Malware

Dr. Natarajan Meghanathan
Associate Professor of Computer Science
Jackson State University
E-mail: natarajan.meghanathan@jsums.edu

Malware: Malicious Software

- Malware is the general name given for unanticipated or undesired effects in programs or program parts caused by an agent with an intent to damage.
- Malicious code can do anything any other program can.
  - The malicious code can write message to the computer screen, stop running a program, generate sound, or erase a file
  - The malicious code can remain dormant until some event triggers the code to act. The triggers could be:
    - A certain time or date
    - A time interval (for example, after 30 minutes)
    - An event (for example, whenever a particular program is executed)
    - A condition (for example, when communication occurs over the network interface)
    - A count (for example, the fifth time something happens)
- Malicious code runs under the user's authority. So the malicious code can read, write, modify, append and delete data files, similar to what an authorized user could do. But malicious code do all these things without the user's permission or knowledge.

List of Malware we will Study

- Virus
- Trojan Horse
- Worm
- Rootkit
- Botnets

Virus

- A virus is a program that can replicate itself and pass on malicious code to other non-malicious programs by modifying them.
- The infected program becomes acting like a virus and infects other good programs.
- The infection spreads in a geometrical rate, affecting the entire computing system first and then affect the systems connected to it.
- Classification of Viruses based on their Lifetime:
  - Transient Virus:
    - The virus has a life that depends on the life of its host
    - The virus runs when its attached program executes and terminates when its attached program ends
    - During its execution, a transient virus may infect other programs
  - Resident Virus:
    - The virus remains active in memory or be activated as a stand-alone program, even after its attached program ends.
    - Resident viruses also target for system library (.dll) files to get attached as they are used several times by the OS and other application programs.
- Compression Virus:
  - A virus could be easily detected if the infected version of a program is longer than the corresponding uninfected one.
  - To thwart such attempts, viruses are also typically coded to compress the targeted executable file so that both the infected and uninfected versions are of identical length.

Activating a Virus Code

- An executable virus code sitting on a disk does nothing. The code needs to be executed to start replicating.

- Virus activation through running a Software installation SETUP program
  - While installing a software on a computer, we may call several other programs – some on the distribution medium (CD), some already existing in the computer and some in memory. If the virus is contained in any one of these programs, the virus is activated.
  - Once activated, the virus could install itself on a permanent storage medium and in any or all of the executing programs in memory.
  - No human intervention is needed after this. The virus can spread by itself.

- Viruses activated through e-mail:
  - The executable of the virus code could be sent as an email attachment. When this attachment is opened, the virus could be activated.
  - Sometimes, the virus could be hidden even in a .jpeg file. If the .jpeg file is opened using a photo viewer software, the virus code could get activated.

Virus Phases

- Dormant Phase:
  - The virus remains dormant (passive) hiding itself to avoid detection.
- Propagation Phase:
  - The virus is activated when the program on which it resides is run. The virus replicates itself to infect new files on new systems
- Triggering Phase:
  - When some logical condition becomes true, the virus moves from a dormant or propagation phase to perform its intended action
- Action Phase:
  - The virus performs the malicious action that it was designed to perform, called the payload. The action could range anywhere from something seemingly innocent (like displaying a silly picture on the computer screen) to something quite malicious (like deleting all critical files on the hard drive).
- A virus code contains the replication code (executed during the propagation phase) and the Payload code – to do the malicious action (executed during the action phase).

Types of Viruses

- Based on the way they spread or the types of files that they infect, viruses can be classified into:
- Program Virus, Macro Virus and Boot Sector Virus
- Program Virus (a.k.a. Executable File Virus):
  - Infects a program by modifying the file containing its object code.
  - Once the infection occurs, whenever the program file is executed, the virus is sure to run. Hence viruses often target to attach themselves to the most common and popular programs that are run several times (e.g., .dll files)
- Macro Virus (a.k.a. Document Virus):
  - Several document preparation programs (like MS Word) support powerful macro systems that facilitate dynamic updating of information (like date, time, figure #, page #, name) in the documents when there is an underlying change in the documents.
  - A Macro Virus targets attaching to these macro systems and replicates whenever a document is opened.
  - As part of replication, the virus searches for other documents to infect.
  - A Macro Virus can also insert itself into the standard document template, so that every newly created document gets infected.
  - Example: Melissa Virus - If a Word document containing the virus is downloaded and opened, then the macro in the document runs and attempts to mass mail itself.
  - To do mass-mail, the macro collects the first 50 entries from the alias list or address book and sends itself to the e-mail addresses in those entries.

