# Last time

- Malicious code: Malware
  - Viruses
  - Trojan horses
  - Logic bombs
  - Worms
- Other malicious code: web bugs

# This time

- Other malicious code
  - Back doors
  - Salami attacks
  - Rootkits
  - Interface illusions
  - Keystroke logging
  - Man-in-the-middle attacks
- Nonmalicious flaws
  - Covert channels
  - Side channels

# Back doors

- A back door (also called a trapdoor) is a set of instructions designed to bypass the normal authentication mechanism and allow access to the system to anyone who knows the back door exists
  - Sometimes these are useful for debugging the system, but *don't forget to take them out* before you ship!

- Fanciful examples:
  - “Reflections on Trusting Trust” (mandatory reading)
  - “The Net”
  - “WarGames”

# Examples of back doors

- Real examples:
  - Debugging back door left in sendmail
  - Back door planted by Code Red worm
  - Port knocking
    - The system listens for connection attempts to a certain pattern of (closed) ports. All those connection attempts will fail, but if the right pattern is there, the system will open, for example, a port with a root shell attached to it.
  - Attempted hack to Linux kernel source code
    - ```
      if ((options == (__WCLONE|__WALL)) &&
                  (current->uid = 0))
              retval = -EINVAL;
      ```
      ```

# Sources of back doors

- Forget to remove them
- Intentionally leave them in for testing purposes
- Intentionally leave them in for maintainance purposes
  - Field service technicians
- Intentionally leave them in for malicious purposes
  - Note that malicious users can use back doors left in for non-malicious purposes, too!

# Salami attacks

- A salami attack is an attack that is made up of many smaller, often considered inconsequential, attacks
- Classic example: send the fractions of cents of roundoff error from many accounts to a single account owned by the attacker
- More commonly:
  - Credit card theives make very small charges to very many cards
  - Clerks slightly overcharge customers for merchandise
  - Gas pumps misreport the amount of gas dispensed

# Privilege escalation

- Most sytems have the concept of differing levels of privilege for different users
  - Web sites: everyone can read, only a few can edit
  - Unix: you can write to files in your home directory, but not in /usr/bin
  - Mailing list software: only the list owner can perform certain tasks

- A privilege escalation is an attack which raises the privilege level of the attacker (beyond that to which he would ordinarily be entitled)

# Sources of privilege escalation

- A privilege escalation flaw often occurs when a part of the system that legitimately runs with higher privilege can be tricked into executing commands (with that higher privilege) on behalf of the attacker
  - Buffer overflows in setuid programs or network daemons
  - Component substitution (see text)

- Also: the attacker might trick the system into thinking he is in fact a legitimate higher-privileged user
  - Problems with authentication systems
    - "`-froot`" attack

# Rootkits

- A rootkit is a tool often used by “script kiddies”
- It has two main parts:
  - A method for gaining unauthorized root / administator privileges on a machine (either starting with a local unprivileged account, or possibly remotely)
    - This method usually expoits some known flaw in the system that the owner has failed to correct
    - It often leaves behind a back door so that the attacker can get back in later, even if the flaw is corrected
  - A way to hide its own existence
    - “Stealth” capabilities
    - Sometimes just this stealth part is called the rootkit

# Stealth capabilities

- How do rootkits hide their existence?
  - Clean up any log messages that might have been created by the exploit
  - Modify commands like `ls` and `ps` so that they don't report files and processes belonging to the rootkit
  - Alternately, modify the kernel so that no user program will ever learn about those files and processes!

# Example: Sony XCP

- Mark Russinovich was developing a rootkit scanner for Windows
- When he was testing it, he discovered his machine already had a rootkit on it!
- The source of the rootkit turned out to be Sony audio CDs equipped with XCP “copy protection”
- When you insert such an audio CD into your computer, it contains an `autorun.exe` file which automatically executes
- `autorun.exe` installs the rootkit

# Example: Sony XCP

- The “primary” purpose of the rootkit was to modify the CD driver in Windows so that any process that tried to read from an XCP-protected CD would get garbled output
- The “secondary” purpose was to make itself hard to find and uninstall
  - Hid all files and processes whose names started with `$sys$`
- After people complained, Sony eventually released an uninstaller
  - But running the uninstaller left a back door on your system!

