Module 3: Access Control
CSC 3207: Computer Security

Department of Computer Science

Outline

- Access control principles - subjects, objects, access rights
- Access control mechanisms and types
- Access control lists (ACLs)
- Access control matrix; capabilities and capability tickets
- Reference Books
  - Chapters 2 and 14; Introduction to Computer Security by Matt Bishop
  - Chapters 2 and 16; Computer Security: Art and Science by Matt Bishop
  - Chapters 4 and 5; Computer Security by Dieter Gollmann

Access Control

- The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner
- The process of granting or denying specific requests to
  - obtain and use information and related information processing services
  - enter specific physical facilities
- A process by which use of system resources is regulated according to a security policy and is permitted only by authorized entities (users, programs, processes, or other systems) according to that policy
- A central element of computer security

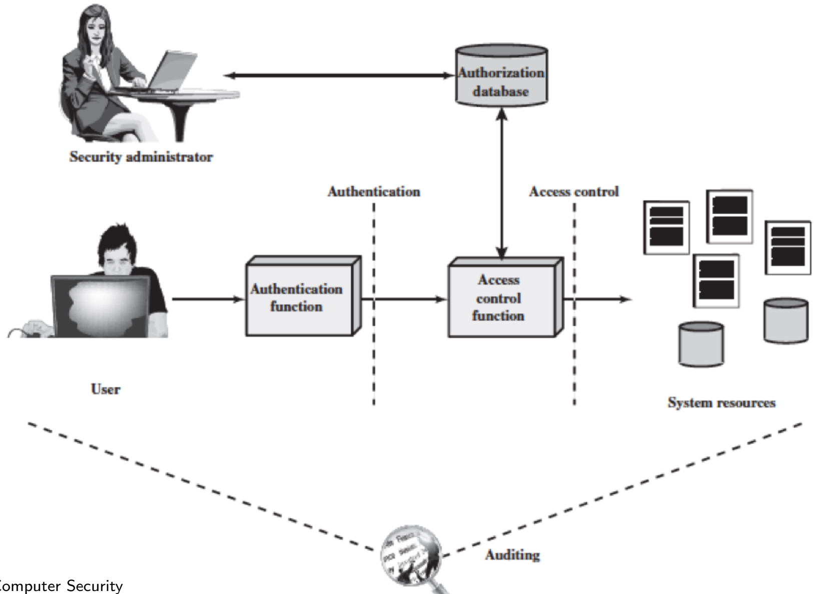
Access Control

- principal objectives of computer security
  - prevent unauthorized users from gaining access to resources
  - prevent legitimate users from accessing resources in an unauthorized manner
  - enable legitimate users to access resources in an authorized manner
- assume have users and groups
  - authenticate to system
  - assigned access rights to certain resources on system
- Access control implements a security policy that specifies who or what may have access to each specific system resource and the type of access that is permitted in each instance

Access Control Context

- **Authentication:** Verification that the credentials of a user or other system entity are valid.
- **Authorization:** The granting of a right or permission to a system entity to access a system resource. This function determines who is trusted for a given purpose.
- **Audit:** An independent review and examination of system records and activities in order to
  - test for adequacy of system controls
  - ensure compliance with established policy and operational procedures
  - detect breaches in security
  - recommend any indicated changes in control, policy and procedures.

Relationship Among Access Control and Other Security Functions



Access Control Context

- An access control mechanism mediates between a user (or a process executing on behalf of a user) and system resources, such as applications, operating systems, firewalls, routers, files, and databases.
- The system must first authenticate an entity seeking access.
- The authentication function determines whether the user is permitted to access the system at all.
- The access control function determines if the specific requested access by this user is permitted.
- A security administrator maintains an authorization database that specifies what type of access to which resources is allowed for this user.
- The access control function consults this database to determine whether to grant access.
- An auditing function monitors and keeps a record of user accesses to system resources.

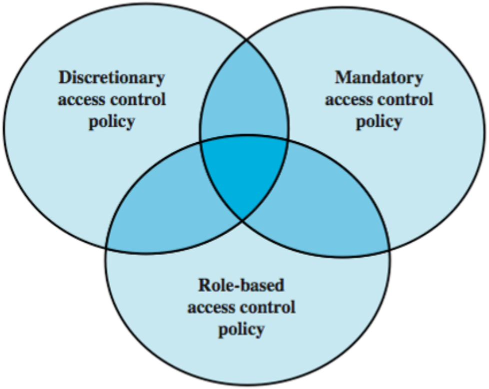
Access Control Context

- In the simple model, the access control function is shown as a single logical module.
- In practice, a number of components may cooperatively share the access control function.
- All operating systems have at least a rudimentary, and in many cases a quite robust, access control component.
- Add-on security packages can supplement the native access control capabilities of the operating system.
- Particular applications or utilities, such as a database management system, also incorporate access control functions.
- External devices, such as firewalls, can also provide access control services.

