Malware: Malicious Software
Viruses, Worms, Trojans, Rootkits

- **Malware** can be classified into several categories, depending on propagation and concealment
  - **Propagation**
    - **Virus**: human-assisted propagation (e.g., open email attachment)
    - **Worm**: automatic propagation without human assistance
  - **Concealment**
    - **Rootkit**: modifies operating system to hide its existence
    - **Trojan**: provides desirable functionality but hides malicious operation
- Various types of payloads, ranging from annoyance to crime
Insider Attacks

- An **insider attack** is a security breach that is caused or facilitated by someone who is a part of the very organization that controls or builds the asset that should be protected.
- In the case of malware, an insider attack refers to a security hole that is created in a software system by one of its programmers.
Backdoors

- A backdoor, which is also sometimes called a trapdoor, is a hidden feature or command in a program that allows a user to perform actions he or she would not normally be allowed to do.

- When used in a normal way, this program performs completely as expected and advertised.

- But if the hidden feature is activated, the program does something unexpected, often in violation of security policies, such as performing a privilege escalation.

- Benign example: Easter Eggs in DVDs and software
Logic Bombs

• A **logic bomb** is a program that performs a malicious action as a result of a certain logic condition.

• The classic example of a logic bomb is a programmer coding up the software for the payroll system who puts in code that makes the program crash should it ever process two consecutive payrolls without paying him.

• Another classic example combines a logic bomb with a backdoor, where a programmer puts in a logic bomb that will crash the program on a certain date.
The Omega Engineering Logic Bomb

• An example of a logic bomb that was actually triggered and caused damage is one that programmer Tim Lloyd was convicted of using on his former employer

• Omega Engineering Corporation. On July 31, 1996, a logic bomb was triggered on the server for Omega Engineering’s manufacturing operations
The Omega Bomb Code

• The Logic Behind the Omega Engineering Time Bomb included the following strings:
  • 7/30/96
    – Event that triggered the bomb
  • F:
    – Focused attention to volume F, which had critical files
  • F:\LOGIN\LOGIN 12345
    – Login a fictitious user, 12345 (the back door)
  • CD \PUBLIC
    – Moves to the public folder of programs
  • FIX.EXE /Y F:\*.*
    – Run a program, called FIX, which actually deletes everything
  • PURGE F:\ALL
    – Prevent recovery of the deleted files
Defenses against Insider Attacks

• Avoid single points of failure.
• Use code walk-throughs.
• Use archiving and reporting tools.
• Limit authority and permissions.
• Physically secure critical systems.
• Monitor employee behavior.
• Control software installations.
Computer Viruses

• A computer virus is computer code that can replicate itself by modifying other files or programs to insert code that is capable of further replication.

• This self-replication property is what distinguishes computer viruses from other kinds of malware, such as logic bombs.

• Another distinguishing property of a virus is that replication requires some type of user assistance, such as clicking on an email attachment or sharing a USB drive.
Biological Analogy

- Computer viruses share some properties with Biological viruses

- Attack
- Penetration
- Replication and assembly
- Release
Virus Phases

• **Dormant phase.** During this phase, the virus just exists—the virus is laying low and avoiding detection.

• **Propagation phase.** During this phase, the virus is replicating itself, infecting new files on new systems.

• **Triggering phase.** In this phase, some logical condition causes the virus to move from a dormant or propagation phase to perform its intended action.

• **Action phase.** In this phase, the virus performs the malicious action that it was designed to perform, called payload.
  – This action could include something seemingly innocent, like displaying a silly picture on a computer’s screen, or something quite malicious, such as deleting all essential files on the hard drive.
Infection Types

• Overwriting
  – Destroys original code
• Pre-pending
  – Keeps original code, possibly compressed
  – E.g., kernel32.dll
• Infection of libraries
  – Allows virus to be memory resident
  – E.g., kernel32.dll
• Macro viruses
  – Infects MS Office documents
  – Often installs in main document template
Degrees of Complication

- Viruses have various degrees of complication in how they can insert themselves in computer code.
Concealment

- **Encrypted virus**
  - Decryption engine + encrypted body
  - Randomly generate encryption key
  - Detection looks for decryption engine

- **Polymorphic virus**
  - Encrypted virus with random variations of the decryption engine (e.g., padding code)
  - Detection using CPU emulator

- **Metamorphic virus**
  - Different virus bodies
  - Approaches include code permutation and instruction replacement
  - Challenging to detect
Computer Worms

• A computer worm is a malware program that spreads copies of itself without the need to inject itself in other programs, and usually without human interaction.

• Thus, computer worms are technically not computer viruses (since they don’t infect other programs), but some people nevertheless confuse the terms, since both spread by self-replication.

• In most cases, a computer worm will carry a malicious payload, such as deleting files or installing a backdoor.
Early History

● First worms built in the labs of John Shock and Jon Hepps at Xerox PARC in the early 80s

● CHRISTMA EXEC written in REXX, released in December 1987, and targeting IBM VM/CMS systems was the first worm to use e-mail service

● The first internet worm was the **Morris Worm**, written by Cornell student Robert Tappan Morris and released on November 2, 1988
Worm Development

• Identify vulnerability still unpatched
• Write code for
  – Exploit of vulnerability
  – Generation of target list
    • Random hosts on the internet
    • Hosts on LAN
    • Divide-and-conquer
  – Installation and execution of payload
  – Querying/reporting if a host is infected
• Initial deployment on botnet

• Worm template
  – Generate target list
  – For each host on target list
    • Check if infected
    • Check if vulnerable
    • Infect
    • Recur

• Distributed graph search algorithm
  – Forward edges: infection
  – Back edges: already infected or not vulnerable
Worm Propagation

- Worms propagate by finding and infecting vulnerable hosts.
  - They need a way to tell if a host is vulnerable
  - They need a way to tell if a host is already infected.
Propagations: Theory

- Classic epidemic model
  - \( N \): total number of vulnerable hosts
  - \( I(t) \): number of infected hosts at time \( t \)
  - \( S(t) \): number of susceptible hosts at time \( t \)
  - \( I(t) + S(t) = N \)
  - \( \beta \): infection rate

- Differential equation for \( I(t) \):
  \[
dI/dt = \beta I(t) S(t)
  \]

- More accurate models adjust propagation rate over time

Source:
Propagation: Practice

- Cumulative total of unique IP addresses infected by the first outbreak of Code-Red I v2 on July 19-20, 2001

Source: