Access control

What is access control

- Security functionality ensuring that authorized users can do what they are permitted to do
  - Hardware
  - Operating systems
  - Web servers
  - Databases
- Preserve confidentiality, integrity, availability

- OS provides access to resources
  - CPU, memory, files, devices, network
- Needs access control to
  - Protect OS from applications
  - Protect applications from each other

Subjects and Objects

- Subject: active entity needing to access resources
- Object: passive entity accessed by subjects
  - Files, devices
- Subjects access objects: they perform actions on objects

- Access control
  - Define what operations subjects can perform on objects
  - Permissions

Example

Subjects :

- Alice and Bob: doctors
- Tom and Paul: nurses
- Sarah: secretary

Objects :

- patients
- documents: medical or administrative

Actions :

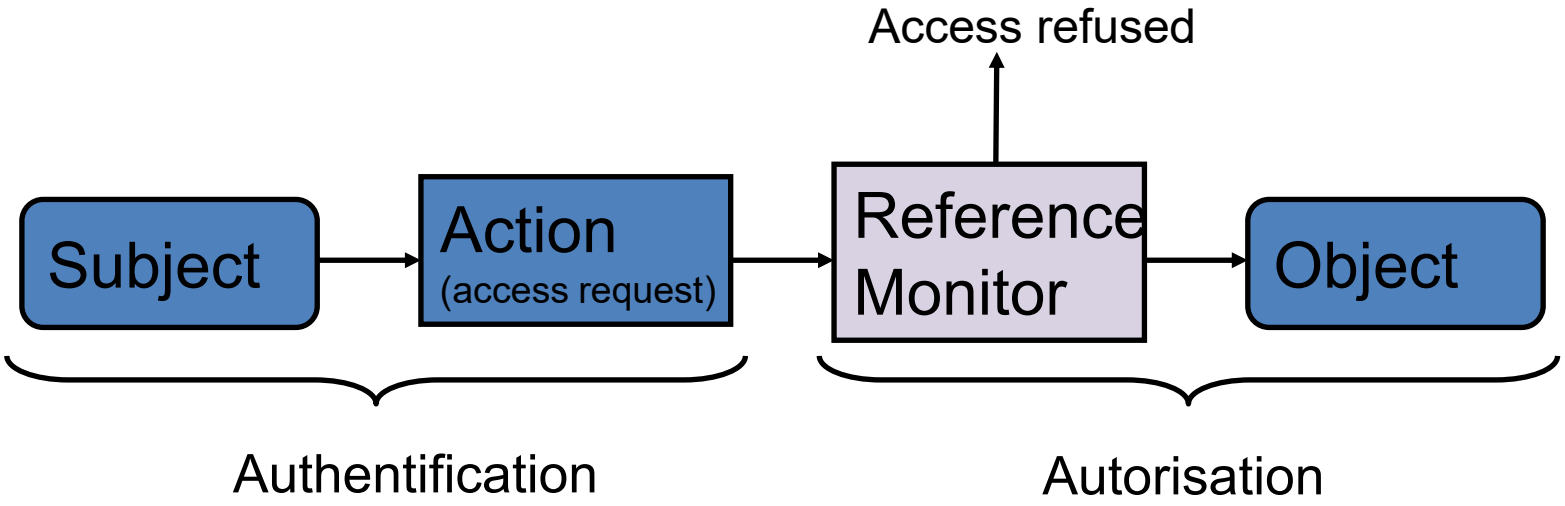
- consult, modify the medical doc of a patient
- create a doc

We can specify :

- what is permitted for an object
- or what a subject can do

Access Control

- Reference Monitor
  - Decision process filtering the access demands
  - Guard controlling whether a subject can access an object



Principle of Least Privilege

- Each subject can access only the resources necessary to perform its task
  - The nurses cannot modify docs of patients
  - They can read them, or add information

Access control classification

- Discretionary Access Control (DAC)
- Mandatory Access Control (MAC)

- Role-Based Access Control (RBAC)
- Task-Based Access Control (TBAC)
- ...

