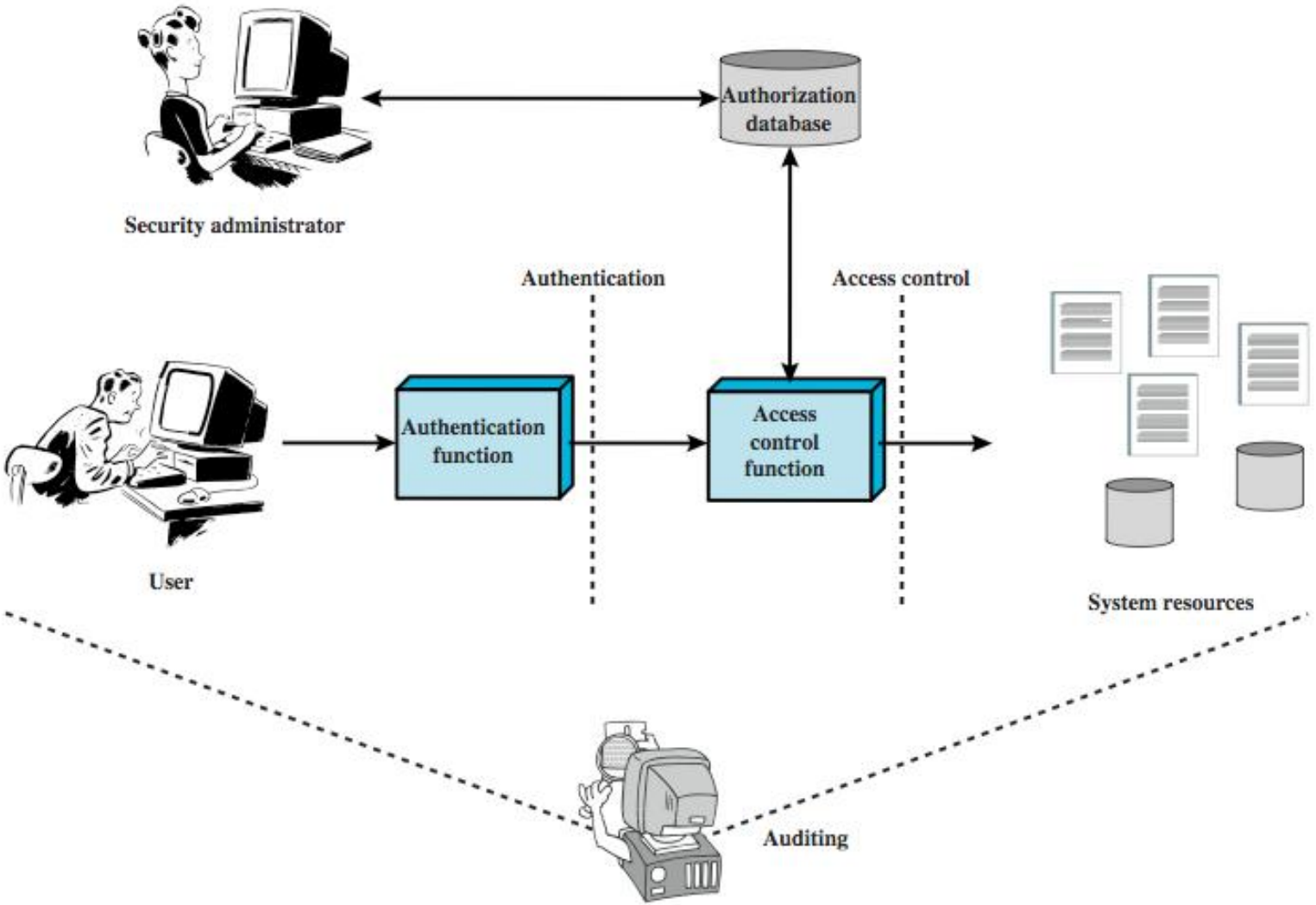
Access Control

Introduction to Computer Security

Week 9 - week 10

Access Control Principles



Overview

- Definition: "The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner"
- Central element of computer security
- Assume have users and groups
  - users authenticate to system
  - assign users access rights to certain resources on system

