

**COMP3301**  
**Lecture 11**  
**Security**

School of Information Technology and Electrical Engineering  
The University of Queensland

**This week**

- Look at security
- Exam Format
- Course Evaluations

**Security**

- What exactly do we mean?
  - inappropriate access of files
  - not allowing appropriate access to services
  - etc.
- What is protection?
  - similar concept
  - can refer to just the mechanism

**Threats**

- Goals:
  - Data confidentiality
  - Data integrity
    - includes adding or removing data
  - System availability
  - Privacy is also related
- Intruders:
  - casual user trying to read someone else's email?
  - intruders trying to get root access?
  - phishing attacks to make money?

**Malicious programs**

- Are also intruders
- Virus – attached to host and reproduces
  - can implement DDoS attacks
  - key loggers
- Worm – free-standing program
  - with mechanism to propagate on networks
- Trojan horse – seemingly genuine program which releases virus or other malice
- Logic Bomb
- Spyware – returns information

**Accidental Data Loss**

- External factors – fires, earthquake, power cuts
- Hardware or Software Errors
- Human Errors

## How can OSs be insecure?

- Some possible faults in OS:
  - Request memory or disk space, then read – has it been zeroed?
  - Are illegal system calls correctly handled?
  - Attempting to kill the password authentication process
  - etc.

## How can OSs be secure?

- From MULTICS designers (1975):
  - Public system design
  - Default = no access
  - Check for current authority
  - Each process gets lowest privileges possible
  - Protection should be simple, uniform, and in lowest layers
  - Acceptable to users

## User authentication

- Passwords
  - Can be weak (dictionary attacks)
  - UNIX passwords:
    - /etc/passwd
    - password encrypted, compared
    - salting attempts to foil dictionary attacks (Add random number to password)
  - one-time passwords
  - challenge-response
- Physical identification
  - key cards
  - fingerprints
  - blood samples?

## Protection mechanisms

- Concept of domain:
  - (object, rights)
  - often refers to a user
  - In UNIX, a system call traps to kernel mode, causing a domain switch

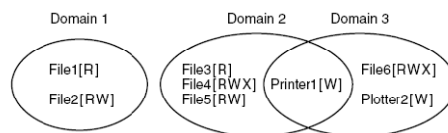


Figure 5-25. A protection matrix.

Domain	Object							
	File1	File2	File3	File4	File5	File6	Printer1	Plotter2
1	Read	Read Write						
2			Read	Read Write Execute	Read Write		Write	
3						Read Write Execute	Write	Write

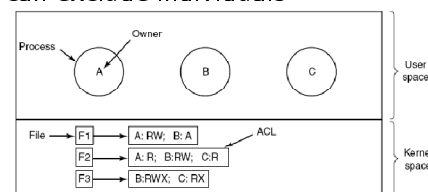
  

Figure 5-26. A protection matrix with domains as objects.

Domain	Object										
	File1	File2	File3	File4	File5	File6	Printer1	Plotter2	Domain1	Domain2	Domain3
1	Read	Read Write									Enter
2			Read	Read Write Execute	Read Write		Write				
3						Read Write Execute	Write	Write			

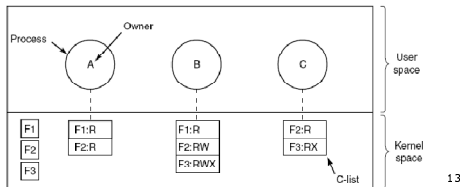
## Access Control Lists

- Each object has a list of allowed access
- Allows fine-grained control
- can exclude individuals



## Capabilities

- Opposite way round:
- Each user has a list of which operations it is capable of
- Possible to isolate untrusted code by given it few capabilities



13

## Security and real systems

- So how do real systems ensure their security?
- Lots of variations (some not so good)
- Some issues:
  - Does a publicly documented/open source system make it more secure?
  - Is Windows less secure simply because it's more popular?
    - if so, why hasn't Java been cracked many times?

14

## Follow-on Courses

- CSSE4004, Semester 1, Distributed Systems
- CSSE4003, Semester 1
  - Embedded Systems Design (Linux-based application development on Smart Phone)
- COMS3000, Semester 1 Information Security
- COMS4507, Semester 2 Advanced Information Security

15

## Exam Format

- One Hour + 5 minutes perusal
- 25%
- 25 marks
- Closed Book, but...
- ... 1 (double-sided) A4 sheet of notes allowed - handwritten or printed
- Any type of calculator allowed
- Sample paper on website, actual paper similar in format

16