Discretionary Access Control (DAC)

- Owner determines access rights
  - Typically identity-based access control
  - Owner specifies other users who have access

- A subject can pass information onto anyother subject
- In some cases, access rights may be transferred
- Users are in charge of access permissions
  - Most systems use this

- Unix file system

Discretionary Access Control (DAC)

- Suppose we can find the owner of each object
- Violates the principle of least privilege

- Access permission may be transferred without informing owner
  - E.g., A grants read permission to B on f, B copies f to f', B is owner of f', and transmits read permission to C
    - A is not informed

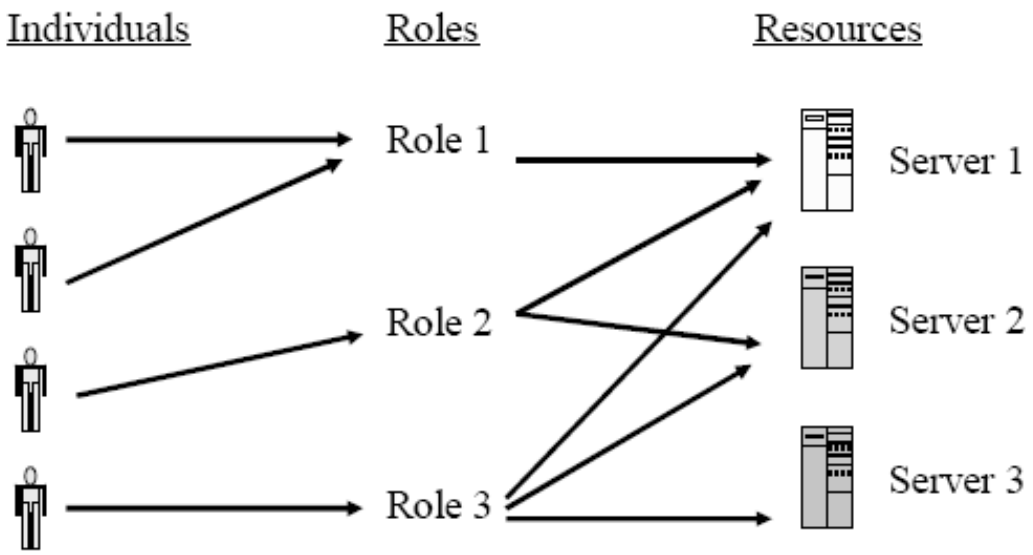
- DAC can limit access for good users
- Not adapted in sensible systems

Mandatory Access Control (MAC)

- Motivation
  - Control information flow even in sensible environment
- Non discretionnary
  - Permissions fixed for each subject
- Rules specify granting of access
  - Also called rule-based access control
- Administrators are in charge of access permissions

Role-Based Access Control (RBAC)

- Access decisions depend on roles
  - Administrators define roles for various job functions
  - Each role contains permissions to perform certain operations
  - Users are assigned one or more roles
- Allow enforcement of both MAC & DAC



Users change frequently, Roles don't

Access Control Matrix

- Primary abstraction for protection in computer security

- Rows: domains (subjects or groups of subjects)
- Columns: objects
- Each entry in the matrix represents an access right of a domain on an object

	bill.doc	edit.exe	fun.com
Alice	$\emptyset$	{execute}	{execute, read}
Bob	{read, write}	{execute}	{execute, read, write}

Access Control Matrix: Examples

- Modelling a programming language
  - S: procedures/modules
  - O: procedures/variables
  - A: execution of functions
- Modelling a local area network
  - S: stations
  - O: stations
  - A: protocols

	Compt	Inc_ctr	Dcr_ctr
Inc_ctr	+		
Dcr_ctr	-		
Manager		call	call

hosts	Station1	Station2	Station3
Station1	own	ftp	ftp, mail
Station2		own, ftp, nfs	ftp, nfs
Station3	mail	ftp	own, ftp, nfs