Access Control Policies

- **Discretionary access control (DAC):** based on the identity of the requester and access rules (e.g., in UNIX based systems, the owner of a file controls the access of that file)
- **Mandatory access control (MAC):** based on comparing security labels with security clearances
  - Meaning of **mandatory**: one with access to a resource cannot pass to others
  - Often applied in military systems where the owner of a file may not be allowed to freely pass the file to another user, or even not allowed to pass message to another user
  - This is in underlying foundation for the design of early day MULTICS — the precursor of UNIX
- **Role-based access control (RBAC):** based on user roles

Access Control Requirements

- Reliable input: a mechanism to authenticate (assuming a user is authentic, e.g., user logged in, or IP address based access control)
- Support for fine and coarse specifications: regulate access at varying levels (e.g., fine-grained at individual user level, coarse-grained for certain roles or resources)
- Least privilege: by default, grant a user minimal authorization required to do his work
- Separation of duty: divide steps among different individuals (e.g., two bank clerks/administrators are required to process a large amount transaction)
- Open and closed policies: accesses specifically authorized or all accesses except those prohibited (blacklisting vs whitelisting)

Access Control Elements

- Subject: entity that can access objects
  - a process representing user/application
  - In UNIX file access control: owner, group, others
- Object: access controlled resource
  - files, directories, records, executable programs
  - number/type depend on environment
- Access right: ways in which subject accesses an object
  - read, write, execute, delete, create, search

