Malware: Malicious Software
An *insider attack* is a security hole in a software system created by one of its programmers.
An **insider attack** is a security hole in a software system created by one of its programmers.

**Backdoors or trapdoors**
- can be special commands or passwords used for debugging or bypassing security because of unusual circumstances
- can be a vulnerability in the program (e.g., buffer overflow)
An *insider attack* is a security hole in a software system created by one of its programmers.

**Backdoors or trapdoors**

- can be special commands or passwords used for debugging or bypassing security because of unusual circumstances
- can be a vulnerability in the program (e.g., buffer overflow)

→ Never removed after software is deployed or never made known to others by programmer.
→ An insider takes advantage of the backdoor.
Logic bombs

- a program that performs a malicious action as a result of a logic condition.

- Classic example: files are deleted when the programmer is not paid in two consecutive payrolls.
Logic bombs

- a program that performs a malicious action as a result of a logic condition.

- Classic example: files are deleted when the programmer is not paid in two consecutive payrolls.

- associated with disgruntled IT employees:

  In March 2002, about 2000 servers of UBS went down. The cause was traced to a logic bomb planted by Roger Duronio, a former systems administrator of UBS.

Cost to UBS: $3.1 million dollars.

Crime pays: Duronio was convicted to 97 months in jail.
Logic bombs

- a program that performs a malicious action as a result of a logic condition.

- Classic example: files are deleted when the programmer is not paid in two consecutive payrolls.

- associated with disgruntled IT employees:

  In March 2002, about 2000 servers of UBS went down. The cause was traced to a logic bomb planted by Roger Duronio, a former systems administrator of UBS.

  **His reason:** his annual bonus was smaller than expected. He also bought stock options that will give him a large payout if UBS’s stock took a dive within 11 days!

  **Cost to UBS:** $3.1 million dollars.

  **Crime pays:** Duronio was convicted to 97 months in jail.
Defenses against insider attacks

- Avoid single points of failure.
- Limit authority and permissions.
- Monitor employee behavior.
- Use code walk-throughs.
  (How Fannie Mae learned that one of their recently terminated employees had inserted a logic bomb in 2008.)
- Control software installations.
- Use archiving and reporting tools.
- Physically secure critical systems.
Computer Viruses

How biological viruses work:

- Virus lays dormant until it finds the right kind of cell.

Figure 4.3: Four stages of a biological virus.
How biological viruses work:

- Virus lays dormant until it finds the *right* kind of cell.
- It uses the cell’s own reproductive processes to replicate itself.

**Figure 4.3**: Four stages of a biological virus.
Computer Viruses

How biological viruses work:

- Virus lays dormant until it finds the right kind of cell.
- It uses the cell’s own reproductive processes to replicate itself.
- Eventually, the viruses escape in great numbers.

**Figure 4.3:** Four stages of a biological virus.
A *computer virus* is code that inserts itself into another file or program which then *replicates* the virus when opened or run. It requires *user assistance* for replication to take place. Often, it will also *perform some malicious task*. 

Different phases:

- **Dormant phase.** The virus exists but is laying low to avoid detection.
- **Propagation phase.** The virus is replicating itself, infecting new files and systems.
- **Triggering phase.** Some logical condition causes the virus to perform its intended action.
- **Action phase.** The virus performs its intended action, a.k.a. its *payload*. 

*Malware*
A computer virus is code that inserts itself into another file or program which then replicates the virus when opened or run. It requires user assistance for replication to take place. Often, it will also perform some malicious task.

Different phases:

- **Dormant phase.** The virus exists but is laying low to avoid detection.
- **Propagation phase.** The virus is replicating itself, infecting new files and systems.
- **Triggering phase.** Some logical condition causes the virus to perform its intended action.
- **Action phase.** The virus performs its intended action, a.k.a. its payload.
Types of viruses:

1. Program or file virus
   - infects a program by modifying the file containing its object code.
Example: the *Jerusalem* virus

- discovered in the 80’s in Jerusalem, Israel.
- infected DOS operating systems files.
- lots of variants but is no longer effective because of the advent of Windows.
Example: the *Jerusalem* virus

- discovered in the 80’s in Jerusalem, Israel.
- infected DOS operating systems files.
- lots of variants but is no longer effective because of the advent of Windows.
- **Propagation method:** when it becomes active, it loads itself into main memory and infects other executable files that are run.
- **Trigger & Payload:** if it is executed on a Friday the 13th, it deletes every file that is run.
2. Macro virus

- *Macros* in document preparation programs (e.g. MS Word, MS Excel, etc.) are “mini-programs” that allow for automating command sequences.