Boot Sector and Bootstrap Loader

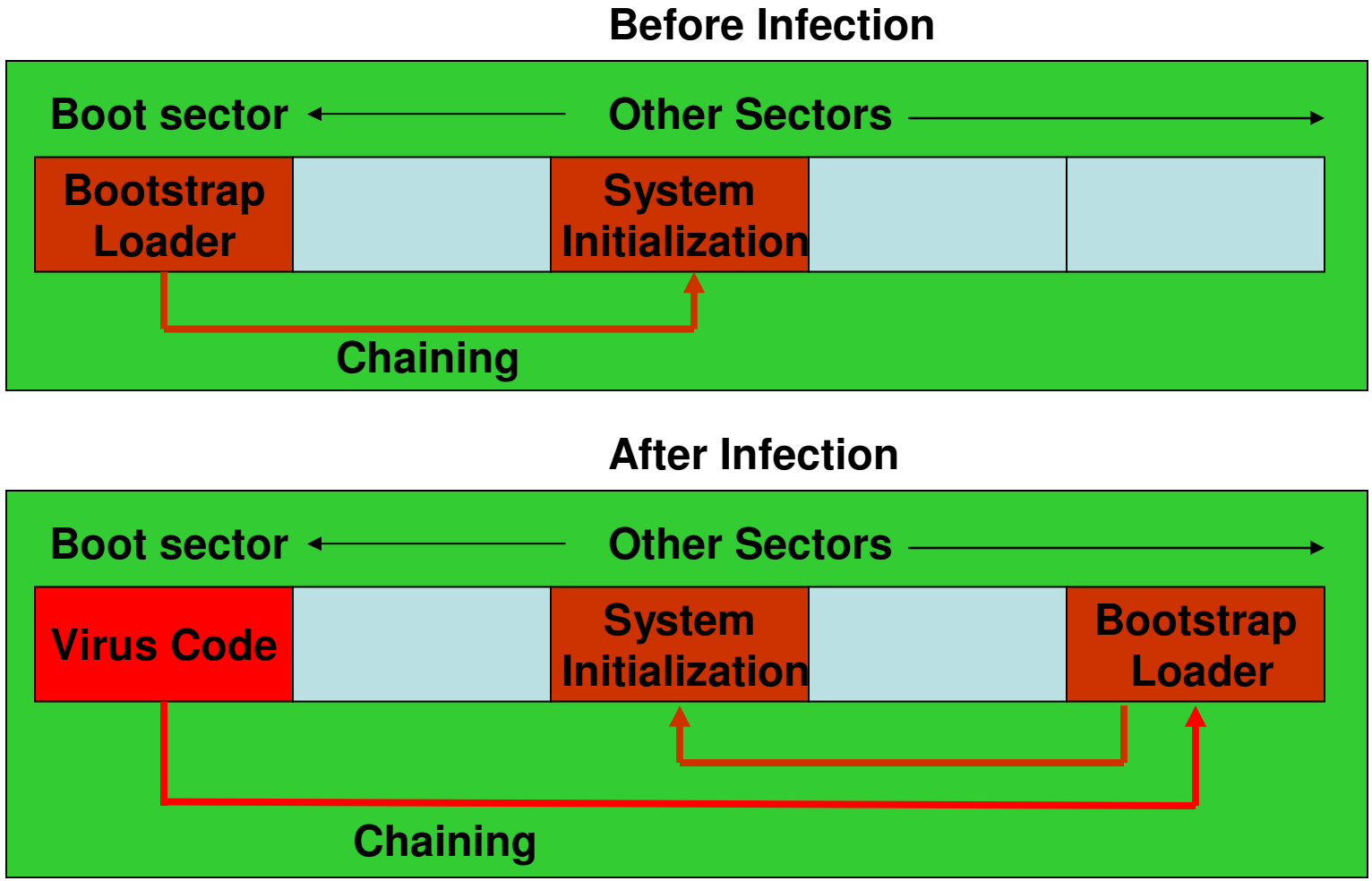
- BIOS:
  - When a computer is started, control begins with firmware (BIOS – Basic Input Output System) that determines which hardware components are present, tests them, and transfers control to an operating system.
  - The BIOS is stored in a special reserved memory called CMOS, part of your hardware, and the CMOS is powered by a battery.
- Bootstrap loader:
  - The BIOS transfers control to the OS by reading a fixed number of bytes (called the bootstrap loader) from a fixed location on the disk called the boot sector to a fixed address in memory and then jumping to that address that contains the first instruction of the bootstrap loader.
  - The bootstrap loader then reads into memory the rest of the OS from disk.
  - To run a different operating system, the user just inserts a disk with the new operating system and a bootstrap loader. When the user reboots from this new disk, the loader there brings and runs another operating system.
- Chaining for bootstrap loader:
  - The boot sector on a PC is slightly less than 512 bytes, but since the bootstrap loader of different OS could be larger, the hardware designers support chaining, in which each block of the bootstrap is chained to (contains the disk location of) the next block.

Boot Sector Viruses

- The chaining simplifies the installation of a virus
- The boot sector could be infected with the virus code, the bootstrap loader relocated and the bootstrap chain could be reconnected.
- Next time, the system is rebooted the virus code in the boot sector would be executed and the virus is activated.
- As the virus gains control of the system very early in the boot process, it can avoid, or at least complicate, detection.
- The files in the boot area are crucial parts of the OS. Consequently, to keep users from accidentally modifying or deleting them with disastrous results, the OS makes them “invisible” by not showing them as part of a normal listing of stored files, preventing their deletion. Thus, the virus code is not readily noticed by users.