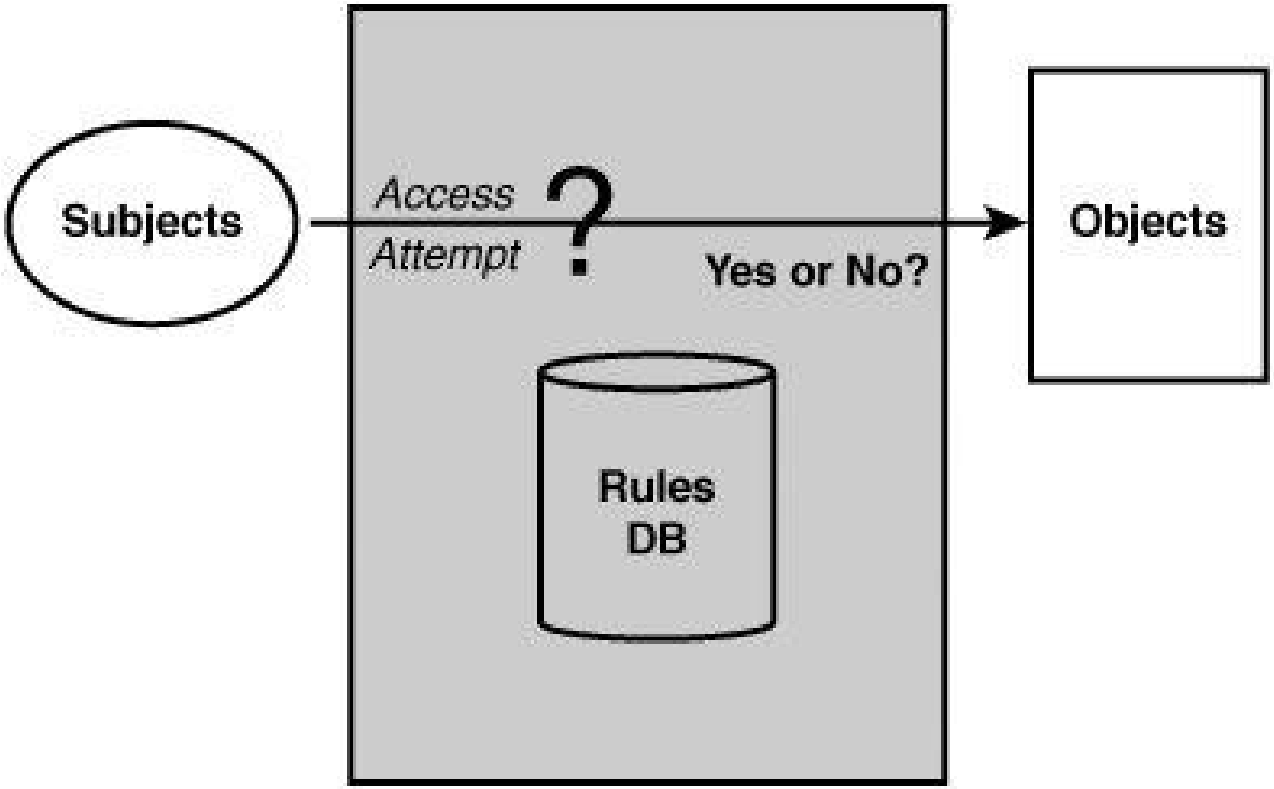
Access Control Model

- Subjects are also objects
- Extend the universe of objects to include processes, devices, memory locations . . .

		OBJECTS								
		subjects			files		processes		disk drives	
		S_1	S_2	S_3	F_1	F_1	P_1	P_2	D_1	D_2
SUBJECTS	S_1	control	owner	owner control	read *	read owner	wakeup	wakeup	seek	owner
	S_2		control		write *	execute			owner	seek *
	S_3			control		write	stop			

* - copy flag set

Access Control Mechanism



Access Control Matrix

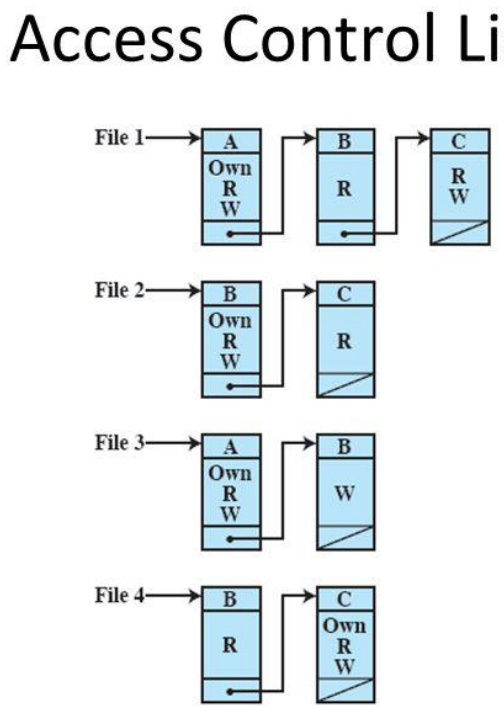
	Afile	Bfile	Cfile	Dfile
Anna	rwx	r	x	
Bill	r	rx	x	
Charles	r	r	rw	r
Damian		r		rw

- Access control matrix is the simplest, most general model
- $M : Subjects \times Objects \rightarrow \mathcal{P}(Action)$
- Subject S is allowed to request action A on object O iff $A \in M(S, O)$
- Access control matrix M represents the protection state of a system (protection state may change over time)
- Useful abstraction, but not very practical

Access Control List (ACL)

- A list of access rights associated with an object
- ACLs are a practical way to represent the access control matrix one row of the matrix is stored for each object. For example, we may specify
  - File1.txt: Alice:{*read*, *write*}, Bob:{*read*}
  - Socket s: Proces 9876: {*open*, *read*, *write*, *close*}
- Saves space for empty cells if no privilege is granted

Access Control List Example



- File 1 owned by A, readable by B, readable and writable by C
- File 2 owned by B, readable by C

Capabilities

- Acess rights associated with a subject
- Another way to represent Access Control matrix (Columns vs Rows)
- Example:
  - Alice's capabilities: file1.txt:$\{r, w\}$; file2.txt:$\{r\}$
  - Bob's capabilities: file1.txt:$\{read\}$
  - Process 4567 capabilities: file1.txt: $\{open, read, write, close\}$