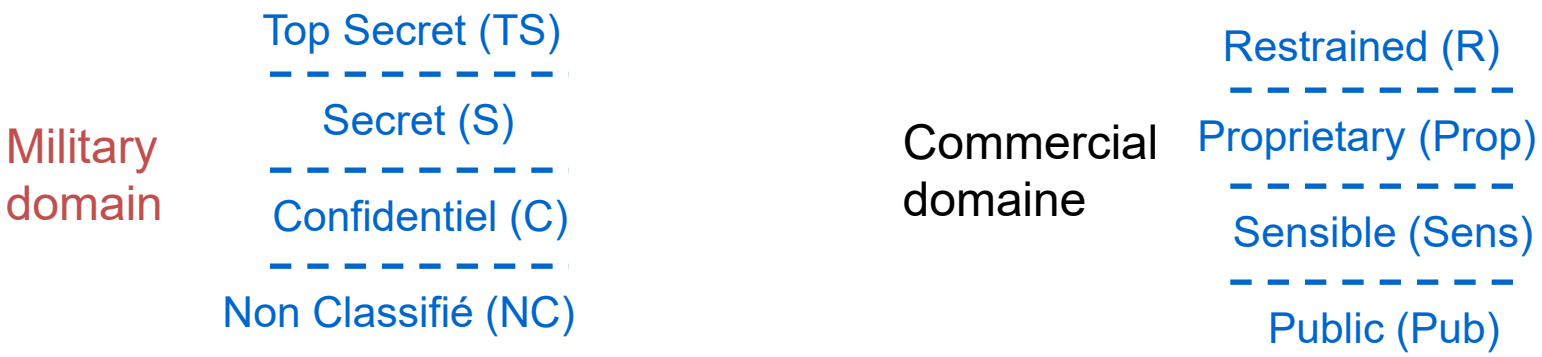
Bell LaPadula (BLP) Model

- Designed for the military: US airforce
- Based on U.S. military classification levels
- Ensures confidentiality

- Multi-level mandatory access control

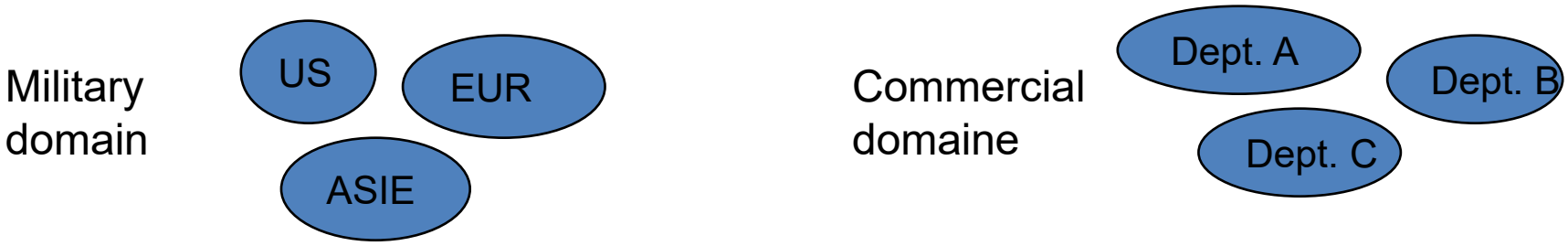
Multilevel access control

- Entities are assigned security levels
- Totally ordered
  - TS>S>C>NC
  - R>Prop>Sens>Pub



Multilevel access control

- Entities are assigned categories
- Principle of need to know
- Partially ordered

