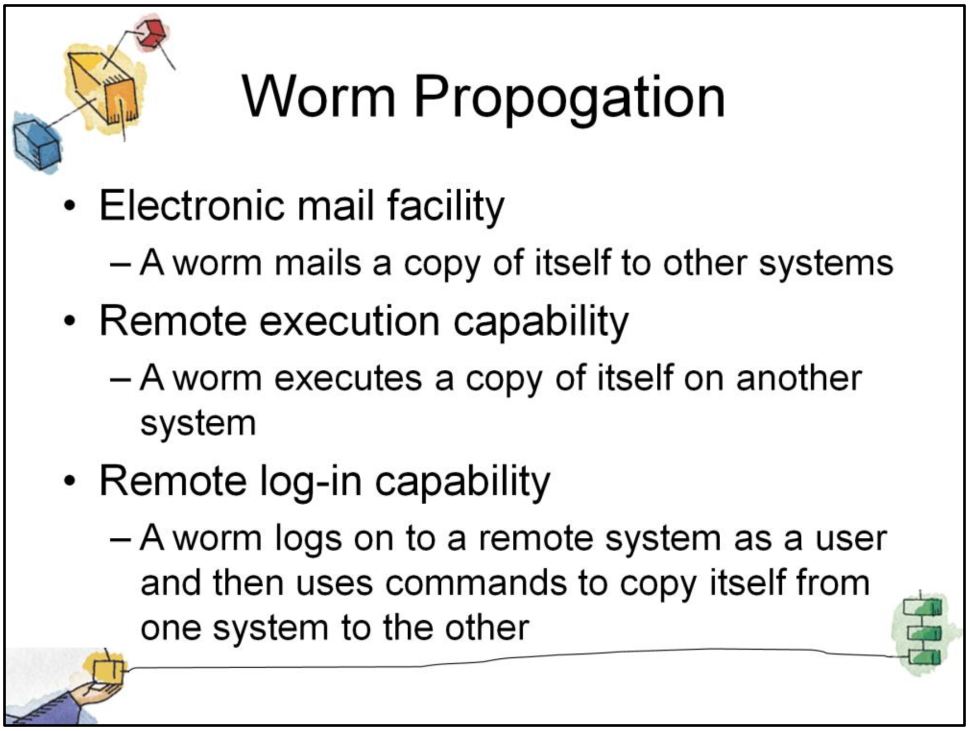
- Upon arrival, the worm may be activated to replicate and propagate again.

In addition to propagation, the worm usually performs some unwanted function.

- An e-mail virus has some of the characteristics of a worm because it propagates itself from system to system.
- However, we can still classify it as a virus because it uses a document modified to contain viral macro content and requires human action.

A worm actively seeks out more machines to infect and each machine that is infected serves as an automated launching pad for attacks on

other machines.



Worm Propogation

- Electronic mail facility
  - A worm mails a copy of itself to other systems
- Remote execution capability
  - A worm executes a copy of itself on another system
- Remote log-in capability
  - A worm logs on to a remote system as a user and then uses commands to copy itself from one system to the other

To replicate itself, a network worm uses some sort of network vehicle.

Examples include the following:

Electronic mail facility:

