Operating Systems

20. Protection

Paul Krzyzanowski

Rutgers University

Spring 2015

Protection & Security

- Security
  - Prevention of unauthorized access to a system
    - Prevent malicious or accidental access
    - “access” may be:
      - user login, a process accessing things it shouldn’t, physical access
    - The access operations may be reading, destruction, or alteration

- Protection
  - The mechanism that provides and enforces controlled access of resources to processes
  - A protection mechanism *enforces* security policies

Principle of Least Privilege

At each abstraction layer, every element (user, process, function) should be able to access ***only*** the resources necessary to perform its task

- Even if an element is compromised, the scope of damage is limited
- Consider:
  - Good: You cannot kill another user's process
  - Good: You cannot open the /etc/hosts file for writing
  - Good: Private member functions & local variables in functions limit scope
  - Violation: a compromised print daemon allows someone to add users
  - Violation: a process can write a file even though there is no need to
  - Violation: admin privileges set by default for any user account
- Least privilege is often difficult to define & enforce

Security Goals

- Authentication
  - Ensure that users, machines, programs, and resources are properly identified
- Integrity
  - Verify that data has not been compromised: deleted, modified, added
- Confidentiality
  - Prevent unauthorized access to data
- Availability
  - Ensure that the system is accessible

The Operating System

The OS provides processes with access to resources

Resource	OS component
Processor(s)	Process scheduler
Memory	Memory Management + MMU
Peripheral devices	Device drivers & buffer cache
Logical persistent data	File systems
Communication networks	Sockets

- Resource access attempts go through the OS
- OS decides whether access should be granted
  - Rules that guide the decision = *policy*

Domains of protection

- Processes interact with objects
  - Objects include:
    hardware (CPU, memory, I/O devices)
    software: files, processes, semaphores, messages, signals

- A process should be allowed to access only objects that it is authorized to access
  - A process operates in a **protection domain**
  - Protection domain defines the objects the process may access and how it may access them

Modeling Protection: Access Matrix

Rows: domains

Columns: objects

Each entry represents an access right of a domain on an object

objects

domains of protection

	F_0	F_1	Printer
D_0	read	read-write	print
D_1	read-write-execute	read	
D_2	read-execute		
D_3		read	print
D_4			print

Access Matrix: Domain Transfers

Switching from one domain to another is a configurable policy

A process in D_0 can switch to running in domain D_1

objects

domains of protection

	F_0	F_1	Printer	D_0	D_1	D_2	D_3	D_4
D_0	read	read-write	print	–	switch	switch		
D_1	read-write-execute	read			–			
D_2	read-execute				switch	–		
D_3		read	print					
D_4			print					

Implementing an access matrix

- A single table is usually impractical
  - Big size: # domains (users) × # objects (files)
  - Objects may come and go frequently

- Access Control List
  - Associate a column of the table with each object

Implementing an access matrix

- Access Control List
  - Associate a column of the table with each object

objects

domains of protection

	F_0	F_1	Printer	D_0	D_1	D_2	D_3	D_4
D_0	read owner	read-write	print		switch			
D_1	read-write-execute	read*			–			
D_2	read-execute				swtich	–		
D_3		read	print					
D_4			print					

ACL for file F_0

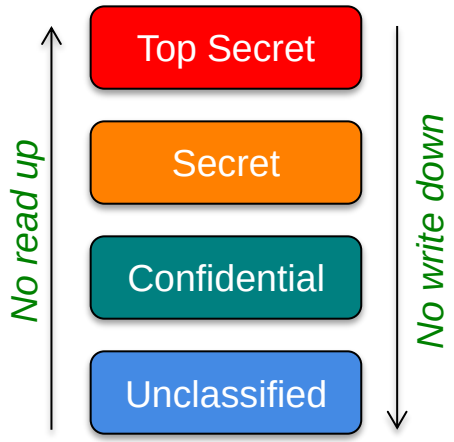
Access Control Models: MAC vs. DAC

- DAC: Discretionary Access Control
  - A subject (domain) can pass information onto any other subject
  - In some cases, access rights may be transferred
  - *Most systems use this*

- MAC: Mandatory Access Control
  - Policy is centrally controlled
  - Users cannot override the policy

Multi-level Access Control

- Typical MAC implementations use a Multi-Level Secure (MLS) access model
- Bell-LaPadula model
  - Identifies the ability to access and communicate data
  - Objects are classified into a hierarchy of sensitivity levels
    - Unclassified, Confidential, Secret, Top Secret
  - Each user is assigned a clearance
  - "No read up; no write down"
    - Cannot read from a higher clearance level
    - Cannot write to a lower clearance level
- Works well for government information
- Does not translate well to civilian life



Confidential cannot read Secret
Confidential cannot write Unclassified

The End