Computer & Information Security (372-1-460-1)

Access Control

Dept. of Software and Information
Systems Engineering, Ben-Gurion University

Prof. Yuval Elovici, Dr. Asaf Shabtai
{elovici, shabtaia}@bgu.ac.il

Spring, 2019

Access Control

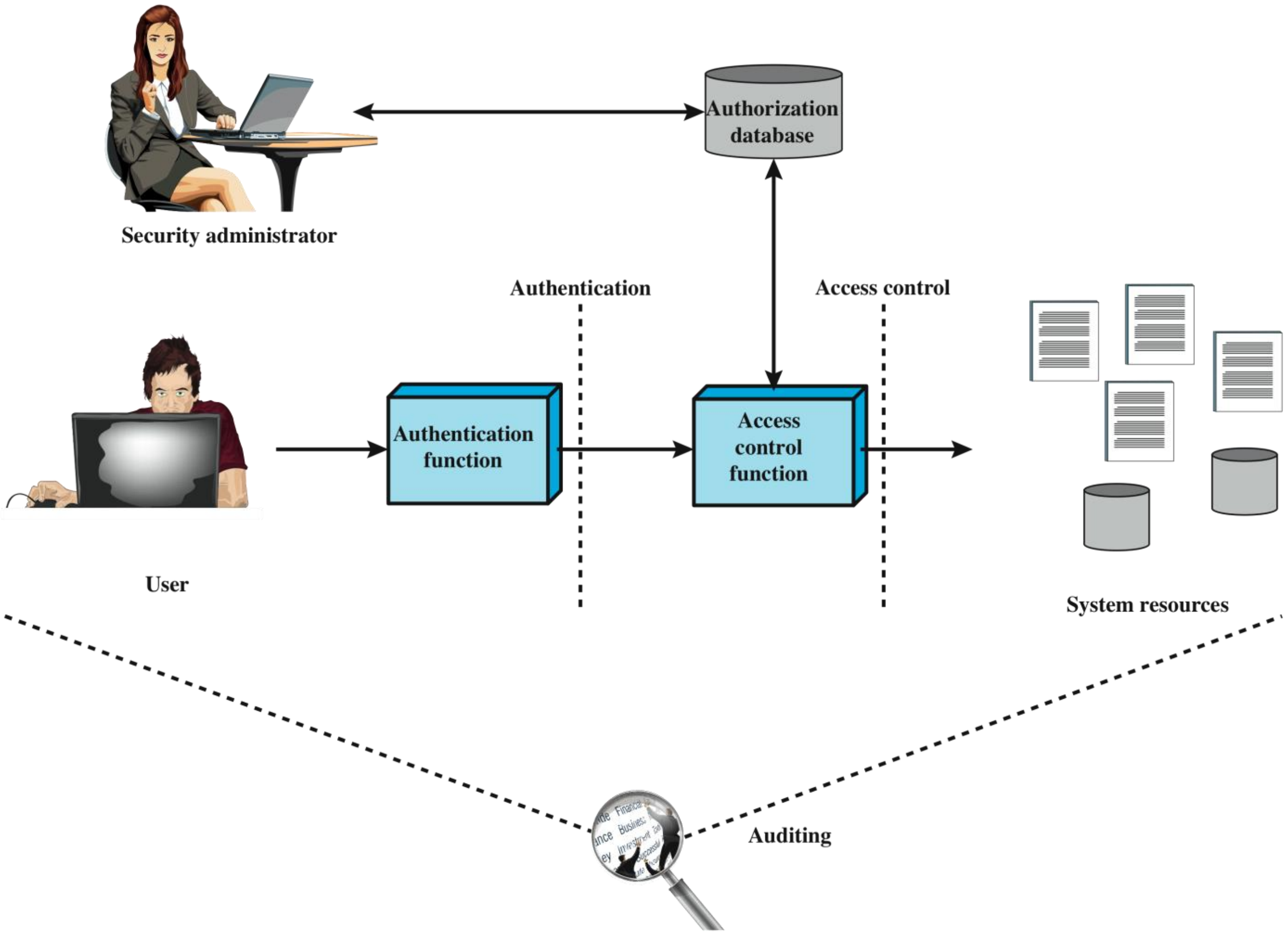
- ITU-T (International Telecommunication Union)
- Recommendation X.800 defines access control as follows:

"The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner."