- A *macro virus* infects a document by inserting itself into the document’s macro or the standard document template. In the latter case, every newly created document gets infected.
Example: the Melissa virus of 1999

- a macro virus that infected MS Word and Excel documents
- first virus to spread itself via mass emailing.
- **Trigger & Payload:** when an infected file is opened, the virus would email the first 40 or 50 addresses in the victim’s address book with an infected file as attachment. It effectively launched a DoS attack.
Example: the Melissa virus of 1999

- a macro virus that infected MS Word and Excel documents
- first virus to spread itself via mass emailing.
- **Trigger & Payload:** when an infected file is opened, the virus would email the first 40 or 50 addresses in the victim’s address book with an infected file as attachment. It effectively launched a DoS attack.

**Question:** A macro virus via LaTex?
Detecting the presence of a virus: look for virus signatures

- Once a computer virus is unleashed into the “wild”, it infects other files.
- Systems administrators and antivirus software companies are alerted. Infected files are brought in.
- Experts look for code fragments that are unique to the virus – a.k.a. the virus’ *signature*.
Detecting the presence of a virus: look for virus signatures

- Once a computer virus is unleashed into the “wild”, it infects other files.
- Systems administrators and antivirus software companies are alerted. Infected files are brought in.
- Experts look for code fragments that are unique to the virus – a.k.a. the virus’ signature.
- Anti-viral softwares are updated to contain the signature. Files are scanned to determine if they contain the virus’ signature.
- If infected files are found, they are quarantined – deleted, modified or replaced.

This is why it is important to keep anti-viral software updated!
The cat-and-mouse game

Virus creators know what anti-viral software are looking for – so they modify!
Current tricks:
1. Encrypted viruses
   - the main virus code is encrypted so that the virus’ signature is hidden.

   Figure 4.5: How an encrypted virus is structured.

   - to detect it: look for decryption code.
2. Polymorphic and metamorphic viruses

- it mutates as it replicates so there is no “static” signature for the virus
2. Polymorphic and metamorphic viruses

- it mutates as it replicates so there is no “static” signature for the virus
- **Polymorphic viruses** take encrypted viruses one step further: each copy is encrypted using a different key.

Metamorphic viruses uses non-cryptographic means to hide itself. It may reorder instructions, add junk instructions etc. To detect it, identify useless code and use inexact signature matching.
2. Polymorphic and metamorphic viruses

- it mutates as it replicates so there is no “static” signature for the virus
- **Polymorphic viruses** take encrypted viruses one step further: each copy is encrypted using a different key.
- **Metamorphic viruses** uses non-cryptographic means to hide itself. It may reorder instructions, add junk instructions etc. To detect it, identify useless code and use inexact signature matching.
Worms

A *worm* is a malicious code that also replicates itself and spreads to other computers usually via computer networks. Unlike viruses, worms *do not* inject themselves into computer programs and usually survive without human interaction.
A *worm* is a malicious code that also replicates itself and spreads to other computers usually via computer networks. Unlike viruses, worms *do not* inject themselves into computer programs and usually survive without human interaction.

**Main mechanism:** exploits security holes in the target computer (e.g., buffer overflow)

**Propagation:** A worm propagates by having each infected computer attempt to infect other target machines via the internet.