- A boot sector virus is spread via infected floppy disks.
- This typically occurs when users inadvertently leave a floppy disk in drive A.
- When the system is next started, the PC will attempt to boot from the floppy. If the disk is infected with a boot sector virus, that virus will infect the boot sector of the user’s local drive C:

Boot Sector Viruses



Boot Sector Viruses

- How the infection could occur?
  - A boot sector virus is spread via infected floppy disks.
  - This typically occurs when users inadvertently leave a floppy disk in drive A.
  - When the system is next started, the PC will attempt to boot from the floppy. If the disk is infected with a boot sector virus, that virus will infect the boot sector of the user's local drive C:
  - If the floppy disk contained only the boot sector virus and is not a bootable disk (i.e., does not have the bootstrap loader), the user will simply see a standard warning that the drive contains a "non-system disk or disk error" and the user will be prompted to "replace the disk and press any key when ready".
  - The user will realize a floppy has been left in the drive, will then remove it and reboot the system, unaware that their system might have been just infected with a boot sector virus.
  - If your floppy disk is bootable, you would not receive the above error message and you will be simply booted to a DOS screen.

  - Note: If the floppy disk did not have a boot sector virus and is not a bootable disk, you will still see the above standard error message and there is nothing to panic. Just remove your floppy disk.

Boot Sector Viruses

- How to avoid the infection?
  - Don't let a floppy disk be in your floppy drive while booting up your system.
  - A boot sector infected hard drive will also infect any floppies used in the system. So write protect your disks when used in other person's systems (especially when used only for read-only purposes).
  - You could change the boot sequence of your PC by changing the boot settings in the BIOS CMOS screen.

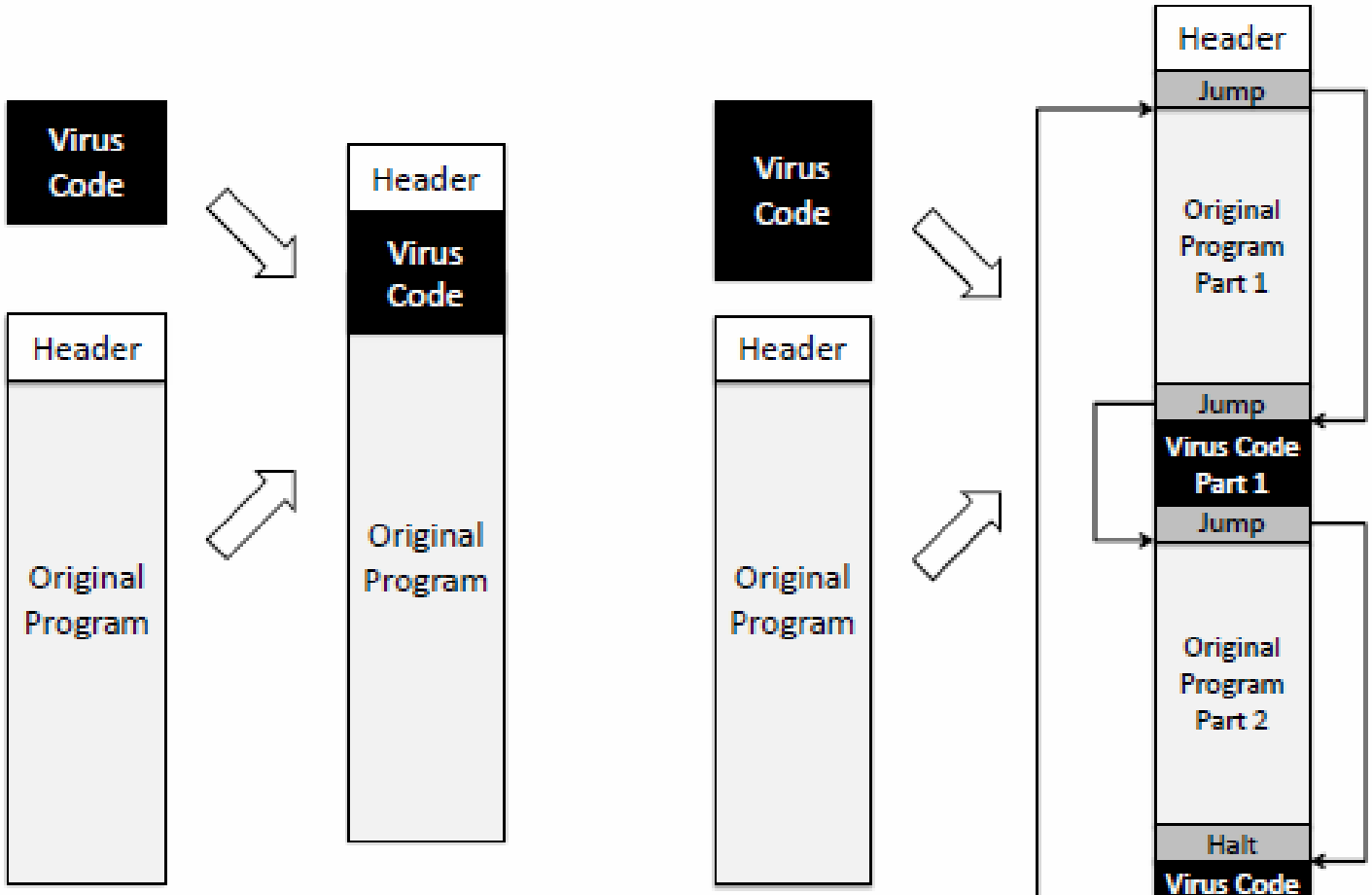
- Brain: The first Boot Sector Virus
  - Brain is a stealth, memory-resident, first boot sector infecting virus, believed to have originated from Pakistan.
  - Upon infection, the Brain virus becomes memory resident, taking up between 3K and 7K of RAM.
  - The Brain virus is able to hide from detection by using the Stealth technique.
    - When a request to interrogate the boot sector arrives, the virus intercepts the request and redirects the read request to the original boot sector located elsewhere on the disk.