Access Control Policies

- An access control policy dictates what types of access are permitted, under what circumstances, and by whom.
- Can be embodied in an authorization database
- Access control policies are generally grouped into the following categories
  - Discretionary access control (DAC)
  - Mandatory access control (MAC)
  - Role-based access control (RBAC)
  - Attribute-based access control (ABAC)
- These four policies are not mutually exclusive. An access control mechanism can employ two or even all three of these policies to cover different classes of system resources.

Access Control Policies


Discretionary access control policy
Mandatory access control policy
Role-based access control policy

Access Control Policies

- Discretionary access control (DAC): Controls access based on the identity of the requestor and on access rules (authorizations) stating what requestors are (or are not) allowed to do.
- DAC is termed discretionary because an entity might have access rights that permit the entity, by its own volition, to enable another entity to access some resource.
- DAC is the traditional method of implementing access control

Access Control Policies

- Mandatory access control (MAC): Controls access based on comparing security labels (which indicate how sensitive or critical system resources are) with security clearances (which indicate system entities are eligible to access certain resources).
- MAC is termed mandatory because an entity that has clearance to access a resource may not, just by its own volition, enable another entity to access that resource.
- MAC is used in environments where information classification and confidentiality is very important (e.g., the military)

Access Control Policies

- Role-based access control (RBAC): Controls access based on the roles that users have within the system and on rules stating what accesses are allowed to users in given roles.
- Attribute-based access control (ABAC): Controls access based on attributes of the user, the resource to be accessed, and current environmental conditions.

Access Control Requirements

- reliable input: a mechanism to authenticate
- fine and coarse specifications: regulate access at varying levels (e.g., an attribute or entire DB)
- least privilege: minimum authorization to do its work
- separation of duty: divide steps among different individuals
- open and closed policies: accesses specifically authorized or all accesses except those prohibited
- administrative policies: who can add, delete, modify rules

Access Control Elements

- The basic elements of access control are: subject, object, and access right.
- subject - entity that can access objects
  - a process representing user/application
  - often have 3 classes: owner, group, world
- object - access controlled resource e.g. files, directories, records, programs etc
  - number/type depend on environment
- access right - way in which subject accesses an object
  - e.g. read, write, execute, delete, create, search

Access Control Elements

- Basic access control systems typically define three classes of subject, with different access rights for each class
- Owner: This may be the creator of a resource, such as a file.
- For system resources, ownership may belong to a system administrator.
- For project resources, a project administrator or leader may be assigned ownership.
- Group: In addition to the privileges assigned to an owner, a named group of users may also be granted access rights, such that membership in the group is sufficient to exercise these access rights.
- In most schemes, a user may belong to multiple groups.
- World: The least amount of access is granted to users who are able to access the system but are not included in the categories owner and group for this resource.

Access right

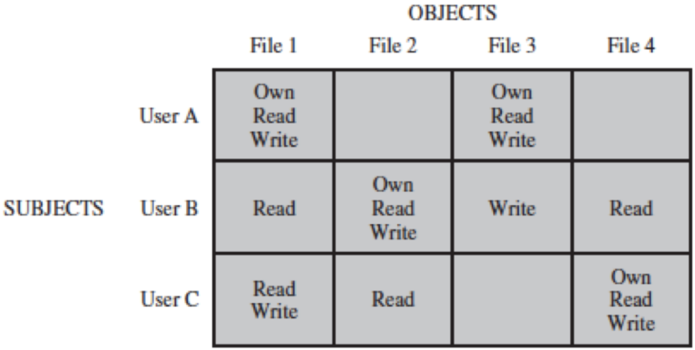
An access right describes the way in which a subject may access an object. Access rights could include

- Read: User may view information in a system resource (e.g., a file, selected records in a file, selected fields within a record, or some combination).
  - Read access includes the ability to copy or print.
- Write: User may add, modify, or delete data in system resource (e.g., files, records, programs).
  - Write access includes read access.
- Execute: User may execute specified programs.
- Delete: User may delete certain system resources, such as files or records.
- Create: User may create new files, records, or fields.
- Search: User may list the files in a directory or otherwise search the directory.