Access Control Principles

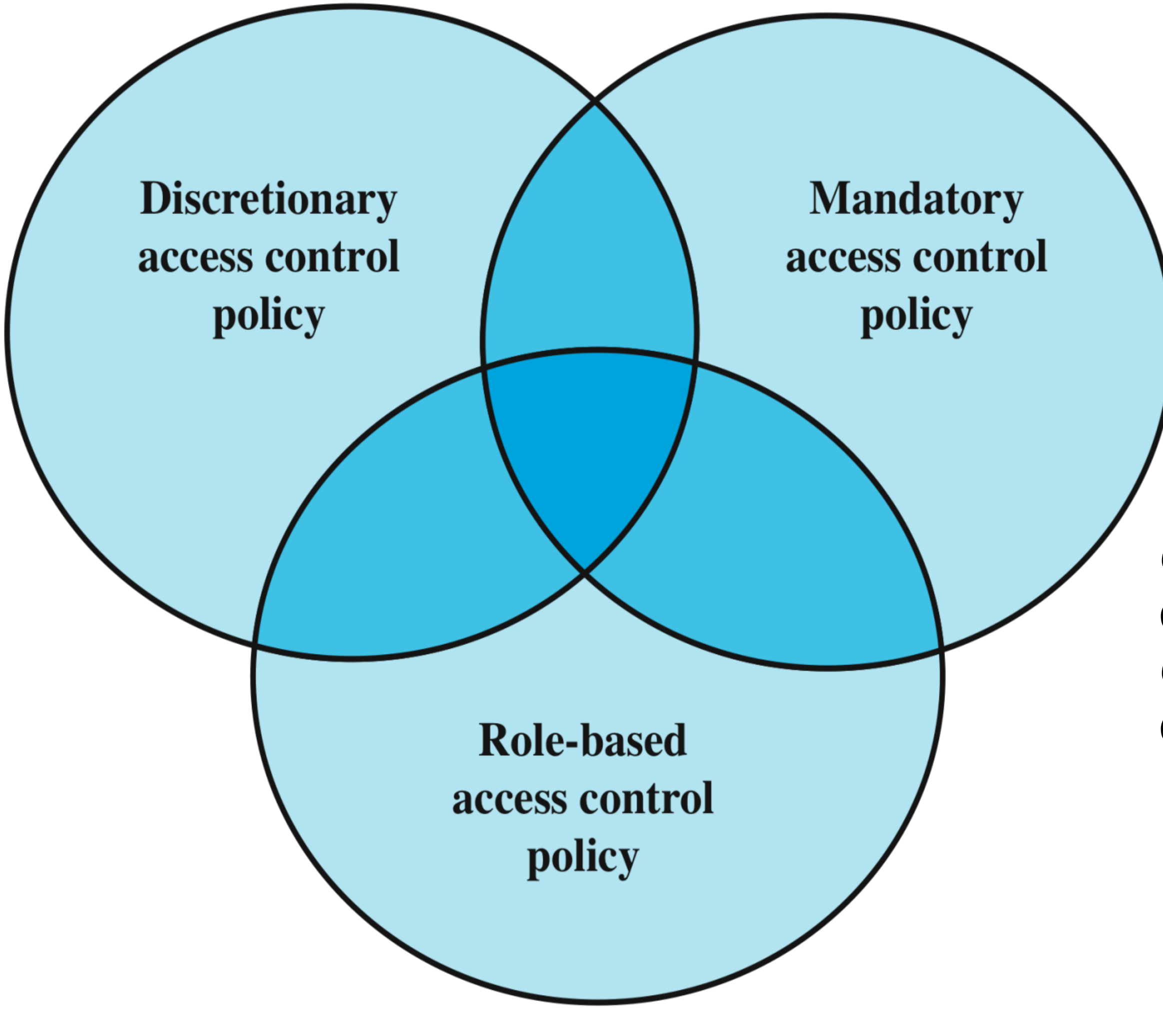
- **Authentication:**
  - Verification that the claimed identity of a user or other system entity are valid.
- **Authorization:**
  - The granting of a right or permission to a system entity to access a system resource.
  - This function determines who is trusted for a given purpose.
- **Audit:**
  -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

Access Control Principles



Multiple Access Control Policies

DAC
Entity can enable another entity to access some resources



MAC
Entity cannot enable another entity to access some resources

RBAC -(Roles and rules defined in the system controls the access

Access Control Requirements

- Reliable input (authenticated user, IP address…)
- Support for proper specifications
- Least privilege
- Separation of duty
- Open and closed policies
- Policy combinations and conflict resolution
- Administrative policies
- Dual control
- Authorization creep
- Security domain – all resources and users that are working under the same security policy
- Single-Sign-On (SSO)

Access Control Basic Elements

- subject – entity capable of accessing objects
  - User or application gains access by process
  - typically held accountable for the actions they initiate
  - often have three classes:
    - owner (e.g files, system administrator, project leader),
    - Group (user may belong to authorization group who share same privileges)
    - World (the least amount of authorizations not included in Owner,Group)

- object – resource to which access is controlled
  - entity used to contain and/or receive information
  - protection depends on the environment in which access control operates

- access right – describes the way in which a subject may access an object
  - e.g. read, write, execute, delete, create, search

Discretionary Access Control (DAC)

- Scheme in which an entity may enable another entity to access some resource

- Often provided using an access matrix
  - one dimension consists of identified subjects that may attempt data access to the resources
  - the other dimension lists the objects that may be accessed

- Each entry in the matrix indicates the access rights of a particular subject for a particular object