Discretionary Access Control

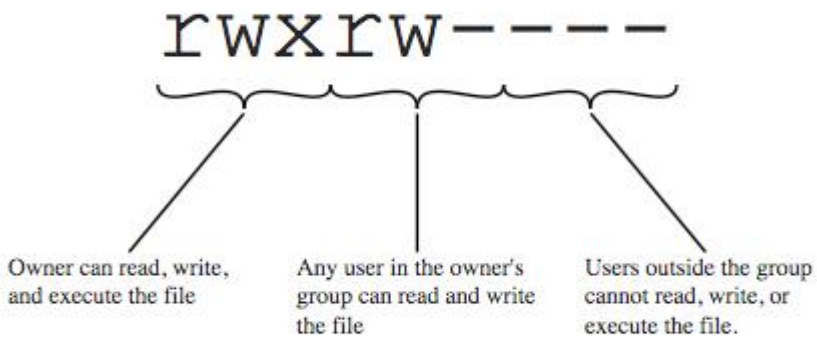
- Data owners set access rights
- Subjects are trusted to make decisions about sharing access rights
  - File owners or creator decides who is allowed to access it
  - User or process that can read a secret file may also share it
- Typical in commercial and consumer systems
- Access may be audited
- Vulnerable to owner mistakes

UNIX File Concepts

- UNIX files administered using inodes (index nodes)
- An inode:
  - control structure with key information on a file (attributes, permissions, . . .)
  - on a disk: an inode table for all files
  - when a file is opened, its inode is brought to RAM
- Directories form a hierarchical tree
  - may contain files or other directories
  - are a file of names and inode numbers

UNIX File Access Control

- Unique user identification number (user ID)
- Member of a primary group identified by a group ID
- 12 protection bits
  - 9 specify read, write, and execute permission for the owner of the file, members of the group and all other users
  - 2 specify SetID, SetGID
  - 1 is the sticky bit (only owner can remove, delete, . . . , a directory)
- The owner ID, group ID, and protection bits are part of the file's inode



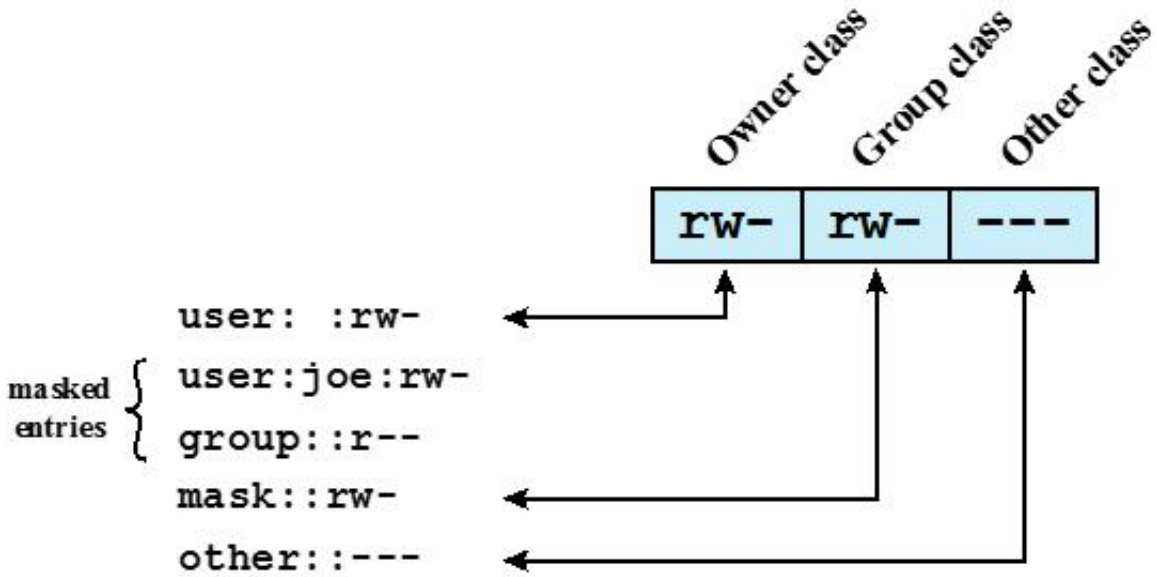
UNIX File Access Control

- "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 (administrator) is exempt from usual access control restrictions