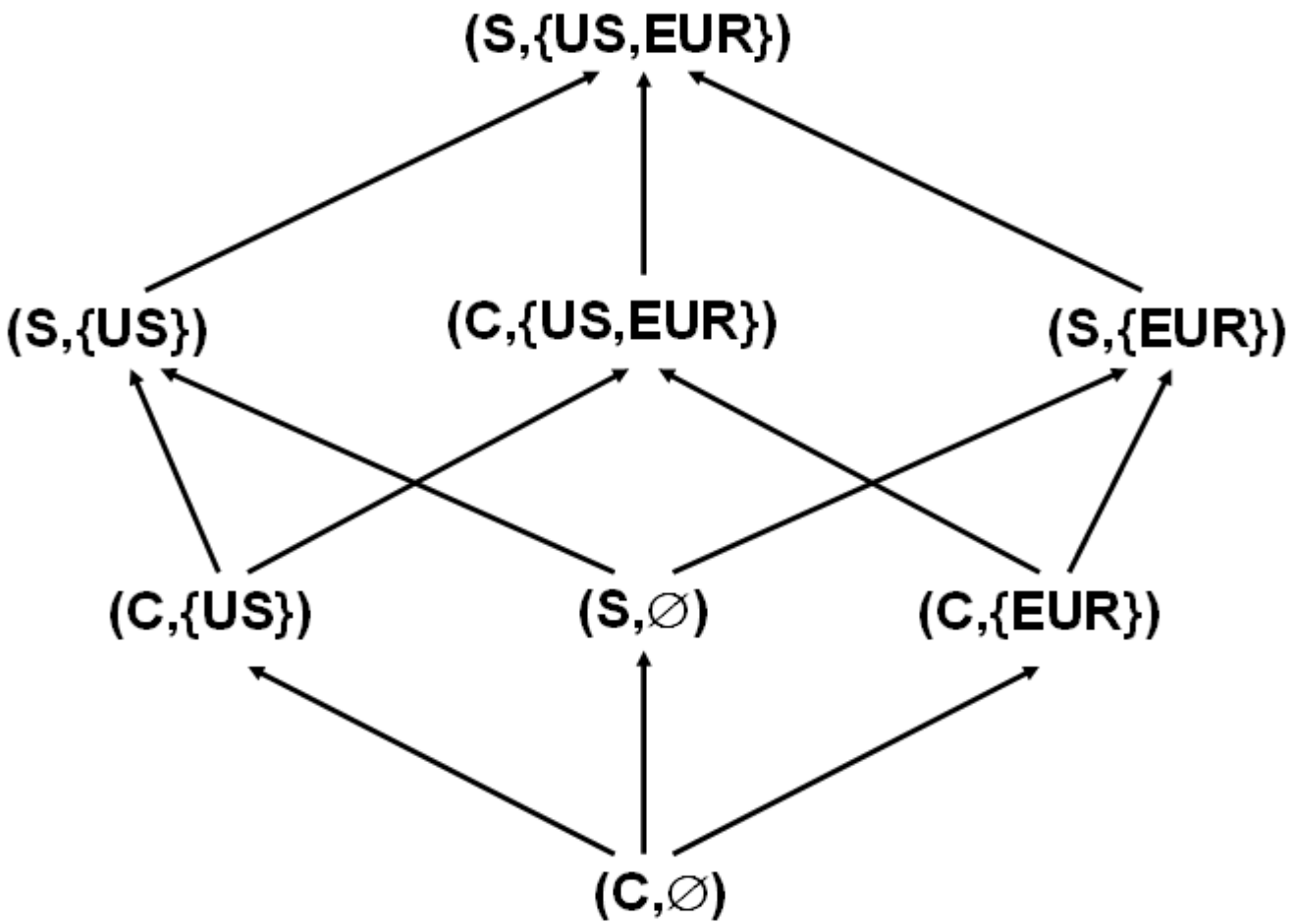


Security labels

- Label of security: LS=Lev×P(Cat)
  - (TS, {US, EU})
- Partially ordered relationship: dom (dominate)
  - (n1, C1) dom (n2, C2) iff n1 ≥ n2 and C2 ⊆ C1
  - Example : (TS, {US,EUR}) dom (S, {US})
- (LS, dom) is a lattice
  - admits an upper and lower-bound