Discretionary Access Control

- one in which an entity may be granted access rights that permit the entity, by its own volition, to enable another entity to access some resource.
- A general approach to DAC, as exercised by an operating system or a database management system, is that of an access matrix
- One dimension of the matrix consists of identified subjects that may attempt data access to the resources.
- this list will consist of individual users or user groups, including terminals, network equipment, hosts, or applications instead of or in addition to users.
- The other dimension lists the objects that may be accessed.
- At the greatest level of detail, objects may be individual data fields. More aggregate groupings, such as records, files, or even the entire database, may also be objects in the matrix.
- Each entry in the matrix indicates the access rights of a particular subject for a particular object.

Access Matrix



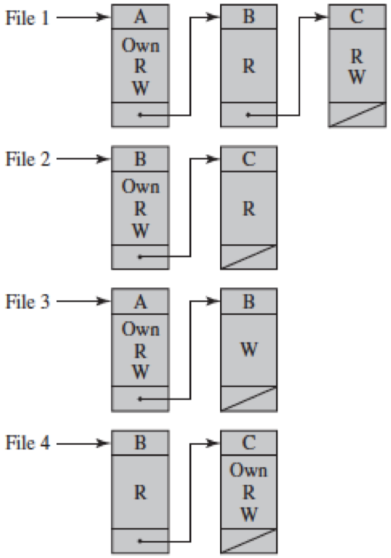
		OBJECTS			
		File 1	File 2	File 3	File 4
SUBJECTS	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

(a) Access matrix

Access Control Lists

- The matrix may be decomposed by columns, yielding access control lists (ACLs)
- For each object, an ACL lists users and their permitted access rights.
- The ACL may contain a default, or public, entry which allows users that are not explicitly listed as having special rights to have a default set of rights.
- The default set of rights should always follow the rule of least privilege or read-only access, whichever is applicable.
- Elements of the list may include individual users as well as groups of users.
- When it is desired to determine which subjects have which access rights to a particular resource, ACLs are convenient, because each ACL provides the information for a given resource.
- ACL is not convenient for determining the access rights available to a specific user.

Access Matrix



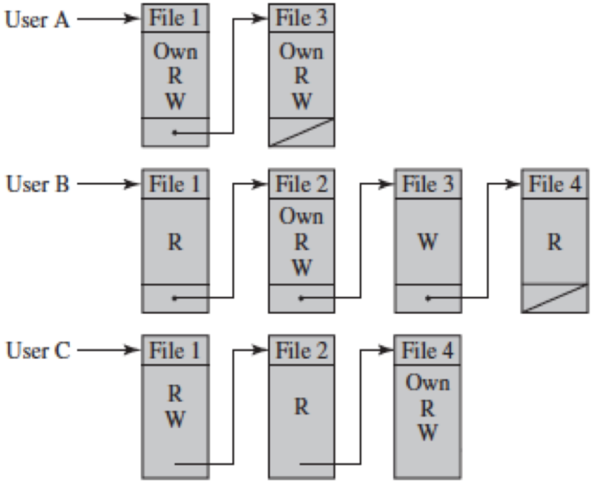
Capability Tickets

- Decomposition by rows yields capability tickets.
- A capability ticket specifies authorized objects and operations for a particular user.
- Each user has a number of tickets and may be authorized to loan or give them to others.
- Because tickets may be dispersed around the system, they present a greater security problem than access control lists.
- The integrity of the ticket must be protected, and guaranteed (usually by the operating system).

Capability Tickets

- The ticket must be unforgeable: have the operating system hold all tickets on behalf of users in a region of memory inaccessible to users.
- An unforgeable token can be included in the capability as a large random password, or a cryptographic message authentication code. This value is verified by the relevant resource whenever access is requested.
  - Appropriate for use in a distributed environment, when the security of its contents cannot be guaranteed.
- Easy to determine the set of access rights that a given user has, but more difficult to determine the list of users with specific access rights for a specific resource.

