Access Control

COMP2700 Cyber Security Foundations

Slides prepared based partly on Chapter 5 of Gollmann's "Computer Security", Wiley, 2011

Outline

- Fundamental terminology
  - Principals & subjects, access operations
- Authentication & authorisation
- Access control structures:
  - Access control matrix
  - Capabilities & access control list
  - Discretionary & mandatory access control

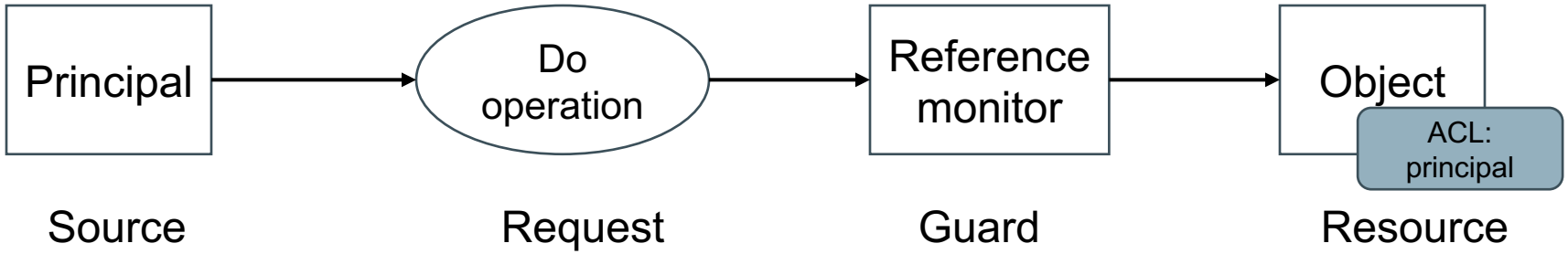
Access Control: Policy vs Mechanism

- A *security policy* is a statement of what is, and what is not, allowed.
- A *security mechanism* is a method, tool, or procedure for enforcing a security policy.
- Example:
  - Policy: A student is not allowed to sit in an exam on behalf of another student.
  - Mechanism: Id check during exam.

Security Policies

- Access control enforces operational security policies.
- A policy specifies who is allowed to do what.
- The active entity requesting access to a resource is called **principal**.
- The resource access is requested for is called **object**.
- **Reference monitor** is the abstract machine enforcing access control; guard mediating all access requests.

Authentication and Authorization



"If s is a statement, authentication answers the question 'Who said s?' with a principal. Thus principals make statements; this is what they are for. Likewise, if o is an object, authorization answers the question 'Who is trusted to access o?' with a principal."

B. Lampson, M. Abadi, M. Burrows, E. Wobber: Authentication in Distributed Systems: Theory and Practice, ACM Transactions on Computer Systems, 10(4), pages 265-310, 1992

Authentication and Authorization

- Authentication: reference monitor verifies the identity of the principal making the request.
- Authorisation: reference monitor decides whether access is granted or denied.
- Reference monitor has to find and evaluate the security policy relevant for the given request.

Users and User Identities

- Requests to reference monitor do not come directly from a **user** or a **user identity**, but from a **process**.
- In the language of access control, the process “speaks for” the user (identity).
- The active entity making a request within the system is called the **subject**.

Principals and Subjects

- A principal is an entity that can be granted access to objects or can make statements affecting access control decisions.
  - Example: user ID
- Subjects operate on behalf of (human users we call) principals; access is based on the principal's name bound to the subject in some unforgeable manner at authentication time.
  - Example: process (running under a user ID)

Access Operations and Access Rights

- On the most elementary level, a subject may
  - observe – look at the contents of an object, or
  - alter – change the contents of an object.
- Some fundamental policies can be expressed with these basic access modes.
  - For practical purposes a richer set of operations is more convenient.
- Access right (privilege/permissions): right to perform an (access) operation.

Access Rights: Bell-LaPadula model

- Bell-LaPadula model (see [Gollmann] chapter 11) has four access rights:
  - execute
  - read
  - append, also called blind write
  - write
- Mapping between access rights and access modes:

	execute	append	read	write
observe			X	X
alter		X		X

Access Rights: Bell-LaPadula model

- In a multi-user O/S, users open files to get access.
  - Files are opened for read or for write access so that the O/S can avoid conflicts like two users simultaneously writing to the same file.
- Write access usually includes read access.
  - A user editing a file should not be asked to open it twice; hence, write includes observe and alter mode.
- Few systems implement append.
  - Allowing users to alter an object without observing its content is rarely useful (exception: audit log).
- A file can be used without being opened (read).
  - Example: use of a cryptographic key; this can be expressed by an execute right that includes neither observe nor alter mode.

Access Rights: Unix/Linux

- Three access operations on files:
  - read: from a file
  - write: to a file
  - execute: a file
- Access operations on directories:
  - read: list contents
  - write: create or rename files in the directory
  - execute: search directory
- Deleting files/subdirectories handled by access operations in the directory.

Administrative rights

The rights to modify access rights, e.g.,

- Rights for creating and deleting files expressed by access control on the directory (Unix).
- Specific create and delete rights (Windows, OpenVMS).
- Specific rights like grant and revoke in database management.
- Rights to modify access control list in Windows.

Access Control Structures

- The structures used for capturing security policies.
- Two requirements:
  - It should help in expressing desired access control policy.
  - We should be able to check the intended policy has been captured correctly.
- Three basic structures:
  - Access control matrix
  - Capability list
  - Access control list

Access Control Matrix

- Access control matrix captures each combination of subject and object and their access rights.
  - Rows → subjects
  - Columns → objects
  - Entries → access operations
- Given an access matrix M, we write $M_{s,o}$ to mean the entry in M whose row corresponds to subject s and whose column corresponds to object o.
- Each entry $M_{s,o}$ contains a set of access rights of subject s for object o, e.g, read, write, and execute for files.

