



TEKNILLINEN KORKEAKOULU

Security of One Computer

—

Hardware and Operating System



Hardware Possession and Control

- Generally computing takes place in some kind of computing hardware
- Typical active parts are
 - CPU
 - Memory
 - ROM, RAM, fixed and removable storage (disks)
 - I/O devices
 - Display, keyboard, communications, extension ports
- In most systems the hardware (and operating system) are considered trusted, as not trusting them would make the system pointless
 - There is also information that is not allowed to be processed on computers at all



Workstation Problems

- In traditional general purpose computing the authorized user is given access based on passwords or some other kind of user authentication
- However workstation computers are often in the complete physical control of the user
- Anybody who has access to the computing device may perform different kinds of actions on it
 - Steal the computer or parts of it
 - Reboot the computer from the user's own media and bypass the operating system security
 - Break physical security measures to access various parts of the device
 - E.g. remove the hard disk to be read on another computer or to add software, giving control to the user
 - Install eavesdropping devices inside the computer



Securing Workstation Computers

- Generally the computer must be protected from theft, vandalism, opening, booting with another media
- Sometimes an authorized user must be prevented from using transportable media devices
 - Disks, writable CD-ROMs
 - USB-, PCMCIA-, parallel port hard drives
 - Digital cameras
- Remember that the operating system access permissions do not prevent accessing files on disks, if that computer is booted with the attacker's own media or if the disks are removed and installed to another computer
- Solutions are in the family of physical security and often include computer case locks, chains, epoxy and video monitoring



Emission Security

- All electrical devices that cause variations in the flow of electricity create some kind of electromagnetic radiation
- General purpose computers are really radio transmitters with several transmission frequencies
 - CPU and bus clock
 - Peripheral transmission rates
 - Especially the video display (of CRT type)
- The basic transmissions are modulated by the data being processed
- The connections to the peripherals are often also antennas
 - Mouse and keyboard cords
- The attacker may extend the range of attack by using directional antenna



Controlling the Computer Emissions

- Tempest is a non-public US military standard that is often used to refer to computing devices with emission control
 - Roughly can be said to require enclosing the whole computer in a Faraday cage
 - Controlling the display emissions is a special problem
- Mostly used by the military
- Emission control is currently considered too expensive for most civilian applications
 - Always a subject to change in the future



Protection From Outside Emissions

- Inserting data (breaking) to a computing device by radiating energy to keyboard and mouse cords is not a known attack
- Destroying computing hardware with a strong electromagnetic pulse (EMP) is a known attack
 - Can be created by a nuclear detonation
 - Can be protected against by shielding like other emissions
- WLAN, Bluetooth, IR-ports etc. are possible venues for an attack
 - As the sender can use as much power as they wish, plain distance is not a protection



Tamper Resistant Hardware

- Problem: hardware is given to end users, but the contents should remain in the control of the owner or originator of the hardware
 - Telephone SIM cards
 - Smart cards (used for access, TV decoders, ID, money...)
 - Cryptographic password tokens (eg. SecurID)
 - Car computers
 - Public ATM machines
- One solution is tamper detection with e.g. seals
 - Especially if the problem domain allows rollback, meaning that the effects of the tampering can be reversed



- A plastic card with a CPU and non-volatile memory
- Can store information and perform low-end processing
- Draws power from the host terminal, often also a clock pulse
- Communicates with the host terminal
- Suomeksi: toimikortti



Smartcard Attacks

- The data on a smartcard (often a crypto key) can be extracted using several methods
 - Gaining access to the circuitry and by reading the data as it is transferred from the memory to the CPU
 - Requires shaving the protective layers of the card or careful drilling
 - Monitoring the power consumption of the card and deducing the key as it is used by clever mathematics (e.g. Chinese Remainder Theorem)
 - Interrupting the operation of the smart card by tampering with the operating voltages or the clock signal
- With simple money cards it is often enough to prevent change to the memory
 - The attacker can try to filter out the EEPROM write voltage
- Smartcards are getting better all the time, but they are not invulnerable



Protecting Data From Hardware Failures

- All mass media storage devices with moving parts are sure to fail
 - Backups are used to store the data in a secure location
 - Redundant Arrays of Inexpensive Disks (RAID) distribute the data to several physical disks using an error correcting code
 - Failure of a single disk should not cause any data loss
 - Beware of manufacturers with good quality control, multiple failures over a weekend are not unknown
- Power spikes or blackouts are no good for data
- All kinds of devices fail, e.g.. tape drive's writing heads
 - Experienced system administrators also read the tapes besides just writing to them



Backup Computing

- Part of *continuity* planning
- Backup media storage location is important
 - A physical location that is unaffected by anything that might destroy the main location is preferred
 - Physical security requirements are thus doubled
 - Bank vaults are usually a good choice for backup tapes
 - Magnetic media is sensitive to temperature
 - Ordinary safes are not enough to protect magnetic tapes from fire
- Entire backup computer systems are sometimes used
 - Expensive
 - Enable the business to continue almost immediately
 - Hot or cold backup



Secure Computing Architectures

- Multilevel secure systems can limit information flow between levels
 - The levels might be different operating system levels or
 - Different users or user groups
- Generally the CPU should provide protection to these levels
 - Some CPUs have complex multilevel structures
- However the protection achieved depends also on the operating system
 - Most general purpose operating systems do not provide very good security or do not use the features of the CPU
- Virtual machines are a low cost approach towards separating users/tasks/data from each other
 - However virtual machines are not unbreakable