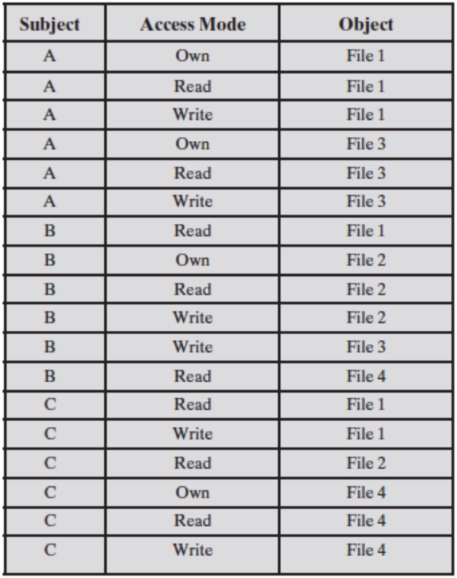
Capability Tickets



Authorization Table

- a data structure that is not sparse, like the access matrix
- more convenient than either ACLs or capability lists
- An authorization table contains one row for one access right of one subject to one resource.
- Sorting or accessing the table by subject is equivalent to a capability list.
- Sorting or accessing the table by object is equivalent to an ACL.
- A relational database can easily implement an authorization table of this type.

Authorization Table



Subject	Access Mode	Object
A	Own	File 1
A	Read	File 1
A	Write	File 1
A	Own	File 3
A	Read	File 3
A	Write	File 3
B	Read	File 1
B	Own	File 2
B	Read	File 2
B	Write	File 2
B	Write	File 3
B	Read	File 4
C	Read	File 1
C	Write	File 1
C	Read	File 2
C	Own	File 4
C	Read	File 4
C	Write	File 4

Access Control Model

- a general model for DAC
- The model assumes a set of subjects, a set of objects, and a set of rules that govern the access of subjects to objects.
- the protection state of a system to be the set of information, at a given point in time, that specifies the access rights for each subject with respect to each object.
- three requirements: representing the protection state, enforcing access rights, and allowing subjects to alter the protection state in certain ways.
- The model addresses all three requirements, giving a general, logical description of a DAC system.

Access Control Model

- To represent the protection state, we extend the universe of objects in the access control matrix to include
- Processes: Access rights include the ability to delete a process, stop (block), and wake up a process.
- Devices: Access rights include the ability to read/write the device, to control its operation (e.g., a disk seek), and to block/unblock the device for use.
- Memory locations or regions: Access rights include the ability to read/write certain regions of memory that are protected such that the default is to disallow access.
- Subjects: Access rights with respect to a subject have to do with the ability to grant or delete access rights of that subject to other objects

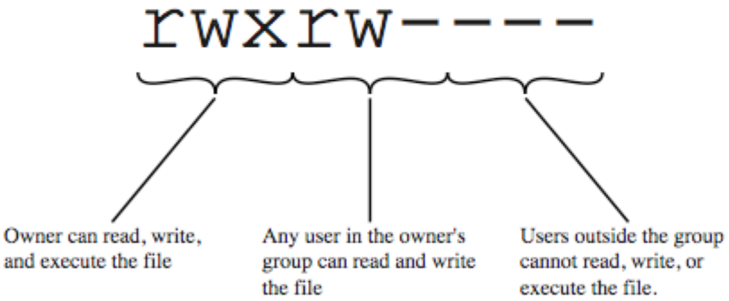
Protection Domains

- The access control matrix model that we have discussed so far associates a set of capabilities with a user.
- A more general and more flexible approach is to associate capabilities with protection domains.
- A protection domain is a set of objects together with access rights to those objects.
- In terms of the access control matrix, a row defines a protection domain.
- In this limited model, each user has a protection domain, and any processes spawned by the user have access rights defined by the same protection domain.

UNIX file Concepts

- UNIX files administered using inodes
- An inode (index node) is a control structure that contains the key information needed by the operating system for a particular file.
- may have several names for same inode
- have inode table / list for all files on a disk
  - copied to memory when disk mounted
- directories form a hierarchical tree
  - may contain files or other directories
  - are a file of names and inode numbers

UNIX File Access Control


rwxrw----
Owner can read, write, and execute the file
Any user in the owner's group can read and write the file
Users outside the group cannot read, write, or execute the file.