Virus Signatures

- A virus cannot be completely invisible as it has to be somewhere in the memory to execute. Also, the viruses execute in particular ways, using certain methods to spread.
- A virus signature is a characteristic byte-pattern that is part of a certain virus or family of viruses.
- A virus signature is used as a digital fingerprint by the virus scanners to automatically detect and in some cases, remove viruses.
- A virus scanner searches memory and long-term storage, monitors execution (like during opening of emails, downloading an email attachment, intercepting system calls associated with file operations so that the file is scanned before being written to the disk) and watches for the virus signatures.
- If a virus scanner recognizes a known virus' pattern, it can block the virus, inform the user and deactivate or remove the virus. So, a virus scanner is effective only if it has been kept up-to-date with the latest information on current viruses.
- Some viruses employ techniques that make detection by means of signatures difficult. These viruses modify their code on each infection and each infected file contains a different variant of the virus.

Virus – Degree of Complication

- Viruses have various degrees of complication in how they can insert themselves in program/object code.



A Metamorphic virus sub-divides itself into multiple pieces and injects them into different locations in a program file

Source: Figure 4.4 from Introduction to Computer Security, M. Goodrich & R. Tamassia, Addison-Wesley (2011)

Recognizable Patterns in Viruses

- If the virus code attached to a program is invariant, then the virus code becomes a detectable signature.
- A virus scanner looking for Code Red Worm could look for a pattern containing the following characters:

```
/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
%u9090%u6858%ucbd3
%u7801%u9090%u6858%ucdb3%u7801%u9090%u6858
%ucbd3%u7801%u9090
%u9090%u8190%u00c3%u0003%ub00%u531b%u53ff
%u0078%u0000%u00=a
HTTP/1.0
```

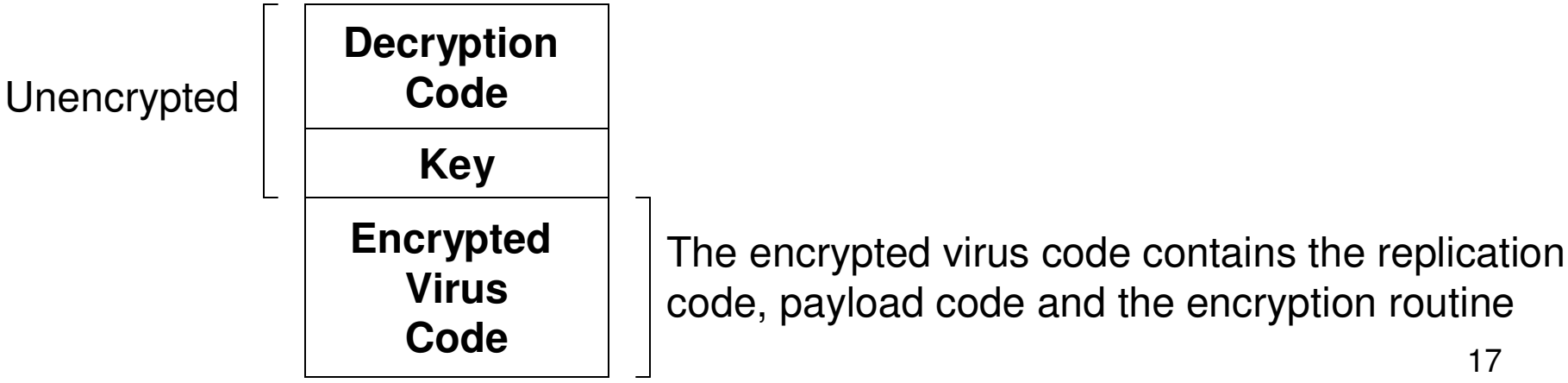
- A virus scanner can also look for suspicious patterns, such as a JUMP instruction as the first instruction of a system program (in case the virus has positioned itself at the bottom of the file but wants to be executed first).
- An anti-virus software can compute a code or checksum to detect changes to “bait” files that are specially created to be infected by a virus. Some viruses are programmed not to infect these “bait” files – small program files or programs that contain certain patterns of “garbage instructions”.