Access Matrix

OBJECTS

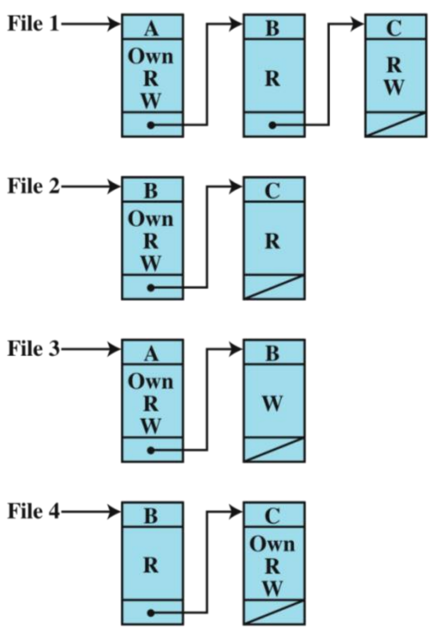
SUBJECTS		File 1	File 2	File 3	File 4
	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

(a) Access matrix

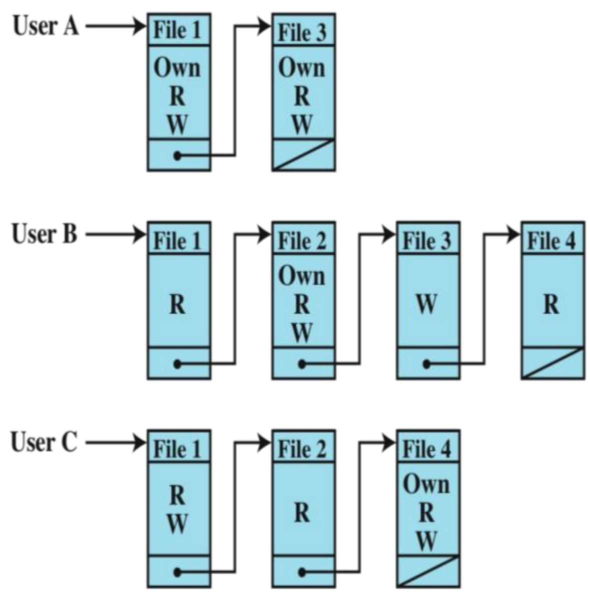
Usually Sparse Matrix, why ?

Example of Access Control Structures

- (b) ACLs = Access Control Lists
  - Per columns = per object
- (c) Capability ticket
  - Per row= per subject (user\system)
  - Can also contain default\public entry



(b) Access control lists for files of part (a)



(c) Capability lists for files of part (a)

Example of Access Control Structures - Authorization Table

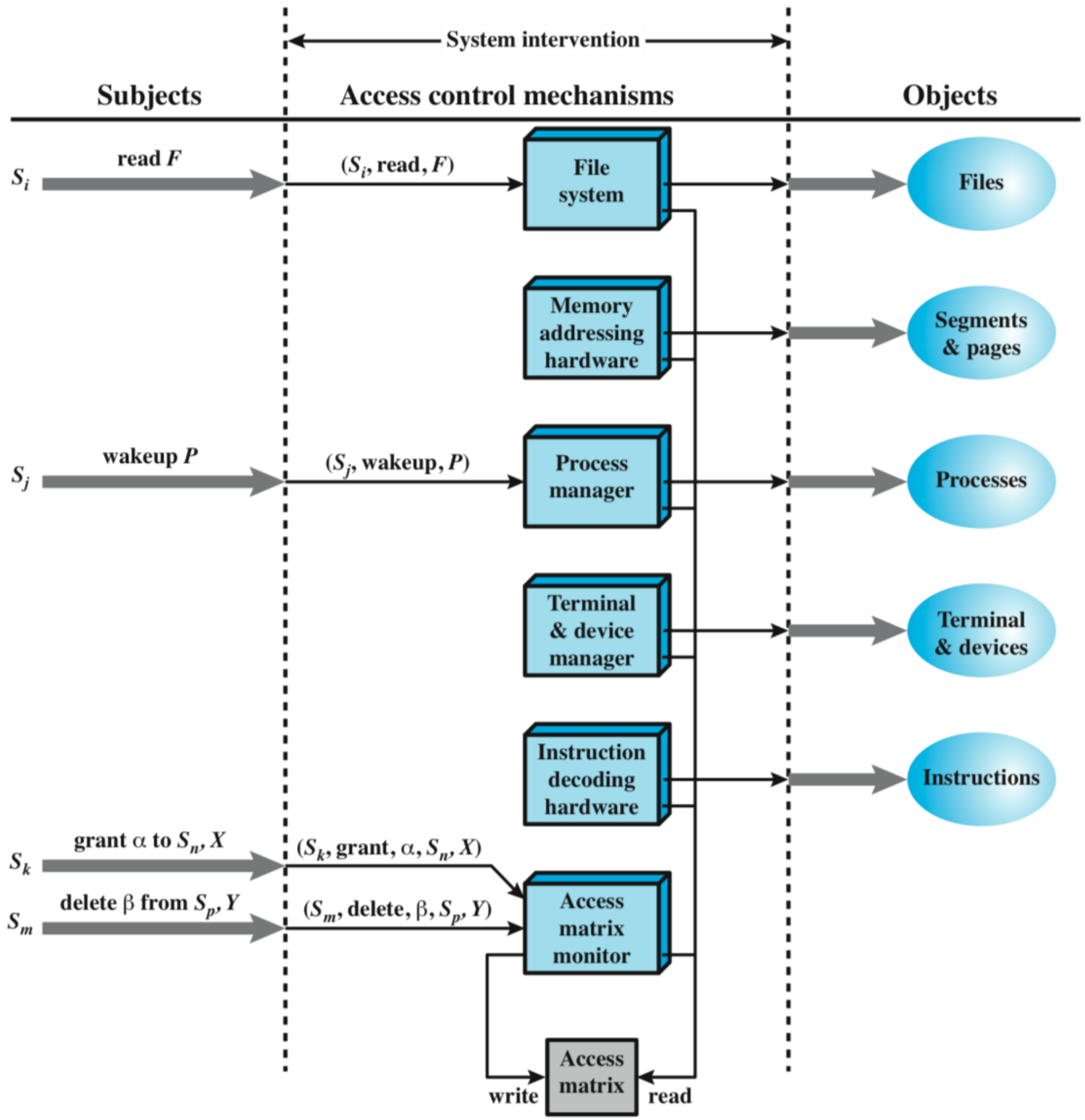
Subject	Access Mode	Object
A	Own	File 1
A	Read	File 1
A	Write	File 1
A	Own	File 3
A	Read	File 3
A	Write	File 3
B	Read	File 1
B	Own	File 2
B	Read	File 2
B	Write	File 2
B	Write	File 3
B	Read	File 4
C	Read	File 1
C	Write	File 1
C	Read	File 2
C	Own	File 4
C	Read	File 4
C	Write	File 4