Example: a simple system

- Consider a system with two processes (subjects) P1 and P2, two memory segments (M1 and M2) and two files (F1 and F2).
- Each process has its own private segment and owns one file.
- Neither process can control the other process.
- Permitted access operations include: read (R), write (W), execute (E), and ownership (Own)

	M1	M2	F1	F2	P1	P2
P1	R,W,E		Own,R, W			
P2		R,W,E		Own,R,E		

Capabilities

- Focus on the subject
  - access rights stored with the subject
  - capabilities ≡ rows of the access control matrix
- Consider an access control matrix for principals Alice & Bob, and objects (files) ‘bill.doc’, ‘edit.exe’, ‘fun.com’.

	bill.doc	edit.exe	fun.com
Alice	-	{exec}	{exec,read}
Bob	{read,write}	{exec}	{exec,read,write}

- Capabilities associated with Alice is just a row in the access matrix:

Alice	edit.exe: {exec}	fun.com: {exec,read}

Access control list

- Access control list (ACL) mechanism focuses on the protection of objects.
  - access rights of principals stored with the object
  - ACLs = columns of the access control matrix

- Each object has an ACL, specifying the subjects (user IDs, user groups, etc) and the access rights of each of the subjects.

Example: ACL and access matrix

- Consider an access control matrix for principals Alice & Bob, and objects (files) ‘bill.doc’, ‘edit.exe’, ‘fun.com’.

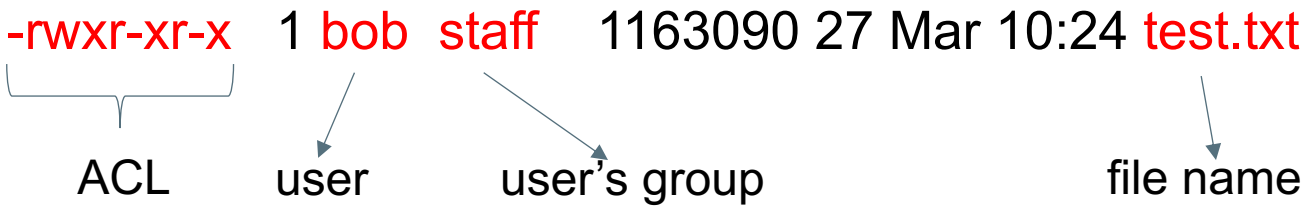
	bill.doc	edit.exe	fun.com
Alice	-	{exec}	{exec,read}
Bob	{read,write}	{exec}	{exec,read,write}

- An ACL for file ‘fun.com’ is just a column in the access matrix:

fun.com	Alice: {exec,read}	Bill: {exec,read,write}

Example: Unix file permission

- In Unix, each file has an ACL with three entries corresponding to:
  - The owner's access right
  - The access rights of all users in the owner's group
  - The access rights of all others.
- Example:



Access rights: r (read), w (write), x (execute)

Example: Unix file permission

- ACL is represented as a bit string.
  - Bit 0 is used for indicating file type, not related to access control.
  - Bit 1, 2 and 3, correspond, respectively, to read, write and execute rights of the user.
  - Bit 4, 5 and 6 correspond to read/write/execute rights for the user's group.
  - Bit 7, 8 and 9 corresponds o read/write/execute rights of all others.

Example: Unix file permission

- When a particular bit is set, it is displayed with its corresponding rights (e.g., r, w or x).

- For example: -rwxr-xr-x means:
  - User (bob) has read, write and execute rights
  - Every member of the user's group (staff) has read and execute rights
  - Everyone else has read and execute rights.

Ownership

Who is in charge of setting security policies?

- **Discretionary access control (DAC):** Define an owner for each resource and let the owner sets the policies.
  - Adopted in most modern operating systems.
  - Focus on user identities – sometimes also called identity-based access control (IBAC).
- **Mandatory Access Control (MAC):** Impose system-wide policies on who are allowed to access what.
  - Policies refer to security labels of objects, e.g., confidential, top secret.
  - Mostly used in the defence sector.

Intermediate Controls

All problems in computer science can be solved by another layer of indirection. -David Wheeler

- For large systems/organisations, intermediate layers can be introduced between subjects and objects to create more manageable policies.
- Examples:
  - Grouping of users
  - Grouping of procedures into *roles* -- role-based access control (RBAC)
  - Introduce hierarchies into access control, e.g., privilege level.

Summary

- Basic terminologies in access control.
- Access control involves authentication and authorization.
- Access control matrix serves as a reference data structure.
  - In practice different methods are used to represent the access control matrix.
- Different paradigms of access control:
  - Centered on identity (IBAC, RBAC), or systems (MAC).

Further Reading

- D. Denning. “Cryptography and Data Security”, Addison-Wesley, 1983. Chapter 4 (Access Control).
  - http://faculty.nps.edu/dedennin/publications/Denning-CryptographyDataSecurity.pdf

- R. Sandhu, D. Ferraiolo, and R. Kuhn: *The NIST Model for Role-Based Access Control: Towards a Unified Standard*, Proceedings of the 5th ACM Workshop on Role-Based Access Control, Berlin, Germany, July 26-27, 2000
  - http://csrc.nist.gov/rbac/sandhu-ferraiolo-kuhn-00.pdf