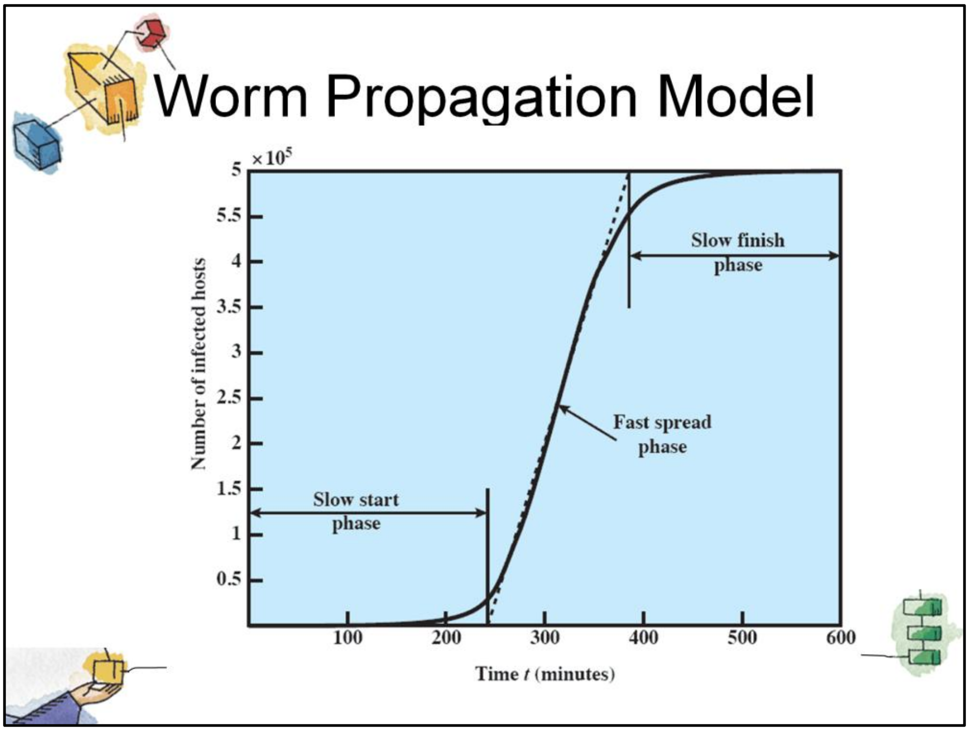
- A worm mails a copy of itself to other systems, so that its code is run when the e-mail or an attachment is received or viewed.

Remote execution capability:

- A worm executes a copy of itself on another system, either using an explicit remote execution facility or by exploiting a program flaw in a network service to subvert its operations.

Remote login capability:

- A worm logs onto a remote system as a user and then uses commands to copy itself from one system to the other, where it then executes.



Worm Propagation Model

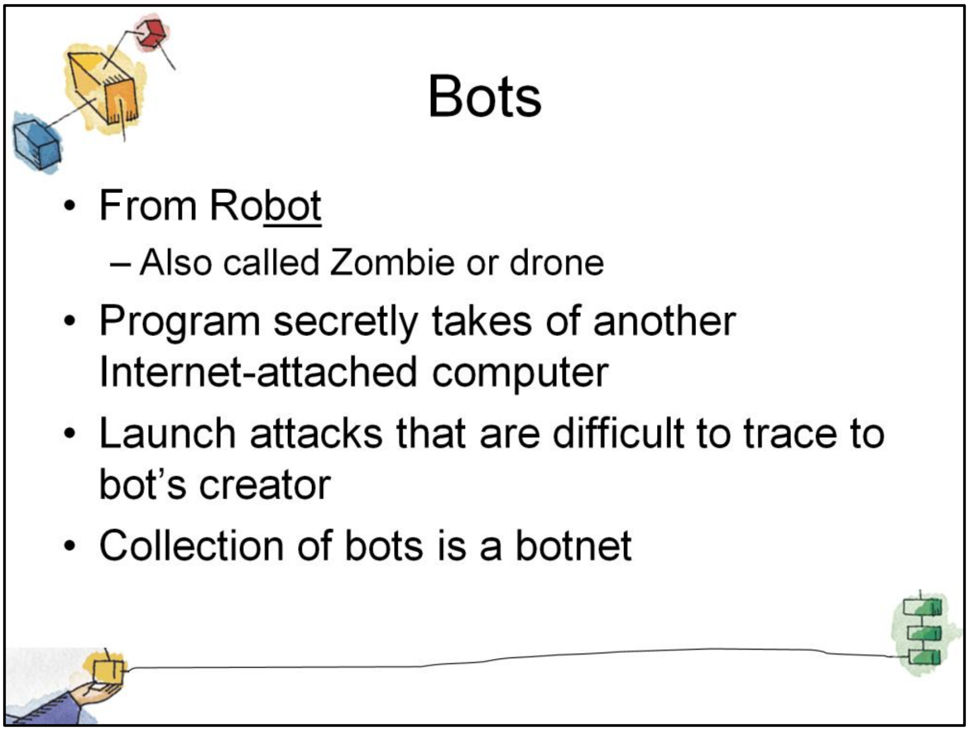
The speed of propagation and the total number of hosts infected depend on a number of factors, including the mode of propagation, the vulnerability or vulnerabilities exploited, and the degree of similarity to preceding attacks.