Quarantine

- Any file that contains a part that matches with the virus signature will be set aside to a protected storage, known as a quarantine, and the spread of infection is prevented.
- The suspected file in the Quarantine is later inspected in detail to determine the next course of action, which could be:
  - Restoring the file to its original location
  - Deleting file from the system
  - Replacing the infected file with its original (uninfected) version by potentially removing the virus code fragments
- Interacting with a file in quarantine is possible only through the antivirus program
- The file in quarantine is harmless because it is encrypted
- Usually the quarantine technique is proprietary and the details are kept secret

Encrypted Virus, Polymorphic Virus

- In order to complicate virus detection, virus writers encrypt the main part of the code (including the replication code, the payload code to do any malicious action as well as to encrypt the replicated virus code).
- To facilitate decryption of the encrypted virus code, a decryption routine and an encryption key are included along with the encrypted virus code.
- To complicate detection and create variation in the signature of the encrypted code, different randomly generated keys are used during each replication/encryption. Also, the plaintext decryption code is morphed (changed) for each variant. Such viruses are called polymorphic viruses.
- To detect encrypted viruses, including polymorphic viruses, virus scanners are updated with signatures of different decryption routines.



Metamorphic Viruses

- If a virus signature is invariant, then the virus scanners could be programmed to look for the signature and remove the virus.
- Understanding this, clever virus writers have developed viruses that have different forms (signatures), but are equivalent to each other in terms of functionality. Such viruses are called metamorphic viruses.
- Metamorphic viruses do not employ any encryption for obfuscation.
- Example:
  - Assume a virus writer has 100 bytes of code and 50 bytes of data.
  - To make two virus instances different, the writer might distribute the version as 100 bytes of code followed by all 50 bytes of data.
  - A second version could be 99 bytes of code, a jump instruction, 50 bytes of data, and the last byte of code.
  - Other versions are 98 code bytes, jumping to the last two; 97 and three; and so forth.
  - Just by moving pieces around, the virus writer can create enough different appearances (signatures) to fool simple virus scanners.
- A more sophisticated metamorphic virus randomly intersperses harmless instructions (adding zero to a number, jump to the next instruction, etc) throughout its code. These extra instructions make it more difficult to locate an invariant signature.

Developing and Detecting Metamorphic Viruses

- Metamorphic viruses are more difficult to develop than polymorphic viruses because in the former, the morphing engine (the code that reorganizes the virus code to produce different forms) itself has to be morphed for each variant. On the other hand, in polymorphic virus, the morphing engine is encrypted and remains the same in each variant of the virus.
- Signature schemes:
  - Sequence Signature: A set of strings that must appear in a particular sequence in a code.
  - Conjunction Signature: A set of strings must appear, in any order, in the virus code.
  - Probabilistic Signature: The appearance of each string (among a set of candidate strings) in the virus code is given a particular score. If the sum of the scores of all the strings exceeds a particular threshold, the virus scanner alerts for the presence of the virus.
- Virus scanners also employ as signatures, the presence of too many:
  - Useless instructions injected throughout the code
  - Reordering of independent instructions
  - Replacement of instructions, with alternate and equivalent instructions

Probability of Detection Anti-Virus Software

- **Question:** Suppose there is a new computer virus, VN1Q, which is both polymorphic and metamorphic. Ron has a new malware-detection program, DetVirus, that is 95% accurate at detecting VN1Q. That is, if a computer is infected with VN1Q, then DetVirus will correctly detect this fact 95% of the time, and if a computer is not infected, then DetVirus will correctly detect this fact 95% of the time. It turns out that the VN1Q virus will only infect any given computer with a probability of 1%. Nevertheless, you are nervous and run DetVirus on your computer, and it unfortunately says that your computer is infected with VN1Q. Find the probability that your computer really is infected?
- **Solution:** Consider a campus that has 4000 computers. According to the data given, the VN1Q virus will infect only 40 of these computers. The DetVirus software will correctly detect the presence of the virus in 38 of these computers and will also falsely detect the presence of the virus in (4000-40)*0.05 = 198 of the 3960 non-infected computers. Hence, the probability that the computer, on which DetVirus raises an alarm, is really infected = 38/(38+198) = 16.1%.