Security labels

- Lattice induced by security labels: Lev={C,S}, Cat={US,EUR}

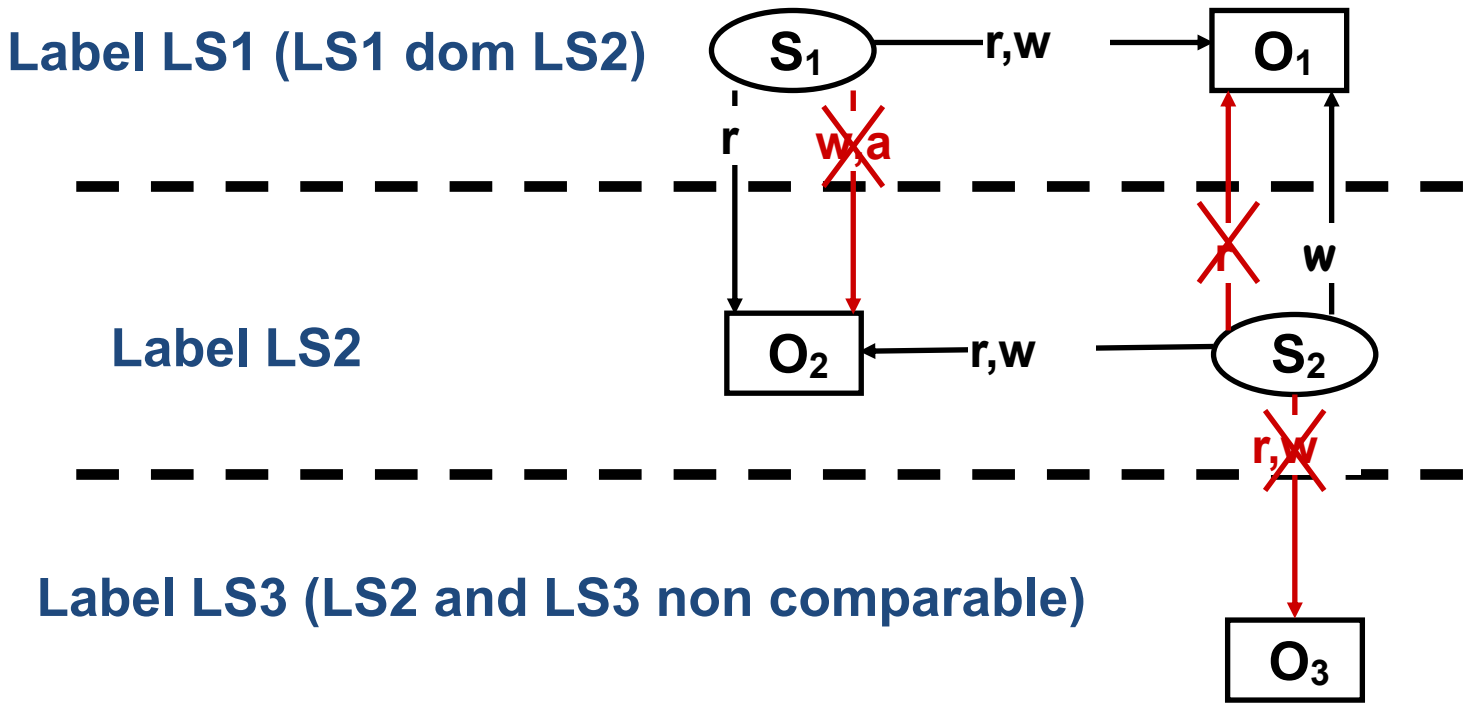


Bell LaPadula (BLP) Model

- LS(S): label of security of subject
- LS(O): label of security of object

- Simple Security Condition: NO READ UP
  - S can read O iff LS(S) dom LS(O)
- *-Property (star-property, ou confinement): NO WRITE DOWN
  - S can write O iff LS(O) dom LS(S)