- For the latter factor, an attack that is a variation of a recent previous attack may be countered more effectively than a more novel attack.

This figure shows the dynamics for one typical set of parameters.

Propagation proceeds through three phases.

1. In the initial phase, the number of hosts increases exponentially.
    - e.g. consider a simplified case in which a worm is launched from a single host and infects two nearby hosts.
    - Each of these hosts infects two more hosts, and so on.
2. This results in exponential growth.
3. After a time, infecting hosts waste some time attacking already infected hosts, which reduces the rate of infection.
    - During this middle phase, growth is approximately linear, but the rate of infection is rapid.
    - When most vulnerable computers have been infected, the attack enters a slow finish phase as the worm seeks out those remaining hosts that are difficult to identify.



Bots

- From Robot
  - Also called Zombie or drone
- Program secretly takes of another Internet-attached computer
- Launch attacks that are difficult to trace to bot’s creator
- Collection of bots is a botnet

A bot (robot), also known as a zombie or drone, is a program that secretly takes over another Internet-attached computer and then uses that computer to launch attacks that are difficult to trace to the bot’s creator.

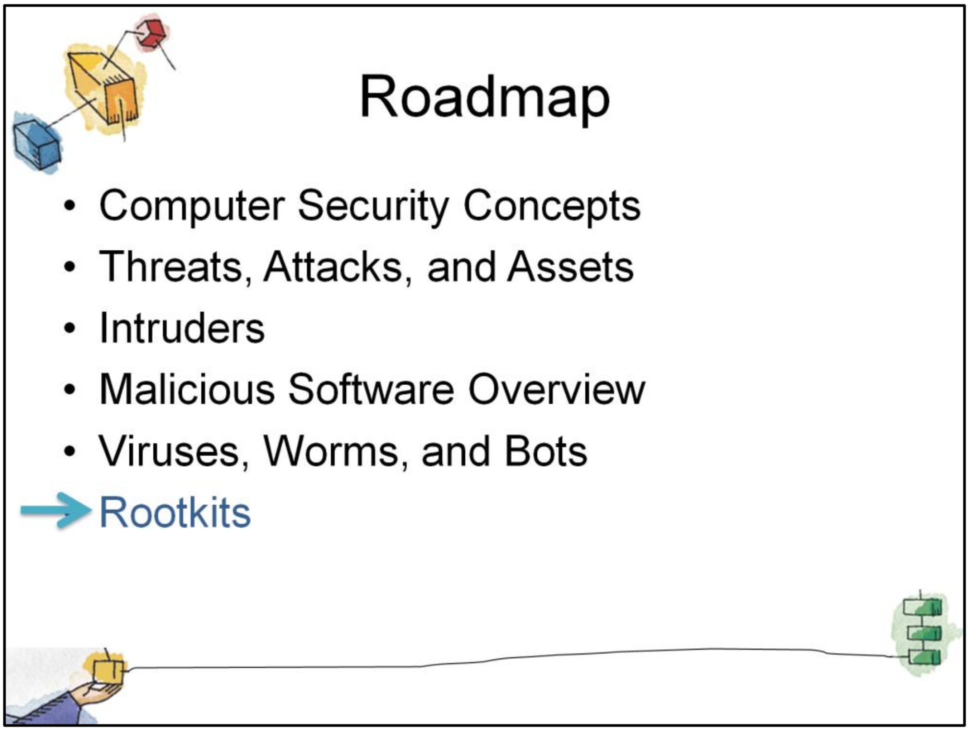
The bot is typically planted on hundreds or thousands of computers belonging to unsuspecting third parties.

The collection of bots often is capable of acting in a coordinated manner;

- such a collection is referred to as a botnet.

