

CS241 System Programming Security

Klara Nahrstedt

Lecture 26

3/29/2006



Content

- Security Environment
- Generic Security Attacks
- User Identification and Biometrics
- Design Principles

Administrative

- MP3 is posted, due April 3, 2006
- Quiz 7 is March 31, 2006
- Material covered in Quiz 7
 - R&R Chapter 4 and Chapter 5
 - Tanenbaum 5.3

Total Approach to Security

- External Security
- User Interface Security -- Establishing user identification and access rights.
- Internal Security -- Controls built into the hardware and software to ensure:
 - Reliable and uncorrupted operation of the system.
 - Integrity of programs and data

Internal Threats/Security

- *Data Confidentiality*
 - have secret data remain secret.
- *Data Integrity*
 - unauthorized users should not be able to modify any data without the owner's permission.
- *System Availability*
 - nobody can disturb the system to make it unusable (e.g., make sure that denial of service does not occur).
- *Privacy*
 - the system protects individuals from misuse of information
- The security system needs to protect against
 - *intruders (adversaries)*
 - *accidental data loss*

Internal Threats

- Security goals and threats

Goal	Threat
Data confidentiality	Exposure of data
Data integrity	Tampering with data
System availability	Denial of service

Intruders

- Common Categories
 - Casual prying by non-technical users
 - Snooping by insiders
 - Determined attempt to make money
 - Commercial or military espionage

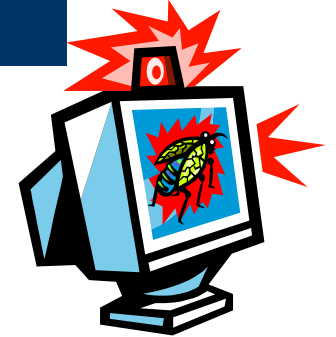
Attacks from Inside of the System

- **Trojan Horse**
 - seemingly innocent program contains code to perform an unexpected and undesirable function.
- **Examples**
 - Modifying, deleting or encrypting the user file; copying them into a place where cracker can retrieve them later, or even sending them to the cracker via email or FTP.
- One approach to do this is to place the program as a free, exciting new game, MP3 viewer, or something that attracts attention.
- The Trojan horse approach does not require the user to break into the computer.

Inside Attacks

- **Login Spoofing**
 - attacker writes a false login program that displays on the screen login prompt. This program asks for name, password, user types in login name and password. The false information is written to a file and the phony login program sends a signal to kill the shell. This action logs the attacker out and triggers the real login program. The user assumes that he/she wrote the wrong password and repeats the steps.
- **Logic Bombs**
 - build in bad behavior (e.g., erase a disk) into operating system if certain action is not taken. For example, as long the programmer feeds in a password every day, the behavior is not visible. When a programmer is fired, the password is not given and the bad behavior is triggered.
- **Trap Doors**
 - code is inserted into the system by the system programmer to bypass some normal check. For example, a login program could be written which allows a user to login independent of what password he/she types. The trap-door bypasses the whole authentication process.
- **Viruses and Worms**

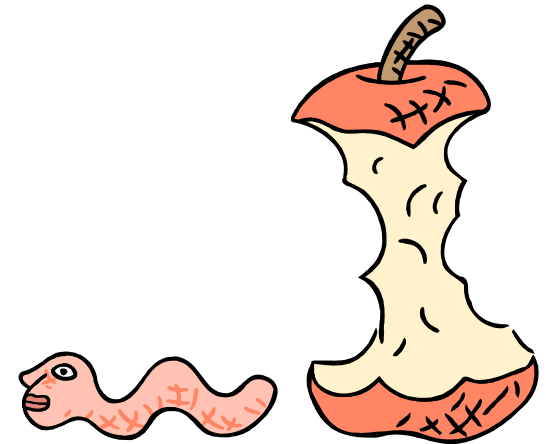
Virus



- Virus is
 - Malware
 - piece of code that can reproduce itself by attaching a copy of itself to another program
- Virus
 - Can cause Denial of Service (DOS) attack
 - Can cause Distributed Denial of Service (DDOS) attack
 - Can cause permanently damaged hardware

Internet Worm

- Free-standing program designed to travel between systems for some particular purpose.
- Consisted of two programs
 - bootstrap to upload worm
 - the worm itself
- Worm first hid its existence
- Next replicated itself on new machines



User Authentication

- **User Attributes:** Something about the person -- e.g., fingerprints, voice-prints, photographs, signatures.
- **User Possession:** Something possessed by the person -- e.g., badges, id cards, keys.
- **User Knowledge:** Something known by the person -- e.g., passwords, lock combinations, mother-in-law's maiden name.

Passwords

- People tend to choose easy-to-remember passwords, which are also easy to guess.
- Short passwords can be guessed by repeated trials of all possibilities.
- Passwords that are too long prompt people to write them down, which risks compromise by loss or theft of the note.
- The best passwords are of length 6-10 chars.
- **Avoid words that are in a dictionary.**
- Passwords made up of nonsense syllables are almost as secure as those made up of randomly chosen characters, but are easier to remember.

How Crackers Break-in?

- **Password guessing**
 - crackers compile potential common words as passwords, and use them to login.
- **War dialers**
 - dial telephone numbers and detect if security is in place (some PC systems don't have passwords).
- **Weak root password**
- **Script kiddies**
 - scripts found on the Internet, use brute force attacks to exploit bugs in specific programs.

Protect Your Passwords

- **One-way encrypt the password file**
 - UNIX designers are so confident of their one-way encryption scheme that the UNIX password file is "read permitted" to all users.
- Encourage users to change passwords often
- Limit the number of attempts to enter a password.
- The standard way to crack UNIX encryption
 - copy the password file to another machine
 - try encrypting the dictionary, permutations of common words, wife names, telephone numbers and comparing it against the password file contents.
- **"Salt" technique** (by Morris and Thompson):
 - associate an n-bit random number, called the **salt**, with each password. The random number is changed whenever the password is changed.

Summary

(General Design Principles of Security)

1. System design should be public
2. Default should be “no access”
3. Check always for current authority
4. Give each process the least privilege possible
 - Least privilege means that the process gets only necessary access to system components it needs to accomplish the given task
5. Protection mechanism should be simple, uniform and built into the lowest layers of the system
6. Chosen schemes must be psychologically acceptable