UNIX Access Control Lists

- Modern UNIX systems support ACLs
- Can specify any number of additional users/groups and associated rwx permissions
- When access is required
  - select most appropriate ACL subject: owner, named users, owning/named groups, others
  - check if have sufficient permissions for access

UNIX extended access control list



(b) Extended access control list

- Used to specify access control rules for extra individuals and groups

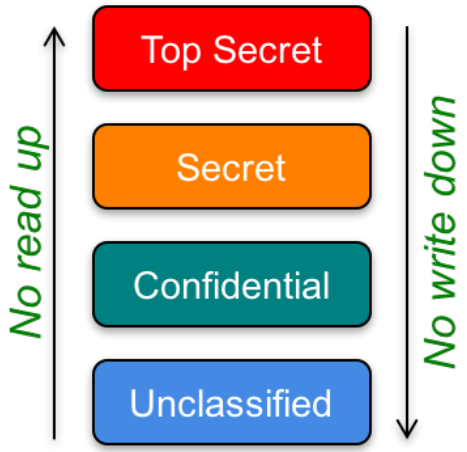
Mandatory Access Control (MAC)

- Access rights based on the policy set by administration
- The AC policy is enforced by the reference monitor and cannot be changed by users (thus not vulnerable to human mistakes)
- Subject cannot leak access rights to others
  - Users who read a secret file cannnot copy, print or email
  - file viewer application prevents cut-and-paste and screen shots
- MAC originates from military policies
  - Intelligence officer may not be allowed to read his own reports
  - Officer can read a secret document but cannot take a copy out of the room
  - Officer who has had contact with foreign agents may lose access to classified information

Mandatory Access Control – Bell-LaPadula Model

- Developed in 1970s
- Initially funded by US Department of Defense, thus very influential
- Has a formal model for access control
- Form a hierarchy and are referred to as security levels
- A subject has a security clearance
- An object has a security classification
- Security classes control the manner by which a subject may access an object

Mandatory Access Control – Bell-LaPadula Model



- Security levels arranged in linear ordering:
  Top Secret (highest), Secret, Confidential,
  Unclassified (lowest)
- Security classes control the manner by which a subject may access an object:
  - Lower level users cannot read higher level
  - Higher level users cannot write to lower level

Clark-Wilson Integrity Model

- widely used in the commercial world
- Data item classified as **constrained data items (CDI)** and **unconstrained data items (UDI)**
- A set of **integrity constraints** for the values in CDIs
- **Transformation Procedures (TPs)** are used to change CDIs
- **Integrity Verification Procedures (IVPs)** assure all CDIs conform to integrity rules
- Two main concepts
  - **Well-formed transactions**: a user can only manipulate data in constrained ways
  - **Separation of duty**: one who creates or certifies a well-formed transaction may not be allowed to execute it

Some Certification and Enforcement Rules in C-W

C1: Integrity Verification Procedures (IVPs) must ensure that all CDIs are in valid states

C2: All Transformation Procedures (TPs) must be certified (must take a CDI from a valid state to a valid final state)

E1: The system must maintain a list of relations specified in C2, and only TPs are allowed to modify CDIs

E2: The system must maintain a relation that for users, TPs and CDIs . . . , such that only executions described by the relation are allowed to perform.

C3: The list of relations in E2 must be certified to meet the separation of duty requirement