A botnet exhibits three characteristics:

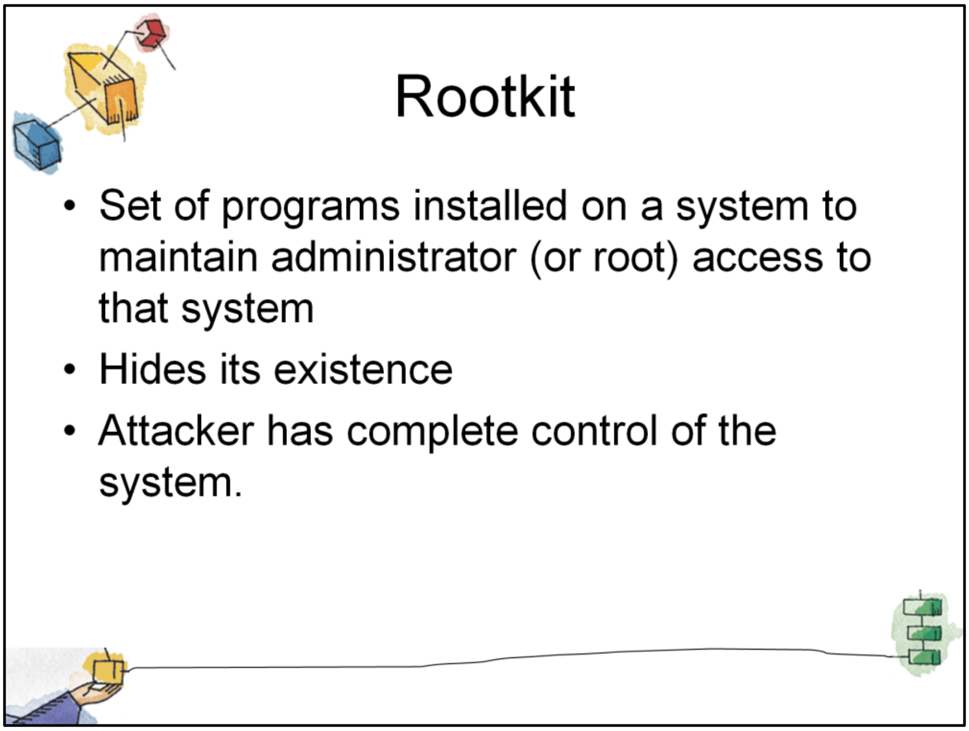
- the bot functionality,
- a remote control facility, and
- a spreading mechanism to propagate the bots and construct the botnet.



Roadmap

- Computer Security Concepts
- Threats, Attacks, and Assets
- Intruders
- Malicious Software Overview
- Viruses, Worms, and Bots

→ Rootkits



Rootkit

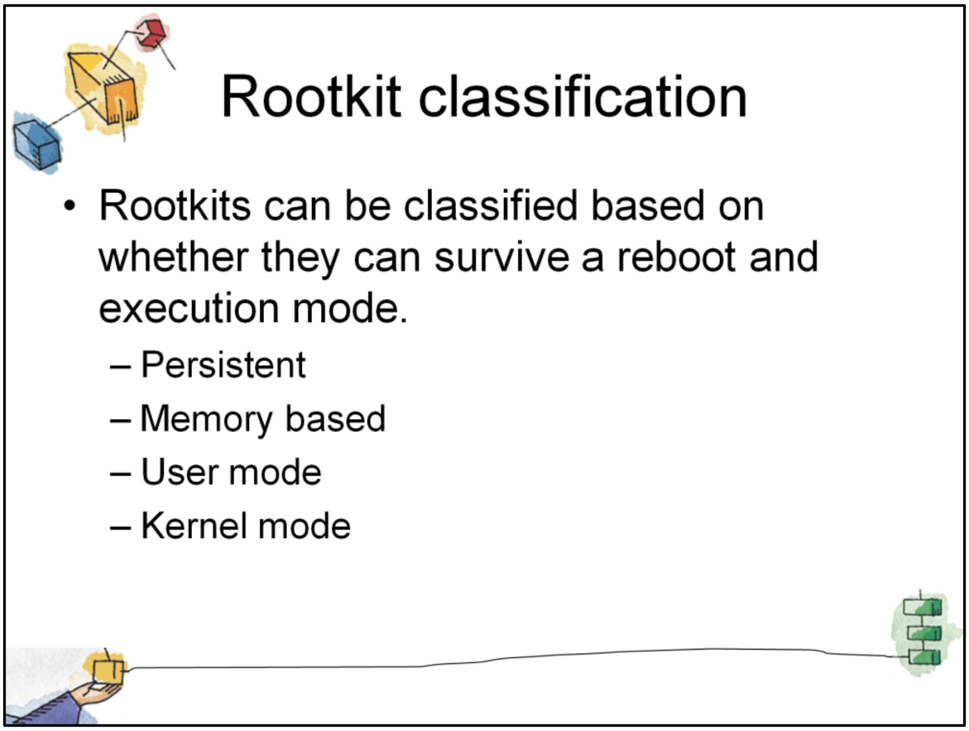
- Set of programs installed on a system to maintain administrator (or root) access to that system
- Hides its existence
- Attacker has complete control of the system.