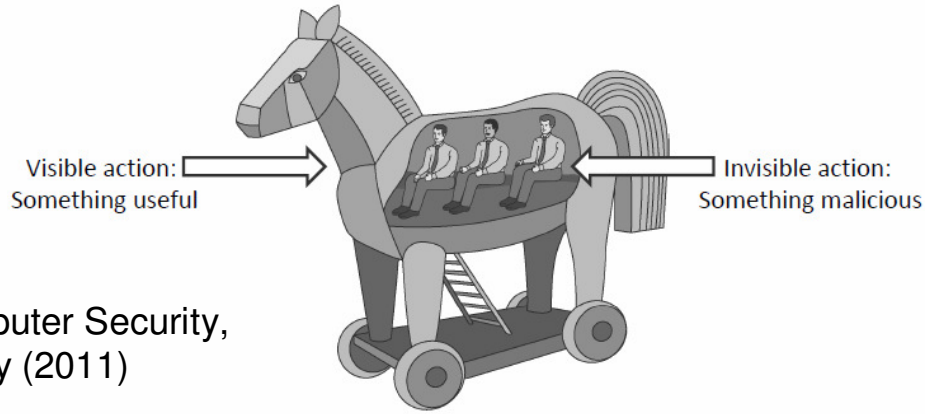
Math Problem on Metamorphic Virus

- **Question:** Suppose that a metamorphic virus, XYZVirus, is 98% useless bytes and 2% useful bytes. Unfortunately, XYZVirus has infected the login program on your Linux system and increased its size from 32K bytes to 1,032K bytes; hence, 1,000K bytes of the login program now consists of the XYZVirus. Bob has a cleanup program, XYZSweep, that is able to prune away the useless bytes of the XYZVirus, so that in any infected file it will consist of 96% useless bytes and 4% useful bytes. If you apply XYZSweep to the infected login program, what will be its new size?

- **Solution:** The useful bytes of the virus in the login program file, before cleanup is 1000*0.02 = 20K bytes. After XYZSweep cleans up the login program, the latter will have 4% useful bytes. Hence, the total size of the XYZVirus occupying the login program after the cleanup is 20K/0.04 = 500K bytes. Hence, the new size of the login program after the cleanup is 500K (total virus code) + 32K (original program code) = 532K.

Trojan Horse (Trojan)

- A Trojan Horse is a malware program that appears to perform some useful task, but which also does something with negative consequences.
- Trojans are either installed as part of the payload of other malware or (deliberately or accidentally) by a user or administrator.
- Unlike viruses, Trojans do not replicate themselves; but, they can be just as destructive and often facilitate an attacker to have unauthorized remote access to computer.
- Example: A login script that solicits a user's identification and password, passes the identification information to the rest of the system for login processing, but also retains a copy of the info for later malicious use.



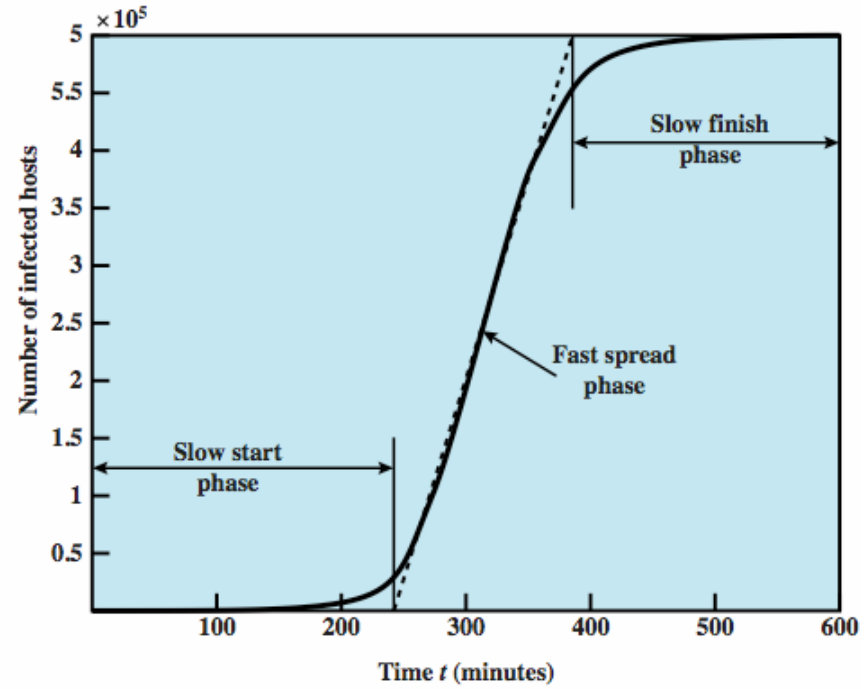
Source: Figure 4.6 from Introduction to Computer Security, M. Goodrich & R. Tamassia, Addison-Wesley (2011)

Computer Worms

- A Computer Worm is a malware that can replicate itself without requiring to be attached to other programs as well as without human intervention.
- In most cases, a worm carries a malicious payload that may include deleting all the files in the infected system and/or installing a Trojan.
- Worm Propagation
  - Worms typically spread by exploiting the vulnerabilities in applications run by Internet-connected systems with security hole.
  - To propagate, a worm usually searches for other systems to infect by examining host tables or similar repositories of remote system addresses; establishes a connection with a remote system; and copies itself to the remote system and cause the copy to be run.
  - Any target machine that is vulnerable to attack is likely to get infected and will also try to infect some other machines in turn.
  - A worm attempts to persist in the infected machine and survive rebooting. In this pursuit (in Windows machines), worms modify the Windows Registry – a database used by the OS to run certain programs and services or device-drivers on startup. Worms typically associate the path to their execution file to: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

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 on a recent previous attack may be countered more effectively than a more novel attack.
- Clearly, the objective in countering a worm is to catch the worm in its slow start phase, at a time when few hosts have been infected.