C4: All Transformation Procedures (TPs) must be certified

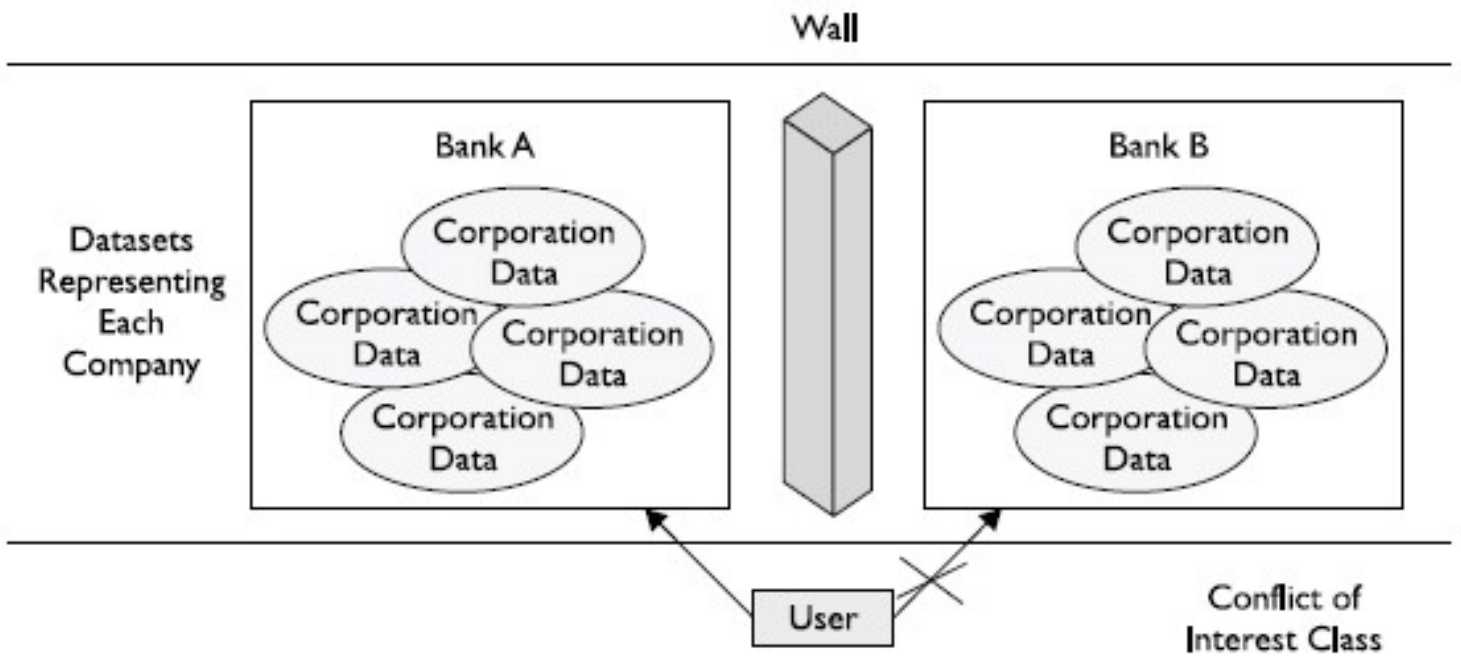
The Chinese Wall Model

- Hybrid model: addresses integrity and confidentiality
- Addresses conflict of interest (CI or CoI)
- Model elements
  - subjects: active entities (users) interested in accessing protected objects
  - objects: individual data items, each about a corporation
  - datasets (DS): all objects concerning one corporation
  - CI class: datasets whose corporations are in competition (conflict of interest or CI)
  - access rules: for reading/writing data

The Chinese Wall Model

- Not a true multilevel secure model
  - the history of a subject's access determines access control
  - Bell-LaPadula model and Clark-Wilson model are not history dependent
- Subjects are only allowed access to info that is not held to conflict with any other info they already possess
- Once a subject accesses info from one dataset, a **wall** is set up to protect info in other datasets in the same CI
- Useful when applied to employees of a consulting company when service competing customers