Bell LaPadula (BLP) Model

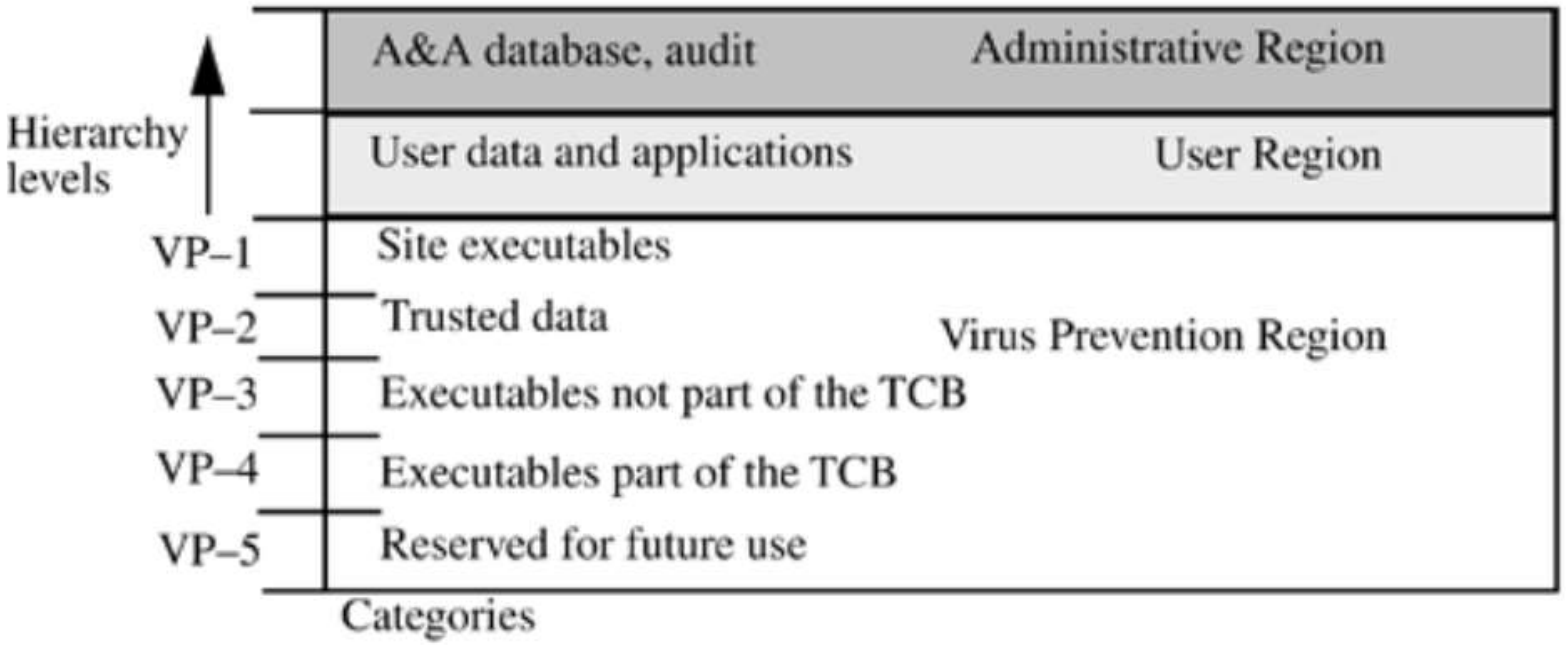


Test

- Consider the system of a hospital
  - S: doctors, nurses
  - O: prescriptions, documents
  - Doctors can read and write prescriptions and docs
  - Nurses can read and write prescriptions, but only add in docs

- Prove we cannot model the system using BLP model
- Consider an extended BLP model
  - Each S has two labels (L1,L2) with L2 dom L1
  - Consider S with labels (L1,L2) and O with label L
    - S can read O iff L2 dom L
    - S can write O iff L dom L1
  - Prove that the extended BLP model can model the system
- Prove that we can create an information flow between two objects with non-comparable labels

Unix Data General B2 system



- S: users & processes; O: files, directories
  - Apply BLP model
- Why UR is between AR and VPR?
- Write-up is not allowed, why?
- S can create files in directory D iff LS(S)=LS(D), why?
  - Problems with /tmpand /var/mail: multilevel directory