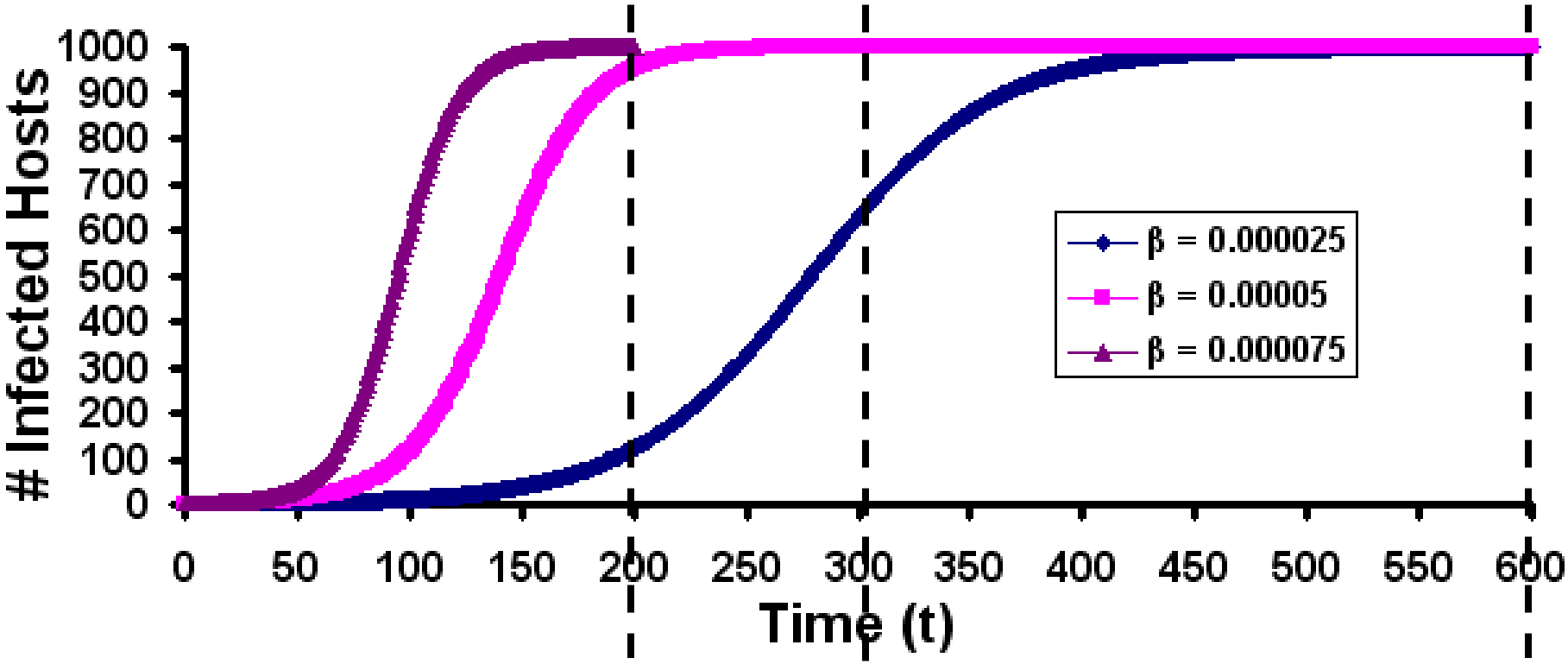


Source: Figure 21.6 from William Stallings – Cryptography and Network Security, 5th Edition

Worm Propagation Model (Math)

- Basis: Epidemic Theory
- Notations:
  - N – the total number of vulnerable hosts
  - I(t) – number of infected hosts at time t
  - S(t) – number of susceptible hosts at time t: We say a host is susceptible if it is vulnerable but not infected yet
  - β – infection rate, a constant associated with the speed of propagation of the worm.
  - Starting from a single infected host, the change of I(t) and S(t) over time can be expressed by the following formulae:
  - $I(0) = 1$
  - $S(0) = N-1$
  - $I(t+1) = I(t) + \beta * I(t) * S(t)$
  - $S(t+1) = N - I(t+1)$
  - **Note:** As β is increased by a factor of α, the saturation time (time at which all the vulnerable hosts are highly likely to be infected) decreases by 1/α

Worm Propagation Model (Math)



Saturation Time for different values of β (the infection rate)

Real-World Examples of Computer Worms

Morris (1998)
I Love You (2000)
Code Red Worm (2001)
Blaster (2003)
Mydoom (2004)
Sasser (2004)
Conficker (2008)