A rootkit is a set of programs installed on a system to maintain administrator (or root) access to that system.

Root access provides access to all the functions and services of the operating system.

The rootkit alters the host’s standard functionality in a malicious and stealthy way.

With root access, an attacker has complete control of the system and can add or changes programs and files, monitor processes, send and
receive network traffic, and get backdoor access on demand.



Rootkit classification

- Rootkits can be classified based on whether they can survive a reboot and execution mode.
  - Persistent
  - Memory based
  - User mode
  - Kernel mode

Rootkits can be classified based on whether they can survive a reboot and execution mode.

A rootkit may be

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.

Memory based:

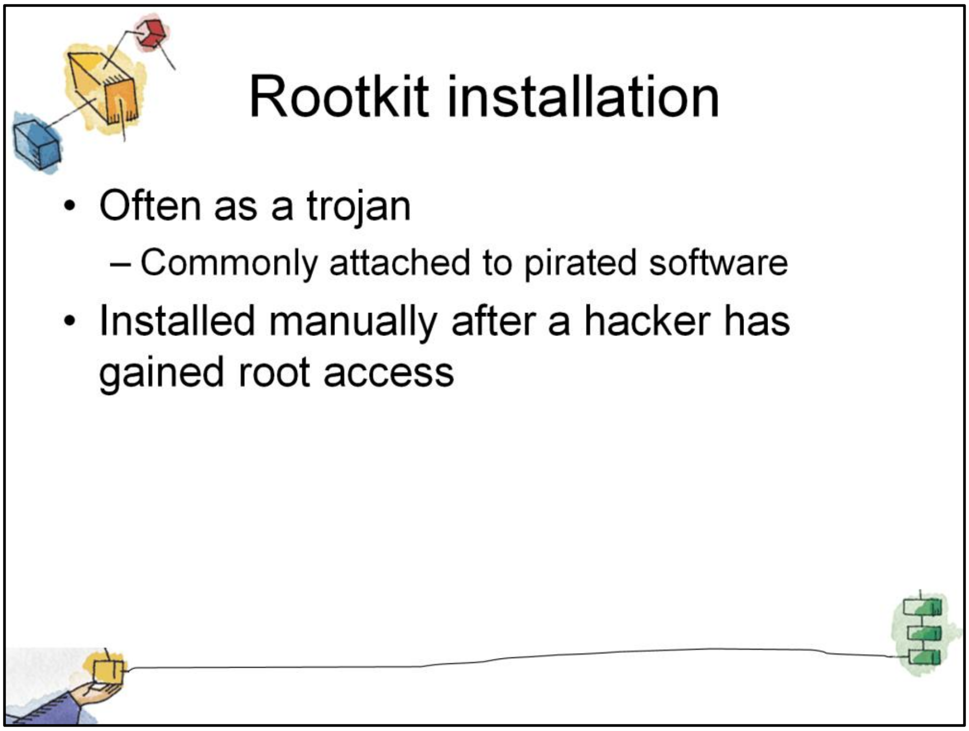
- Has no persistent code and therefore cannot survive a reboot.

User mode:

- Intercepts calls to APIs (application program interfaces) and modifies returned results.
- E.g. 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.