Extended Access Control Matrix

		OBJECTS								
		subjects			files		processes		disk drives	
		S_1	S_2	S_3	F_1	F_2	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 Function

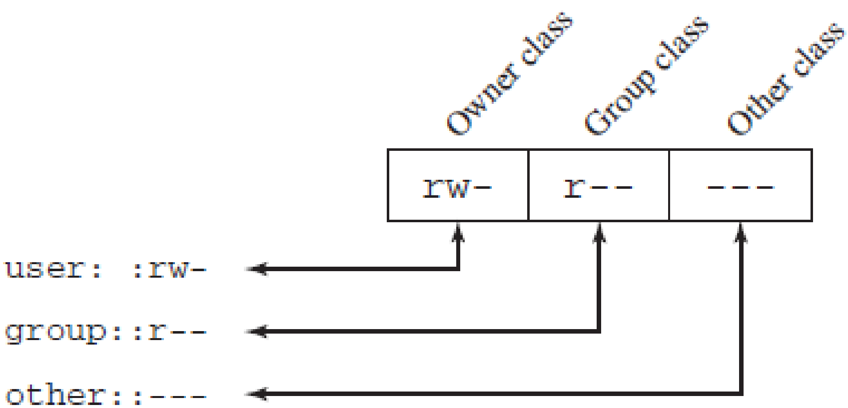


Access Control System Commands

Rule	Command (by S_o)	Authorization	Operation
R1	**transfer** $\begin{Bmatrix} \alpha * \\ \alpha \end{Bmatrix}$ **to** S, X	'α*' in $A[S_o, X]$	store $\begin{Bmatrix} \alpha * \\ \alpha \end{Bmatrix}$ in $A[S, X]$
R2	**grant** $\begin{Bmatrix} \alpha * \\ \alpha \end{Bmatrix}$ **to** S, X	'owner' in $A[S_o, X]$	store $\begin{Bmatrix} \alpha * \\ \alpha \end{Bmatrix}$ in $A[S, X]$
R3	**delete** α **from** S, X	'control' in $A[S_o, S]$ or 'owner' in $A[S_o, X]$	delete α from $A[S, X]$
R4	$w \leftarrow$ **read** S, X	'control' in $A[S_o, S]$ or 'owner' in $A[S_o, X]$	copy $A[S, X]$ into w
R5	**create object** X	None	add column for X to A; store 'owner' in $A[S_o, X]$
R6	**destroy object** X	'owner' in $A[S_o, X]$	delete column for X from A
R7	**create subject** S	none	add row for S to A; execute **create object** S; store 'control' in $A[S, S]$
R8	**destroy subject** S	'owner' in $A[S_o, S]$	delete row for S from A; execute **destroy object** S

Example: UNIX File Access Control

- Inode (Index node) = control structure needed for OS for particular file. Contains:
  - Each user has unique user identification number (user ID)
  - Each user is a member of a primary group identified by a group ID
  - 12 protection bits
    - specify read, write, and execute permission for the owner of the file, members of the group and all other users
  - First 9 bits include authorization to USER,GROUP and other.
  - the owner ID, group ID, and protection bits are part of the file's inode