Worm propagation can best be captured in a graph theory perspective through the well-known DFS algorithm

Worm Countermeasures

- There is considerable overlap in techniques for dealing with viruses and worms.
- Once a worm is resident on a machine, antivirus software can be used to detect it.
- In addition, because worm propagation generates considerable network activity, the monitoring of that activity can form the basis of a worm defense.
- Defense strategies are primarily focused at blocking the outgoing connection attempts from the suspected hosts or sometimes the entire network.
  - The blocks are implemented at the network firewall as packet filters
  - Intrusion detection systems can also be deployed to develop heuristic models capturing the worm signatures, which can be then incorporated as part of the filtering rules of proxy firewalls to examine outgoing and incoming packets.

Rootkits

- A rootkit is a stealthy type of malware that alters the system utilities or the OS itself to prevent detection.
  - Often employed by malware like viruses and Trojan to hide their detection
- For example, a rootkit can infect the Windows Process Monitor utility (that lists the currently running processes) and hide by removing itself from the list. A rootkit can detect the Windows Explorer and hide itself from being listed when the user browses files.
- Rootkits can run in two modes:
  - ***User-mode rootkits*** – work by altering system utilities or libraries on disk.
    - Easy to detect using offline cryptographic hash computation and using digitally signed code libraries.
  - ***Kernel-mode rootkits*** – work at the lowest level of the OS by getting loaded as a device-driver and survive reboots by altering the System Registry/ daemons
  - Kernel-mode rootkits hide their detection by ***function hooking*** – a technique by which OS functions loaded into the kernel memory are modified or replaced with customized functions that hide the existence of the rootkit and its associated malware programs.
    - Kernel files that list the registries can also be modified by a rootkit running with kernel-privileges.

Rootkits

- Software to detect kernel-mode rootkits store the signature of regular kernel functions commonly targeted by rootkits, and inspect kernel memory functions to determine if any modifications have been made to these functions.
- However, since kernel rootkits operate at the highest level of system privileges, they might preemptively detect the anti-rootkit software and disable them.
- Hence, an in-depth offline analysis of an infected system, including inspection of the registry and the boot records, is required to detect rootkits.
- RootkitRevealer (Kernel-mode Anti-Rootkit Software)
  - Two scans of file system
  - High-level scan using the Windows API to list and read the files corresponding to the system utilities, kernel functions, and registries
  - Raw scan using low-level disk access methods on the files and read one block at a time
  - Discrepancy reveals presence of rootkit

Zero-Day Attacks

- A “Zero-day attack” is an attack that exploits a vulnerability that was previously unknown, even to the software designers who created the system containing this vulnerability. Hence, cannot be detected using signature-based schemes.
- Malware programmers often test their code against several state-of-the-art anti-malware programs and release their malware only if these programs fail to detect them. This is often done to trigger zero-day attacks.

- Zero-day attacks are best detected using heuristics incorporated into an anomaly-based Intrusion detection system that regularly monitors programs (that are also sometimes in an isolated virtual machine environment) for any abnormal activities such as:
  - Deleting files
  - Sending information over the Internet
- Care should be taken while deciding at the heuristics and the threshold number of abnormal activities before raising an alarm, so that the anomaly-based IDS does not generate too much of false positives.

Botnets

- A Botnet is a network of compromised machines (called zombies) that are under the control of a centrally owned command-and-control (called bot herder).
- Earlier, botnets hosted the command-and-control stations at static IP addresses that were coded into the bot software of each infected machine.
- To prevent detection and shutdown, botnets now change the address of the bot herder, on a daily basis, dynamically generating and updating the zombies, through unexpected channels to receive commands.
- Botnets are typically used to launch distributed denial of service attacks that lead to harnessing credit card information from several computers as well as generate spam emails
- Roughly, 25% of the computers in the Internet have been estimated to be part of some kind of botnet.

Terminology of Malicious Programs

- Virus - Malware that, when executed, tries to replicate itself into other executable 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.
- Logic Bomb - A program inserted into software by an intruder. A logic bomb lies dormant until a predefined condition is met; the program then triggers an unauthorized act.
- 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.
- Backdoor (Trapdoor) - Any mechanism that bypasses a normal security check; it may allow unauthorized access to functionality.

Source: Table 21.1 from William Stallings – Cryptography and Network Security, 5th Edition

Terminology of Malicious Programs

- Mobile Code - Software (e.g., script, macro, or other portable instruction) that can be shipped unchanged to a heterogeneous collection of platforms and executed with identical semantics.
- Rootkit - Set of hacker tools used after attacker has broken into a computer system and gained root-level access.
- Zombie (bot) - Program installed on an infected machine that is activated to launch attacks on other machines.
- Spyware - Software that collects information from a computer and transmits it to another system.
- Adware - Advertising that is integrated into software. It can result in pop-up ads or redirection of a browser to a commercial site.

Source: Table 21.1 from William Stallings – Cryptography and Network Security, 5th Edition