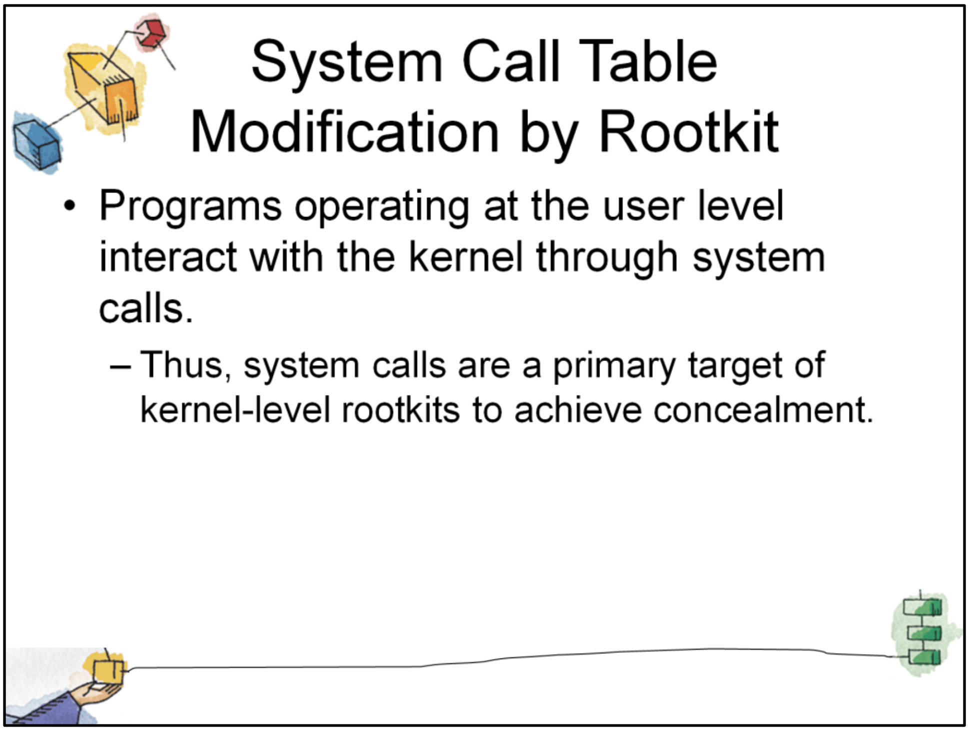
Rootkit installation

- Often as a trojan
  - Commonly attached to pirated software
- Installed manually after a hacker has gained root access

Unlike worms or bots, rootkits do not directly rely on vulnerabilities or exploits to get on a computer.

- One method of rootkit installation is via a Trojan horse program.
- The user is induced to load the Trojan horse,which then installs the rootkit.

Another means of rootkit installation is by hacker activity.



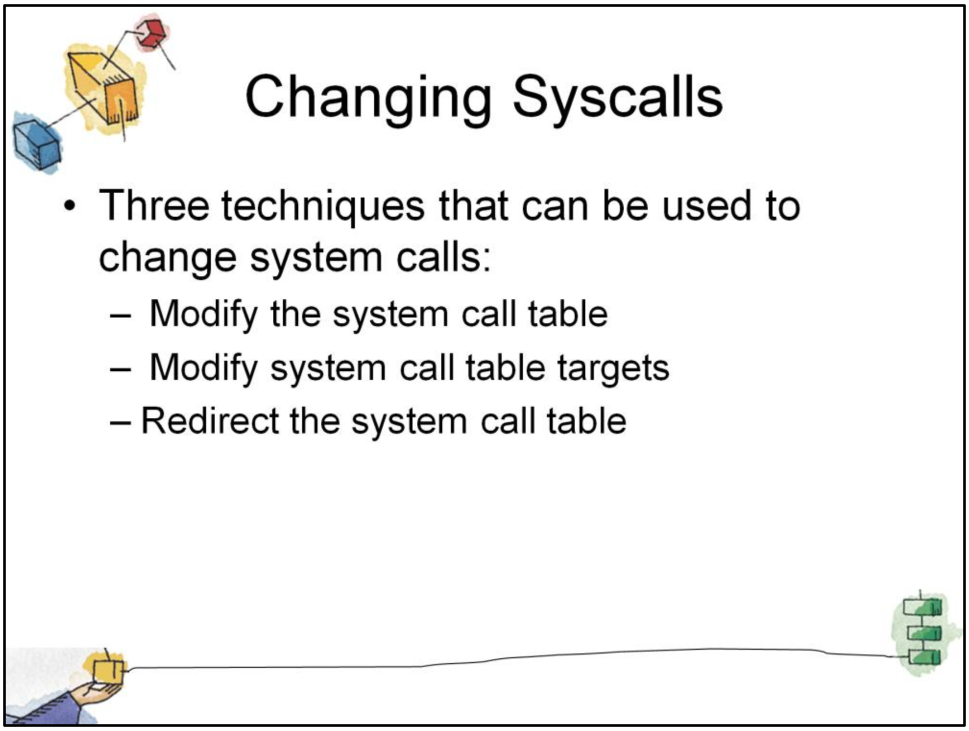
System Call Table Modification by Rootkit