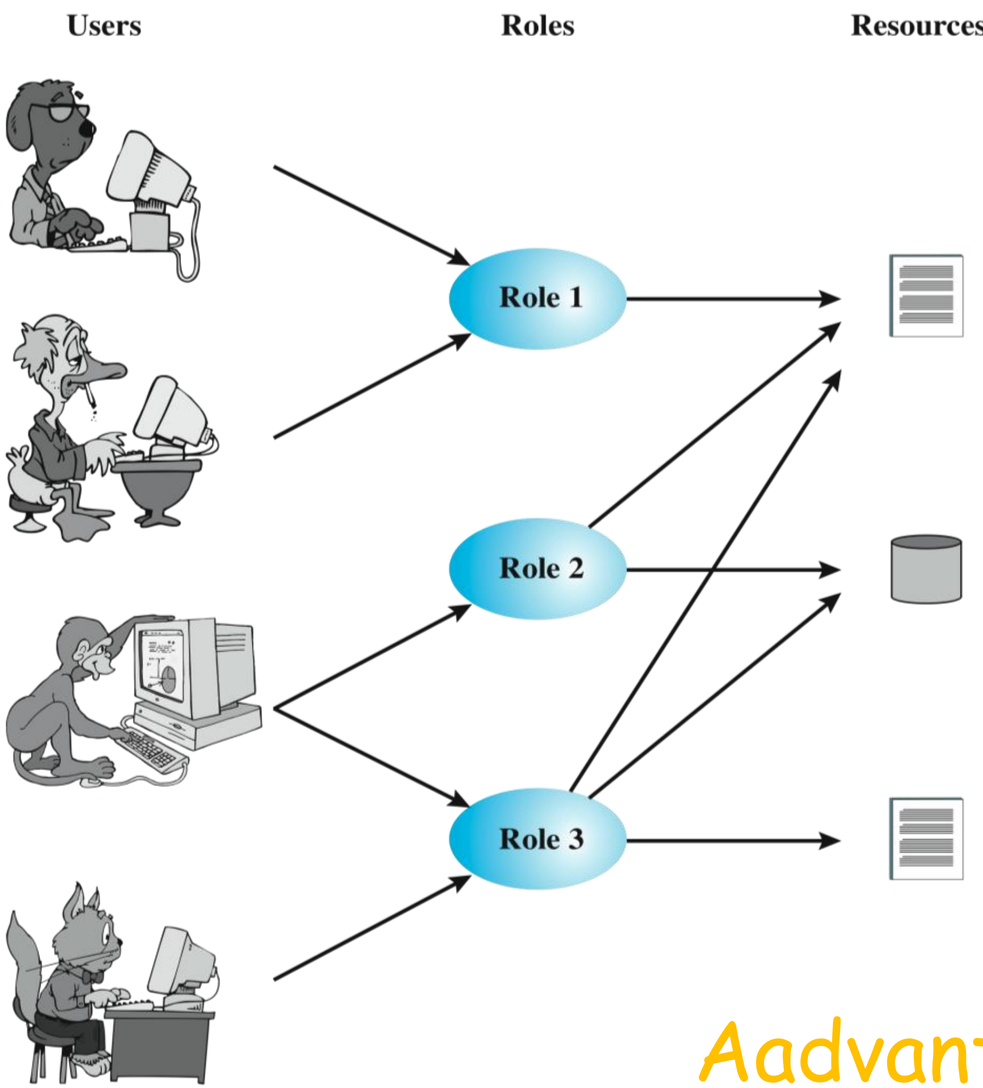


(a) Traditional UNIX approach (minimal access control list)

Traditional UNIX File Access Control

- Last 3 bits, specify additional behaviors:
- "set user ID"(SetUID)
- "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
  - when applied to a directory it specifies that only the owner of any file in the directory can rename, move, or delete that file
- superuser
  - is exempt from usual access control restrictions
  - has system-wide access

Role-Based Access Control (RBAC)



Aadvantage over DAC ?

RBAC - Access Control Matrix

	R_1	R_2	• • •	R_n
U_1	✖			
U_2	✖			
U_3		✖		✖
U_4				✖
U_5				✖
U_6				✖
⋮				
U_m	✖			

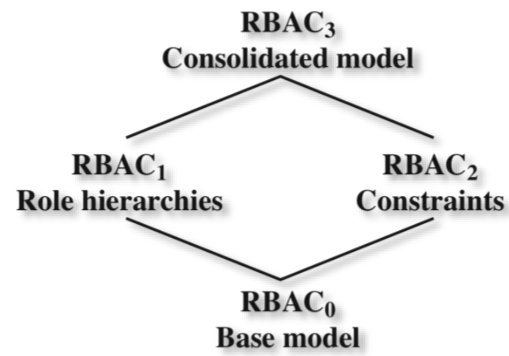
- Assigning Users to Roles:
  - More Users than Roles
  - User Can be assigned to multiple roles
  - Roles can be assigned to multiple users

- Assigning Roles to Objects:
  - More Objects than Roles
  - Role can be treated as an object – for role hierarchies definition.