# Keystroke logging

- Almost all of the information flow from you (the user) to your computer (or beyond, to the Internet) is via the keyboard
  - A little bit from the mouse, a bit from devices like USB keys
- An attacker might install a keyboard logger on your computer to keep a record of:
  - All email / IM you send
  - All passwords you type
- This data can then be accessed locally, or it might be sent to a remote machine over the Internet

# Who installs keyboard loggers?

- There are certainly keyboard loggers installed by malware
  - Capture passwords, especially banking passwords
  - Send the information to the remote attacker
- But most keyboard loggers are installed by one family member to spy on another
  - Spying on children
  - Spying on spouses
  - Spying on boy/girlfriends

# Kinds of keyboard loggers

- Application-specific loggers:
  - Record only those keystrokes associated with a particular application, such as an IM client
- System keyboard loggers:
  - Record all keystrokes that are pressed (maybe only for one particular target user)
- Hardware keyboard loggers:
  - A small piece of hardware that sits between the keyboard and the computer
    - Works with any OS
    - Completely undetectable in software

# Interface illusions

- You use user interfaces to control your computer all the time
- For example, you drag on a scroll bar to see offscreen portions of a document
- But what if that scrollbar isn't really a scrollbar?
- What if dragging on that “scrollbar” really dragged a program (from a malicious website) into your “Startup” folder (in addition to scrolling the document)?
  - This really happened

# Interface illusions

- We expect our computer to behave in certain ways when we interact with “standard” user interface elements.
- But often, malicious code can make “nonstandard” user interface elements in order to trick us!
- We think we're doing one thing, but we're really doing another
- How might you defend against this?

# Phishing

- Phishing is an example of an interface illusion
- It looks like you're visiting Paypal's website, but you're really not.
  - If you type in your password, you've just given it to an attacker
- Advanced phishers can make websites that look every bit like the real thing
  - Even if you carefully check the address bar, or even the SSL certificate!

# Man-in-the-middle attacks

- Keyboard logging, interface illusions, and phishing are examples of man-in-the-middle attacks

- The website/program/system you're communicating with isn't the one you *think* you're communicating with

- A man-in-the-middle intercepts the communication from the user, and then passes it on to the intended other party
  - That way, the user thinks nothing's wrong, because his password works, he sees his account balances, etc.

# Man-in-the-middle attacks

- But not only is the man-in-the-middle able to see (and record) everything you're doing, and can capture passwords, but once you've authenticated to your bank (for example), the man-in-the-middle can hijack you session to insert malicious commands
  - Make a $700 payment to attacker@evil.com
- You won't even see it happen on your screen, and if the man-in-the-middle is clever enough, he can edit the results (bank balances, etc.) being displayed to you so that there's no visible record (to you) that the transaction occured
  - Stealthy, like a rootkit

# Nonmalicious flaws

- For the rest of this lecture, we'll look at flaws in systems that, although not inserted maliciously, and not inadvertent errors, can still be exploited to cause a failure

- We will look at two main sources of nonmalicious flaws:

  - Covert channels

  - Side channels

# Covert channels

- Suppose Alice has access to very sensitive information, and Eve is an attacker who wants it
  - Medical information
  - Banking information
  - Alice's own password
- Eve can even arrange for malicious code to be running on Alice's machine
  - Trojan horse, logic bomb, etc.

# Covert channels

- Normally, Eve would just have the Trojan horse send the sensitive data to her over the Internet

- But Alice is too clever for that!
  - She closely watches all Internet traffic from her computer
  - Better, she doesn't connect her computer to the Internet at all!

- How does Eve get Alice's data?

# Covert channels

- If there's no information at all that goes from Alice to somewhere Eve can get it, there's really nothing Eve can do.
  - But this is rare
- Suppose Alice publishes a weekly report summarizing some (nonsensitive) statistics
- Eve can “hide” the sensitive data in that report!
  - Modifications to spacing, wording, or the statistics itself
  - This is called a covert channel
  - See the text for an example (and Assignment 1)

# Side channels

- What if Eve can't get Trojaned software on Alice's computer in the first place?

- It turns out there are some very powerful attacks called side channel attacks
  - Eve watches how Alice's computer behaves when processing the sensitive data
  - Eve usually has to be somewhere in the physical vicinity of Alice's computer to pull this off
    - But not always!

# Side channels

- Eve can learn information about what Alice's computer is doing (and what data it is processing) by looking at:
  - RF emissions
  - Power consumption
  - Audio emissions
  - Reflected light from a CRT
  - Time it takes for Alice's computer to perform a computation
- These are especially powerful attacks when “Alice's computer” is a smart card (like a SIM chip or satellite TV card) that stores some kind of secret but is physically in Eve's possession

# Recap

- Other malicious code
  - Back doors
  - Salami attacks
  - Rootkits
  - Interface illusions
  - Keystroke logging
  - Man-in-the-middle attacks
- Nonmalicious flaws
  - Covert channels
  - Side channels

# Next time

- Controls against security flaws in programs
- Look at the stages of the software development lifecycle
- How to get the best chance of controlling all of the flaws?