Operating System Security Paradigm

- Current multipurpose OSes have two divisions for software to run in
 - Kernel
 - User space
- Any programming code in the kernel space has full access to the computer it is running on
- Code running in the user space has access rights based on the User ID it is running under
 - On Unix UID 0 is reserved for the super user or root and the kernel automatically gives this UID complete access
- Notice the difference between Unix kernel and root access
 - Kernel processes can access anything
 - Root processes can order the kernel to access anything



- A process is a data structure in the computer's memory
- It has
 - Program code to be executed
 - Data area
 - Associated data (owner, open files etc.)
- The operating system tells the CPU to execute the program code
- If the CPU architecture supports it, the process can usually access only its own memory area
 - Interprocess communications require that a specific mechanism is set up
- The process communicates to the kernel through system calls



Unix File System

- On the Unix system a file is a collection of data blocks represented by the *inode* of the file
- The inode is a data structure listing the owner, group access rights and other data for the file and the blocks which make up the file
- Directories are files of *type d* (directory) and they link file names to inode numbers
 - Thus on the Unix file system a file does not have a name, only its inode number
 - But the file has an owner and other properties
 - The inode structure keeps track of the number of links pointing to it, when the number reaches 0, the file is deleted



Unix Authorization

- A process is running under a specific user ID (UID) number, which refers to an user account in the system
 - And a group ID (GID)
- When the process tries to access some object via a system call, the kernel verifies the access
- The process UID is usually inherited from the process that creates (*spawns*) the new process
 - Initially set for the user's shell when a user logs in
- Defined in the `/etc/passwd` -file:
`kiravuo:x:22596:100:Timo Kiravuo:/u/kiravuo:/bin/bash`
 - There are various directory servers to replace `/etc/passwd`



Unix Access Rights

- The access rights for a file type object are typically stored in the inode for that file
- The rights are
 - Read to read a file or list a directory
 - Write to write to a file or to create a file in a directory
 - Execute to run a program or to access a file in a directory
- These rights are defined for the
 - User (owner) of the file
 - Group of the file
 - All other users



- An example

```
$ ls -ld /etc/passwd /tmp ~kiravuo ~kiravuo/foo
-rw-r--r--    1 root      other 1583265 Nov 26 23:05 /etc/passwd
drwxrwxrwt  389 root      root   35103 Nov 26 23:20 /tmp
drwx-----   8 kiravuo  users   4096 Nov 15 13:08 /u/kiravuo
drwxr-xr-x   2 kiravuo  users   4096 Mar 23  2006 /u/kiravuo/foo
```

- `/etc/passwd` is owned by the root, can be edited by the root and only read by others
- Anybody can create a file in the `/tmp` directory
- User kiravuo's home directory can only be accessed by him
- The directory `foo` can be read by anybody, except it is in kiravuo's directory, thus it can not be accessed, since Unix's file system is hierarchical



Suid and Sgid Mechanisms

- How to change your password?
 - The password file is owned by the root
- ```
$ ls -l `which passwd`
-r-sr-xr-x 1 root wheel 75636 Sep 27 11:12 /bin/passwd
```
- But the passwd program has the *set user ID* bit turned on
  - Which means that the program is executed under the root's UID and thus it can access the password file
  - This process is a trusted subject which can bridge the basic multilevel security
  - The security relies on the quality of the code for this program



# How to Attack the Operating System From the Inside?

---

- Assuming that an attacker is inside the system, they have access to the whole *security perimeter*
  - Security perimeter implies all the various protections, usually breaking any of them breaks the system's security
- The attacker can search for any *suid* program that might have a fault in it

```
find / -perm -4000 -o -perm -2000 -print
find / -perm +4000 -o -perm +2000 -print
find / -user root -perm +0022 -print
```
- The attacker can try to overload the system
  - Some systems do interesting things when certain filesystems start to fill up



- There are scanners, which search for known faults inside systems
  - Buffer overflows are a typical fault
  - Race conditions, e.g.:
    - find a (root) process which creates a temporary file and reads it back
    - After the process has checked that the file is only root readable, delete the file and replace with a symlink to a file you want to read
- ```
if (access(file, R_OK) != 0) {  
    exit(1);  
}  
fd = open(file, O_RDONLY);  
// do something with fd...
```
- Code from: Wikipedia.org, "Time-of-check-to-time-of-use"
- And so on



- Once inside and with the administration rights intruders typically install a *root kit*
 - Replaces standard system binaries with ones that hide the intrusion
 - Intruder's processes, files, log entries
 - System administrator should have correct binaries on non-writable media (CD, write protected diskette)
 - Preferably bootable media



Making the OS Secure

- As seen, many ways to break the OS security
- To make an OS secure from internal users
 - Limit the amount of actions available to users
 - Only the needed software
 - Perhaps only one application available
 - Keep the OS up to date
 - Study all the little details of the particular OS
 - Why no . in \$PATH, why no suid shell scripts...
 - Monitor the system
 - Learn the sounds of the server
 - Intrusion detection software



Looking at the Big Picture

- Once again, security comes from planning
- Specify the requirements with customers
 - Makes it easier to define what software is actually needed and anything else can be left out
 - Helps to avoid *creeping featurism* of growing requirements
- Make a realistic plan reflecting the likelihood of various threats realizing and their effects (risk analysis)
- Build defense in depth and limit the risk
 - If a SIM of a telephone company is broken, copied, re-engineered or whatever, the maximum potential risk is that some telephone calls are made from one telephone
 - The SIM is only used to authenticate the user
 - Multiple SIMs in the network will trigger the fraud detection



- Hardware in the user's hands can not be trusted completely
- Operating systems have many moving parts and only one of them has to fail to break security
- Besides good planning, as prof. Alastor Moody recommends, "Constant vigilance" is required from the system administrator