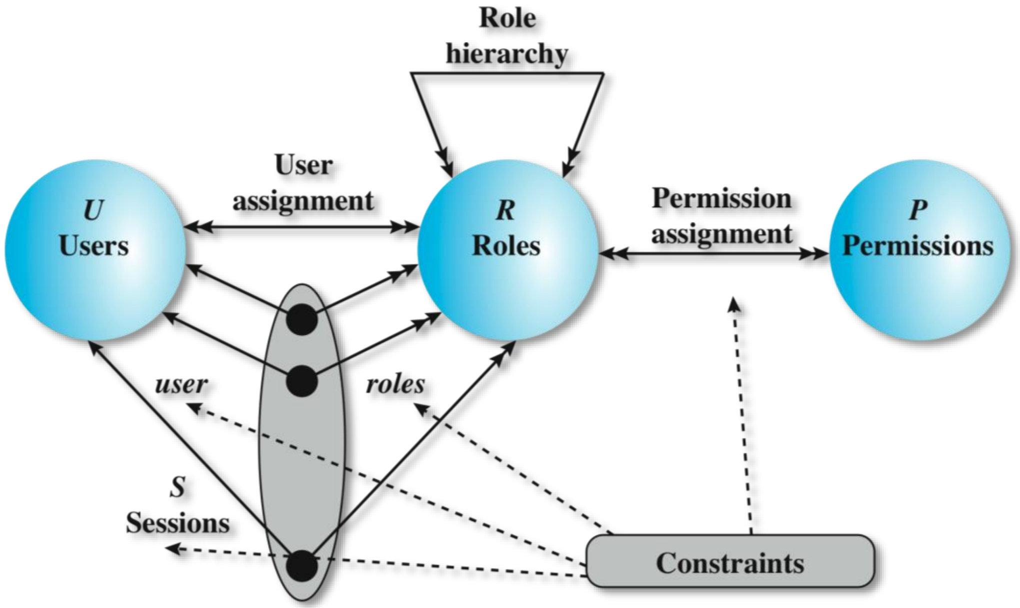
OBJECTS

ROLES	R_1	R_2	R_n	F_1	F_1	P_1	P_2	D_1	D_2
R_1	control	owner	owner control	read *	read owner	wakeup	wakeup	seek	owner
R_2		control		write *	execute			owner	seek *
⋮									
R_n			control		write	stop			

Role-Based Access Control Models



(a) Relationship among RBAC models



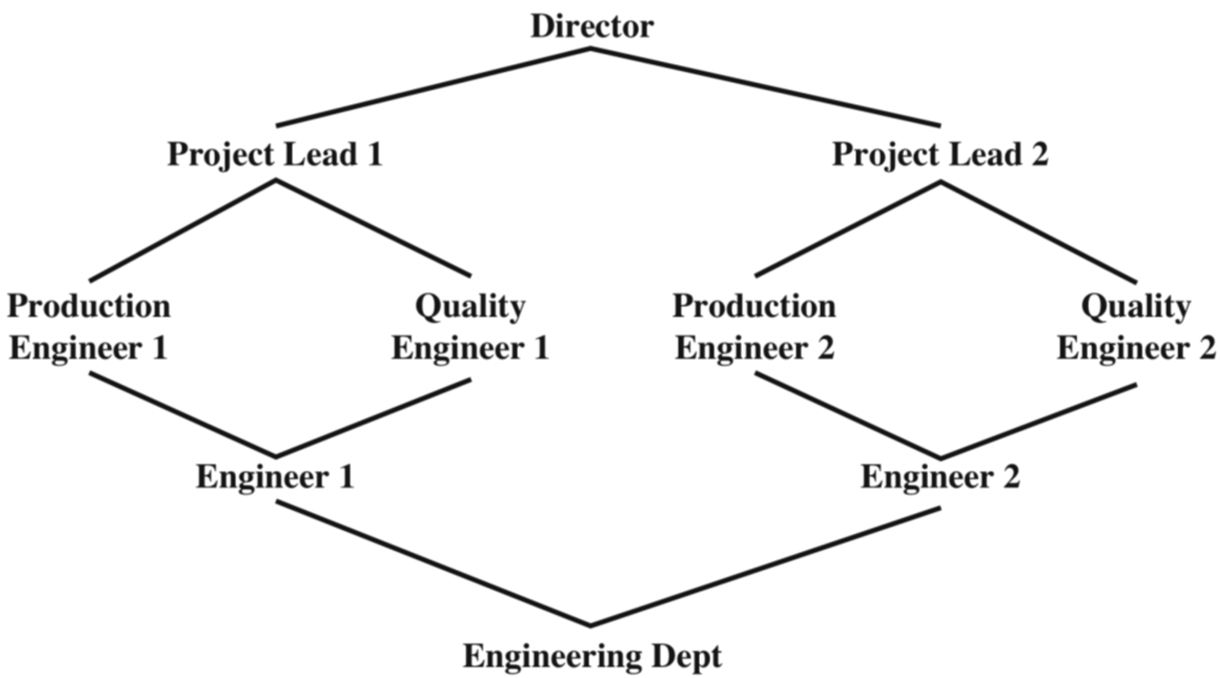
(b) RBAC models

Models	Hierarchies	Constraints
RBAC$_0$	No	No
RBAC$_1$	Yes	No
RBAC$_2$	No	Yes
RBAC$_3$	Yes	Yes

Minimum functionality of RBAC system

Figure 4.9 A Family of Role-Based Access Control Models. RBAC$_0$ is the minimum requirement for an RBAC system. RBAC1 adds role hierarchies and RBAC$_2$ adds constraints. RBAC3 includes RBAC$_1$ and RBAC$_2$. [SAND96]

Example of Role Hierarchy - RBAC1



Constraints - RBAC2

- Provide a means of adapting RBAC to the specifics of administrative and security policies of an organization
- A defined relationship among roles or a condition related to roles
- Mutually exclusive roles
  - a user can only be assigned to one role in the set (either during a session or statically)
  - any permission (access right) can be granted to only one role in the set
- Cardinality
  - setting a maximum number with respect to roles
- Prerequisite roles
  - dictates that a user can only be assigned to a particular role if it is already assigned to some other specified role

Other Access Control Models

- Context-bases access control (location)
- Rule-based access control (calendar)
- Content-based access control (confidentiality)

Information classification Why is it important ?