UNIX File Access Control

- "set user ID" (SetUID) or "set group ID" (SetGID)
  - system temporarily uses rights of the file owner / group in addition to the real user's rights when making access control decisions
  - enables privileged programs to access files / resources not generally accessible
- sticky bit - on directory limits rename/move/delete to owner
- superuser is exempt from usual access control restrictions

UNIX File Access Control

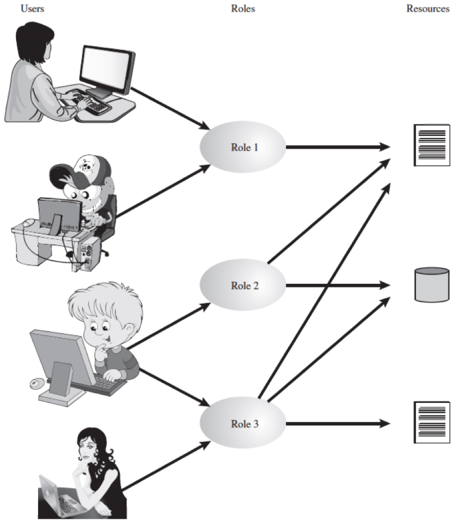
- modern UNIX systems support ACLs
- can specify any number of additional users / groups and associated rwx permissions
- group perms also set max ACL perms
- when access is required
  - select most appropriate ACL - owner, named users, owning / named groups, others
  - check if have sufficient permissions for access

Role-based Access Control

- Associate permissions with job functions within an organization
  - Each job defines a set of tasks
  - The tasks need permissions
  - The permissions define a role
- Bank Teller
  - Read/Write to client accounts
  - Cannot create new accounts
  - Cannot create a loan
- Role defines only the permissions allowed for the job
- Traditional DAC systems define the access rights of individual users and groups of users.
- Based on the roles that users assume in a system rather than the user's identity
- Assign access rights to roles instead of individual users

Role-based Access Control

Access based on ‘role’, not identity



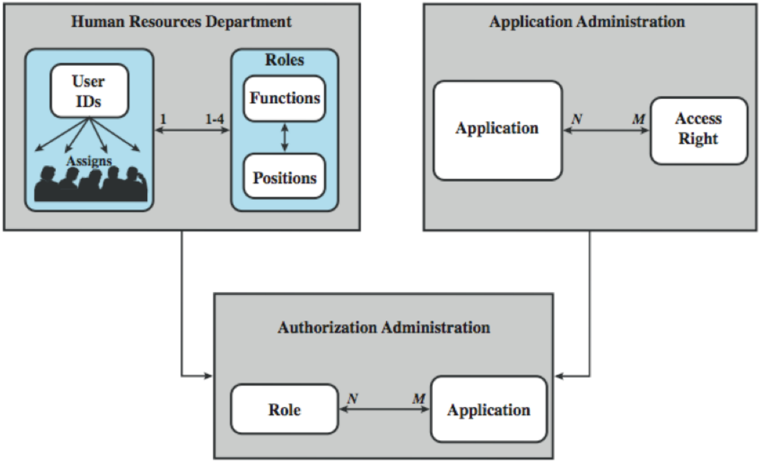
Role-based Access Control Model

- Model consists of two relationships
  - Role-permission assignments
  - User-role assignments
- Assign permissions to roles
  - These are largely fixed
- Assign a user to the roles they can assume
  - These change with each user
  - Administrators must manage this relationship

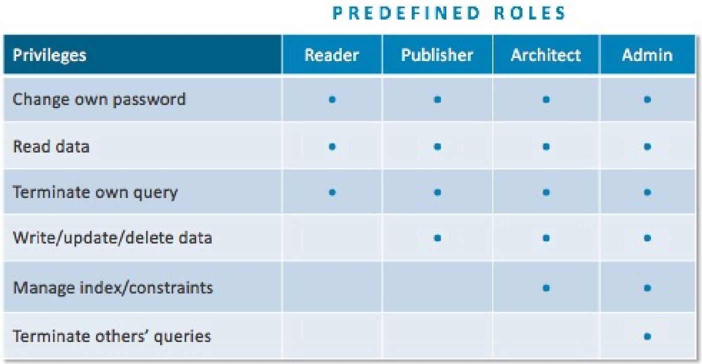
Role hierarchy

- role hierarchy: reflect hierarchical structure of roles in an organization
- job function with higher responsibility means greater authority - project manager, team leader, senior programmer
- roles can inherit access rights from multiple subordinate roles

Example: Bank Role-based Access Control



Example: Role-based Access Control



PREDEFINED ROLES

Privileges	Reader	Publisher	Architect	Admin
Change own password	●	●	●	●
Read data	●	●	●	●
Terminate own query	●	●	●	●
Write/update/delete data		●	●	●
Manage index/constraints			●	●
Terminate others' queries				●