Operating Systems Security

Fred Long

based on slides originally written by Mike Tedd
and modified by Edel Sherratt

- There are many problems with shared and networked systems:
  - sensitive data
  - interference with data
  - destruction of data
  - viruses and worms, etc.
- The essential need is to be able to restrict access to data and facilities to *authorised* users
- Really secure systems put everything into metal cages (“Faraday cages”)
- Most systems cannot be protected like this

File Protection

Discretionary Security

- This refers to restricting access to files (directories, devices, etc.) by the owner granting or denying access
- In general, implementation of discretionary security needs lists of users and what they are allowed to access (*access control lists*) associated with each file
- Unix just has: read, write, execute for owner, group, world

Mandatory Security

- This is the type of security beloved by the military (and banks)
- All resources (files, etc.) are classified, i.e., marked with labels such as ‘secret’
- All agents (users, programs, etc.) have a security clearance, e.g., cleared to ‘secret’
- The labels form a partial ordering, e.g., ‘top secret’ is higher than ‘secret’
- There are rules like:
  read access => agent >= file
  write access => agent <= file

Encryption

- Encryption may be used to protect important data when it is transmitted, and sometimes when it is on disc
- The *Secure Socket Layer* (SSL) protocol may be used to secure Internet traffic
- *Pretty Good Privacy* (PGP) may be used to secure email

Programming Language Level Security

- Java has introduced security at the programming language level, and Java 2 has fine-grained security

User Authentication

- Usually, before using a system, a user must log on and be identified, typically by providing a password
- Unix encrypts passwords using a *one way* encryption algorithm
- Although the encryption is irreversible, hackers who grab the password file can try to guess likely passwords and see if they encrypt to any entries in the file

Super User

- Administrative functions like: backing up, disc check, machine shut-down, user registration, etc., must be protected
- This leads to the idea of *administrator privilege*
- Unfortunately, Unix has the concept of a *super user* who can do anything
- The super user password is very critical!

- Unix discretionary security being very weak leads to many functions needing to be run as super user, e.g., the mailer runs as super user in order to append to any user’s mailbox
- Loopholes and bugs in such code can be exploited by hackers to gain super user privileges
- The most often exploited bug is buffer overflow