- Examples
  - unclassified, confidential, secret, top secret (used by US government)
  - public, internal use, confidential (used in the private sector)

- Everyone should know how sensitive or critical is the information in order to handle it appropriately

- To be able to provide proper security level and control measures (e.g., avoid unauthorized disclosure)

- To be able to comply with the organization management policy, regulations

- Follow best practices:
  - least privilege principle,
  - compartmentalization,
  - need-to-know,
  - separation of duties

Multilateral Security

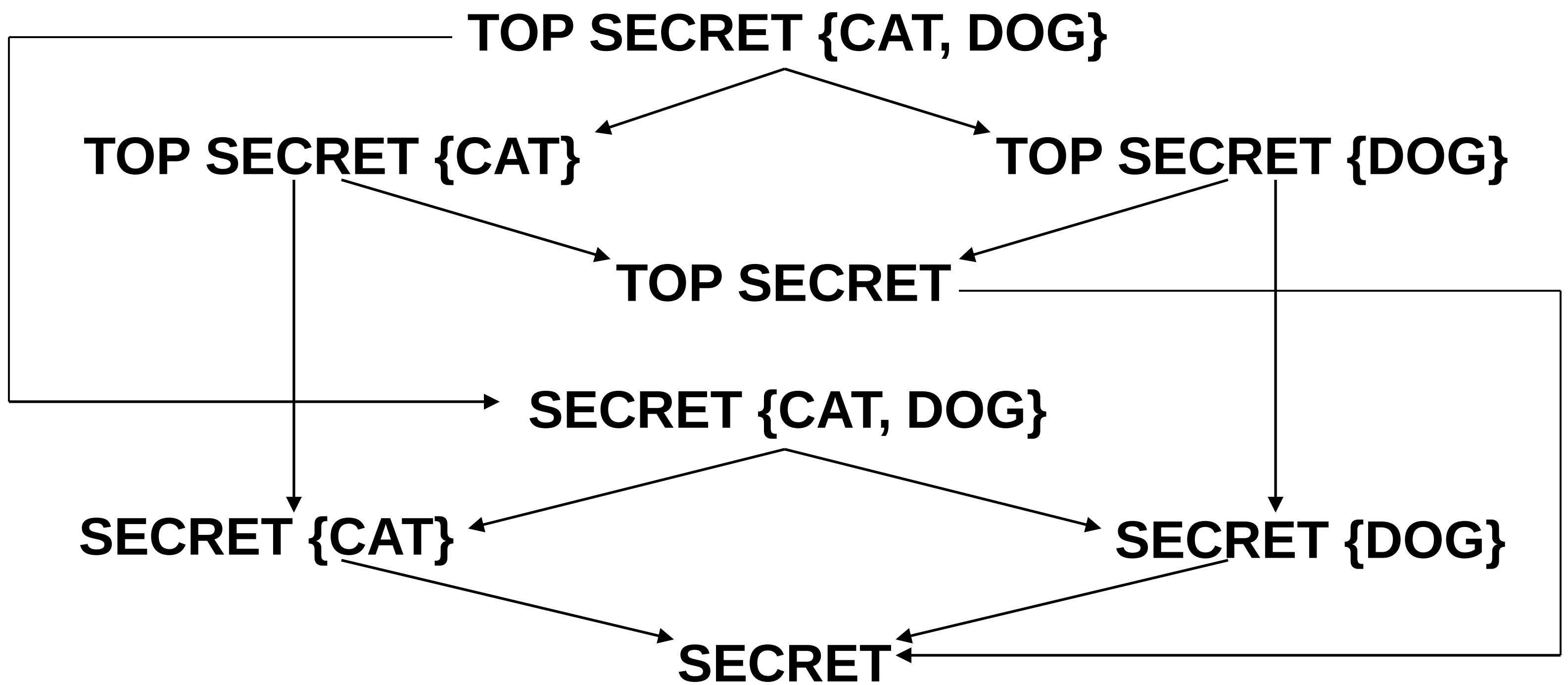
- Multilevel Security (MLS) enforces access control **up and down**
- Simple hierarchy of security labels may not be flexible enough
- Multilateral security enforces access control **across** by creating compartments
- Suppose **TOP SECRET** divided into **TOP SECRET {CAT}** and **TOP SECRET {DOG}**
- Both are **TOP SECRET** but information flow restricted across the **TOP SECRET** level

Multilateral Security

- Why compartments?
  - Why not create a new classification level?
- May not want either of
  - **TOP SECRET {CAT} $\geq$ TOP SECRET {DOG}**
  - **TOP SECRET {DOG} $\geq$ TOP SECRET {CAT}**
- Compartments allow us to enforce the **need to know** principle
  - Regardless of your clearance, you only have access to info that you need to know