**Sample payloads:** launch DoS attacks on selected websites, do cryptoviral extortion attacks by encrypting files, install backdoors to allow the creation of a “zombie” computer (e.g., botnets)

**Network harm:** Worms almost always causes network issues because their propagation method consumes network bandwidth.
The First Worm: the Morris Worm, a “good” worm
written by Robert Morris as a graduate student at Cornell
launched from MIT in November 1998 to “gauge the size of the internet”
exploited known vulnerabilities in Unix sendmail, finger, and rsh/rexec, as well as weak passwords
effectively caused an DoS attack
affected about 1/10 of the Internet at that time. Cost about $10M to clean up.
Robert Morris became the first person convicted under the 1986 Computer Fraud and Abuse Act.
now, a professor at MIT.
Why it caused so much damage:

- Before copying itself into another computer, it would first check if the computer was already infected.
- If no, it would go ahead and infect it. If yes, it would not 6/7 of the time.
- The 1/7 probability of re-infection was high enough that it caused infected computers to slow down significantly.
Why it caused so much damage:

- Before copying itself into another computer, it would first check if the computer was already infected.
- If no, it would go ahead and infect it. If yes, it would not infect the computer 6/7 of the time.
- The 1/7 probability of re-infection was high enough that it caused infected computers to slow down significantly.

To detect worms: signature-based file scans, network-level scans and filters
A Trojan horse is code that performs or appears to perform some useful task, but which also does something malicious on the side.
A Trojan horse is code that performs or appears to perform some useful task, but which also does something malicious on the side.

Main mechanism: entice users to download the program.

A double whammy: sometimes, users actually pay for these programs! They end up losing money and potentially giving away their credit card information.
Sample disguises & payload:

- **AIDS Trojan (late 1980’s):** provide important information / cryptographic extortion attack
- **Mocmex Trojan (2008):** digital photo frame/ when connected into a Windows machine, code is transferred that collects and transmits passwords
- **False antivirus software such as MacDefender (2011):** claims computer is infected and sells software/ hijacks user’s browser to display adult websites; also exposes users to identity theft. Sometimes these rogueware even disable the functionalities of real antivirus software.
26M new malware samples were identified in 2011. This is equivalent to 73,000 strains per day.

A classification:

A big chunk of the trojan horses were fake anti-viral software.
A rootkit is a kind of stealthy code whose main purpose is to prevent itself or other malicious codes from being detected. It does this by altering system utilities or the operating system itself.
A **rootkit** is a kind of *stealthy* code whose main purpose is to prevent itself or other malicious codes from being detected. It does this by altering system utilities or the operating system itself.

- It might modify a process monitor so that the rootkit is never listed in the current list of processes running. Hence, a user will never know it exists by simply checking the process monitor.

- It is often bundled with other malware such as Trojan horses or worms.
Rootkits can run at two levels:

- **User-mode**
  - alters system utilities or libraries on disk
  - relatively easy to detect: make sure important code libraries and critical system components are digitally signed or hashed. If signature or hash values are modified – they indicate the presence of a rootkit.

- **Kernel-mode**
  - works at the lowest level of the operating system
  - kernel rootkits are typically loaded as device drivers
  - a common method used is function hooking: modifies some kernel functions
  - much harder to detect
Rootkits can run at two levels:

- **User-mode**
  - alters system utilities or libraries on disk
  - relatively easy to detect: make sure important code libraries and critical system components are digitally signed or hashed. If signature or hash values are modified – they indicate the presence of a rootkit.

- **Kernel-mode**
  - works at the lowest level of the operating system
  - kernel rootkits are typically loaded as device drivers
  - a common method used is *function hooking*: modifies some kernel functions
  - much harder to detect
Rootkit Revealer by Bryce Cogswell and Mark Russinovich
(Sysinternals)

1. perform two scans of the file system:
2. a “high-level” scan using say a Windows API
3. a “raw” scan using disk access methods
4. if there is a discrepancy in the two scans, then a rootkit is present
5. could be defeated by rootkit that intercepts and modifies results of raw scan operations
The Sony BMG copyright protection rootkit scandal

1. In 2005, Sony BMG distributed music CDs with a software called Extended Copy Protection.

2. Unknown to the buyers of the CDs, the software included a rootkit that would install itself on PCs running Windows whenever the CD is placed in the optical disk drive.

3. The primary intent of the rootkit was to enforce copy protection.

4. To avoid detection, the rootkit hid from the user any file starting with "sys".
   
   **Problem:** writers of viruses and worms can use this feature to hide their files! Thus, the rootkit created a security hole!