Security

- Many kinds of threats:
  - destruction of data
  - unauthorized use of data
  - denial of service
- Security is an issue for entire system in its environment
- It is hard
  - People try to make it fail
  - Many targets: design, implementation, interactions.

Protection

- Goal is to ensure resources are used and accessed exactly as intended.
- Based on Policies
- Deals with factors internal to the computer system

Examples

- Trojan Horses
- Trap Door
- Buffer overflows
- Worms
- Viruses
- General

Trojan Horses

- Two different meanings
  - 1. A program that intends to run and do something undesirable
  - 2. User tricked into running it.
- e.g. text editor that “steals” content of files being edited as in downloads from a website
- (2) User tricked into running the wrong program.
  - Unix PATH variable includes “.”
  - Misspellings of common commands
- Fundamental problem is that privilege of command is determined by user ...

Trap Door

- A Trojan horse does bad things to an unspecting user.
- Lets a perpetrator do things they would normally not be allowed to do
  - e.g. when run by root
  - surprise response to a special input.
- e.g. special login program that allows author to login via a special user name.
- Usually obscure so a casual examination of the code will miss it.

Buffer Overflows

- Send a program input that is longer than expected.
- e.g. fixed size input buffers and no limit on size of input.
- Consequences:
  - crash
  - much worse when buffer is on the stack ...
- Very machine dependent but the X86 family is very general ...

Worms

- These are programs that replicate themselves from one system to another – usually via a network.
- They tend to consume resources leading to a Denial Of Service Attack.
- Arbitrarily bad behavior
- Usually spread by using legitimate access of the user they are running under (e.g. .rhosts)
- May use buffer overflows
- Mail to all users in the address book
- Modifies system parameters to restart the worm when rebooted (registry)

Viruses

- Not a free standing program, but a fragment attached to a legitimate program
- Essentially a dynamic Trojan horse or back door.
- Especially a problem on single user systems with weak or non-existent protection
  - makes it easy to infect system files
- Microsoft Office Macros and/or Active X lead to problems with Word files, email, and Web content.
  - no need to reboot
  - execution not even expected
- Direct Denial of Service often at the same time

General Problems

- Allocate virtual memory and look for data
- Illegal system calls, or system calls with the wrong parameters or number of parameters
- Hit “break” during login to bypass login check.
- Modify any OS data structures stored in user data space – eg. on an open() call.
- Look for “Don't do XXX” in the documentation.
- e.g. Unix lpr command on end of job
- e.g. mkdir x; rm x; ln /etc/passwd x;
- e.g. TENIX logins across pages
- e.g. OS/360 File opens
- Social Engineering

Protection Domains

- Process should have access to specific objects have the right to do specific things with each.
- Rights should change to reflect what is needed at the time
- At any time, a process is operating in a protection domain that determines its access rights.
- The domain should change to reflect actual needs.
- In most systems
  - domain changes are rare
  - more access than is needed is used.

Examples ...

- When you compile, should the compile have access to all your files?
  - Trojan horses and trapdoors frequently use such un-needed access.
- Often, protection domain == user
  - All processes beloning to a user have the same rights
- Changing protection domains
  - UNIX setuid so effective user id is the same as the file owner
  - Windows Server “execute as”
- Grants additional rights to specific programs, but not a solution to the first problem above.