The Chinese Wall Model

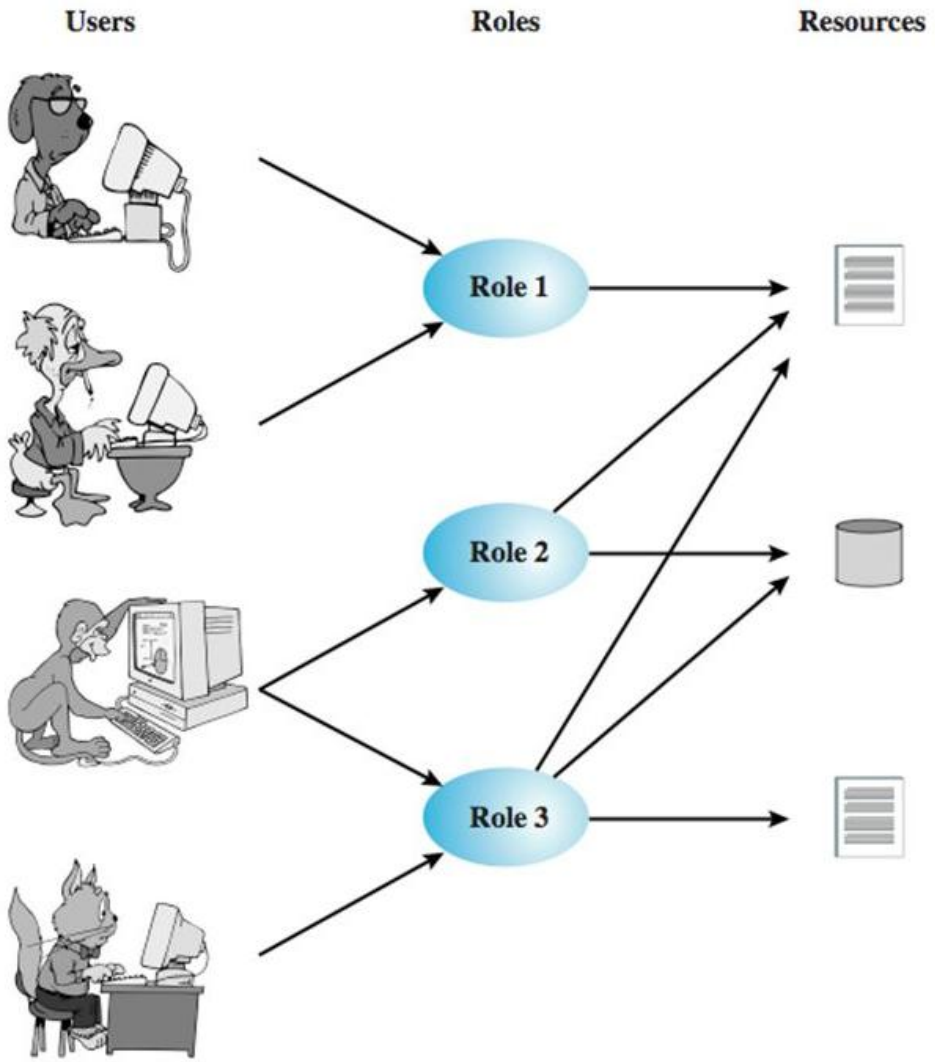


- Two CI classes in the above: Bank A and Bank B
- An access to one class will automatically drop the right to access to other class

Role-Based Access Control

- Access assigned to 'roles', not directly to users (or user identity)
- Users assigned to roles (if a role is required for his/her job)
- Many-to-many relationship between users and roles
- Roles can be static or dynamic
- Standardized by NIST (National Institute of Standards and Technology)

Role-Based Access Control



General RBAC and Variations

- minimal model $RBAC_0$, consisting of
  - Individual users
  - Roles (e.g., student, teacher, admistrator)
  - Permissions: access privileges to objects/resources
  - Session: a mapping from users to roles
- $RBAC_1$: $RBAC_0$ + role hierarchies
- $RBAC_2$: $RBAC_0$ + constraints (e.g, mutual exclusive roles, cardinality)
- $RBAC_3$: $RBAC_0$ + role hierarchies + constraints (this is required to express complex constraints, e.g., prerequisite)

Summary

- Access Control principles: subjects, objects, access rights
- Access Control Matrix, Access Control List (ACL), Capabilities
- Discretionary Access Control (DAC) and its use in the UNIX file systems
- Mandatory Access Control (MAC)
- Bell LaPadula Model — no read-up, no write-down
- Clark-Wilson Model, Chinese Wall Model
- Role Based Access Control (RBAC):
  user, role, role hierarchies, constraints