- 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.

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.

- 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.



Changing Syscalls

- Three techniques that can be used to change system calls:
  - Modify the system call table
  - Modify system call table targets
  - Redirect the system call table

Three techniques that can be used to change system calls:

1. **Modify the system call table:**

   • The attacker modifies selected syscall addresses stored in the system call table.

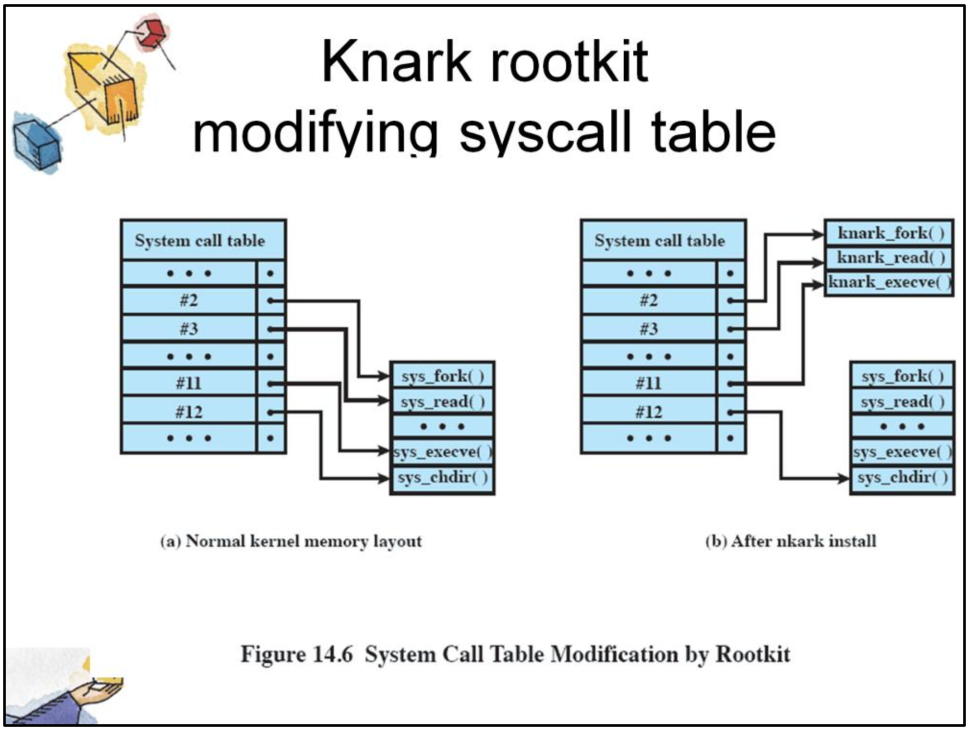
   • This enables the rootkit to direct a system call away from the legitimate routine to the rootkit's replacement.

2. **Modify system call table targets:**

   - The attacker overwrites selected legitimate system call routines with malicious code.
   - The system call table is not changed.

3. **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.



Knark rootkit modifying syscall table

(a) Normal kernel memory layout

(b) After nkark install

Figure 14.6 System Call Table Modification by Rootkit

This figure shows how the knark rootkit achieves this by modifying the system call table