Multilateral Security

- Arrows indicate "≥" relationship



- Not all classifications are comparable, e.g.,

TOP SECRET {CAT} vs SECRET {CAT, DOG}

Information classification Process

- Define/develop functional policies and procedures
  - Identify the appropriate number of classification levels and their definitions
  - Define the classification process
  - Define data/information owners
  - Declassification
  - How information in each class should be handled/protected (e.g., labeling magnetic media and hard copy, encryption)

Roles and responsibilities

- Data\system owners
  - Classify data and systems
  - Set user access to data and systems
  - Decide on business continuity priorities

- Custodians
  - Handle data and systems that do not belong to them
  - Ensure the security of data and systems according to the owners' specification

Bell-LaPadula (BLP) Confidentiality Model (1973)

- By David Bell and Leonard LaPadula
- Hierarchical,
- Addresses the confidentiality of information in a system
- Mandatory access control
- Formalize the U.S. Department of Defense (DoD) multilevel security (MLS) policy; the basis for the Orange Book
- Simple security property – read capability only
  - A subject can read information at or below his level of secrecy, but cannot read any information above his level of secrecy (no read up)
- Star property – write capability only
  - A subject can write information at or above his level of secrecy, but cannot write any information above his level of secrecy (no write down)
- Strong Star property – read and write capability
  - A subject can read and write information only at his level of secrecy

Higher secrecy level

Assigned level

Lower secrecy level

Covert Channel

Covert Channel

- MLS designed to restrict legitimate channels of communication
- May be other ways for information to flow
- For example, resources shared at different levels may signal information
- **Covert channel**: "communication path not intended as such by system's designers"

Covert Channel Example

- Alice has **TOP SECRET** clearance, Bob has **CONFIDENTIAL** clearance
- Suppose the file space shared by all users
- Alice creates file FileXYzW to signal "1" to Bob, and removes file to signal "0"
- Once each minute Bob lists the files
  - If file FileXYzW does not exist, Alice sent 0
  - If file FileXYzW exists, Alice sent 1
- Alice can leak **TOP SECRET** info to Bob!

Covert Channel Example

Alice:	Create file	Delete file	Create file		Delete file
Bob:	Check file	Check file	Check file	Check file	Check file
Data:	1	0	1	1	0
Time:					

Covert Channel

- Other examples of covert channels
  - Print queue
  - ACK messages
  - Network traffic, etc., etc., etc.
- When does a covert channel exist?
  1. Sender and receiver have a shared resource
  2. Sender able to vary property of resource that receiver can observe
  3. Communication between sender and receiver can be synchronized

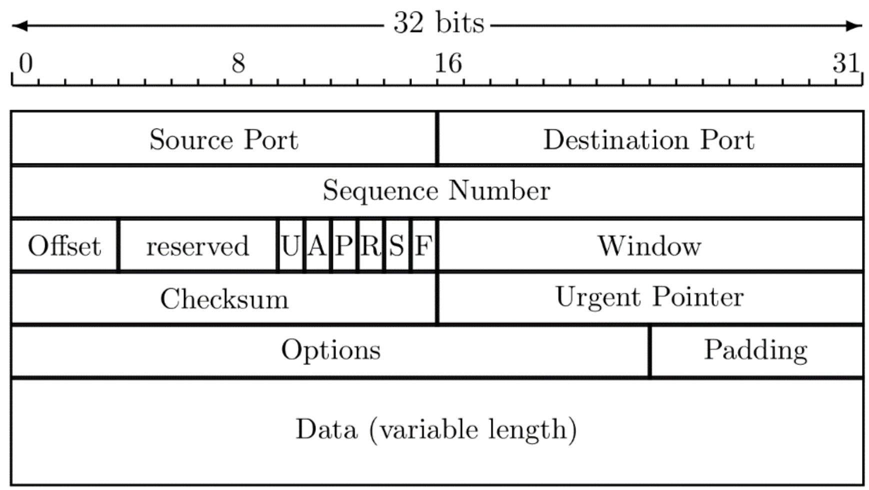
Covert Channel

- Covert channels exist almost everywhere
- Easy to eliminate covert channels…
  - Provided you eliminate all shared resources and all communication
- Virtually impossible to eliminate all covert channels in any useful system
  - DoD guidelines: goal is to **reduce covert channel capacity** to no more than 1 bit/second
  - Implication is that DoD has given up trying to eliminate covert channels!

Covert Channel

- Consider 100MB **TOP SECRET** file
  - Plaintext version stored in **TOP SECRET** place
  - Encrypted with AES using 256-bit key, ciphertext stored in **UNCLASSIFIED** location
- Suppose we reduce covert channel capacity to 1 bit per second
- It would take more than 25 years to leak entire document thru a covert channel
- But it would take less than 5 minutes to leak 256-bit AES key thru covert channel!

Real-World Covert Channel



- Hide data in TCP header “reserved” field
- Or use covert_TCP, tool to hide data in
  - Sequence number
  - ACK number

Real-World Covert Channel

- Hide data in TCP sequence numbers
- Tool: covert_TCP
- Sequence number X contains covert info