System Security

Access Control Models

Giovanni Russello

g.russello@auckland.ac.nz

http://www.cs.auckland.ac.nz/compsci725s2c/

Access Control Models

- Discretionary AC (DAC)
- Mandatory AC (MAC)
- Role-based AC (RBAC)

Discretionary Access Control

- Subjects are able to assign on the objects they control access rights to other subjects
- Model used in operating systems and DB management systems
- often provided using an access matrix

Access Control Matrix

	File1	File2	File3	File4
User A	Own Read Write		Own Read Write	
User B	Read	Own Read Write	Write	Read
User C	Read Write	Read		Own Read Write

Access Control List

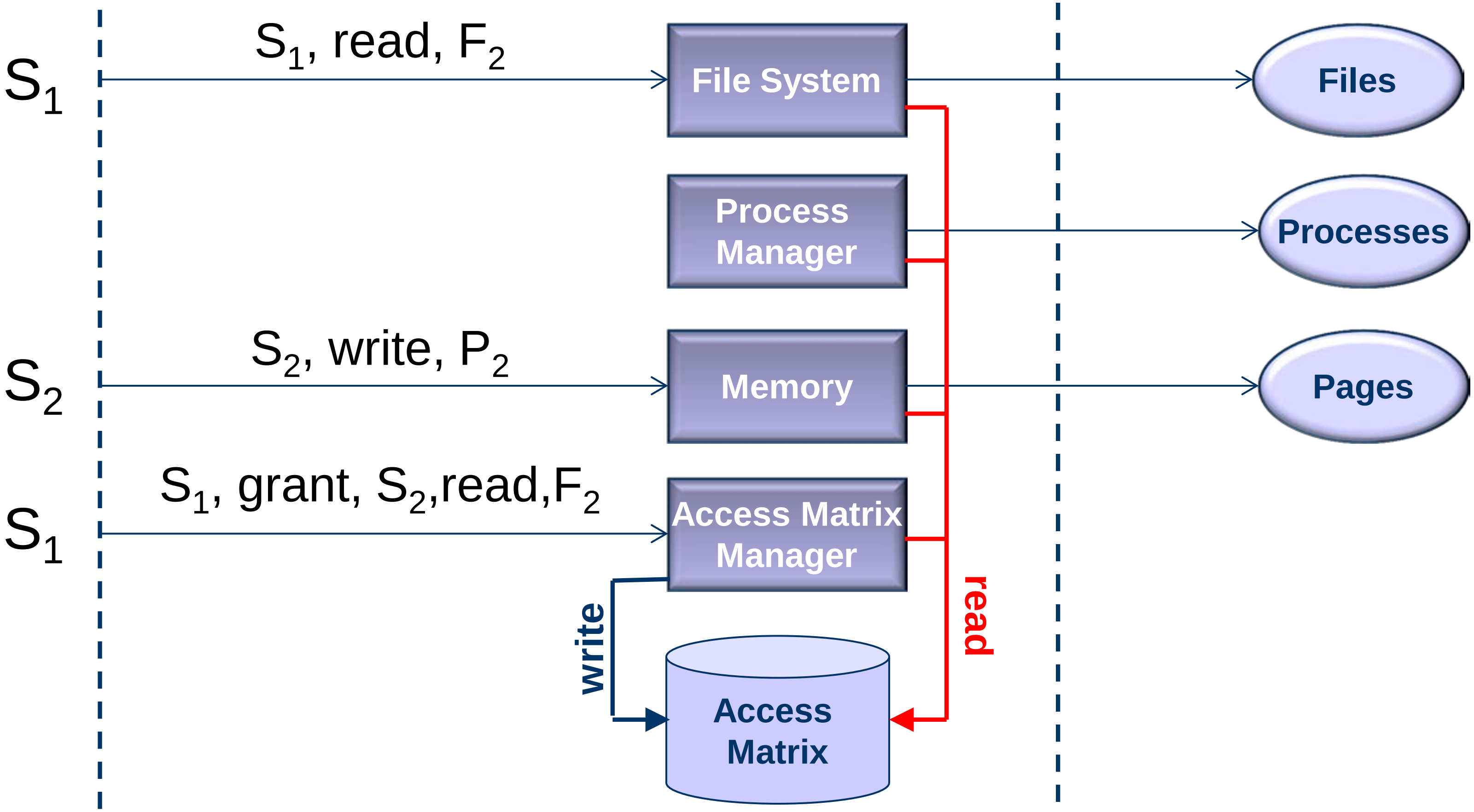
	File1	File2	File3	File4
User A	Own Read Write		Own Read Write	
User B	Read	Own Read Write	Write	Read
User C	Read Write	Read		Own Read Write

Capability List

	File1	File2	File3	File4
User A	Own Read Write		Own Read Write	
User B	Read	Own Read Write	Write	Read
User C	Read Write	Read		Own Read Write

Capability Myths Demolished: http://srl.cs.jhu.edu/pubs/SRL2003-02.pdf

Access Matrix Details



Mandatory AC

Entities cannot enable other entities to access their resources

It enforces a lattice between labels assigned to subjects and object

- security labels: how sensitive or critical a system resource is
- security clearances: which entities are eligible to access certain resources

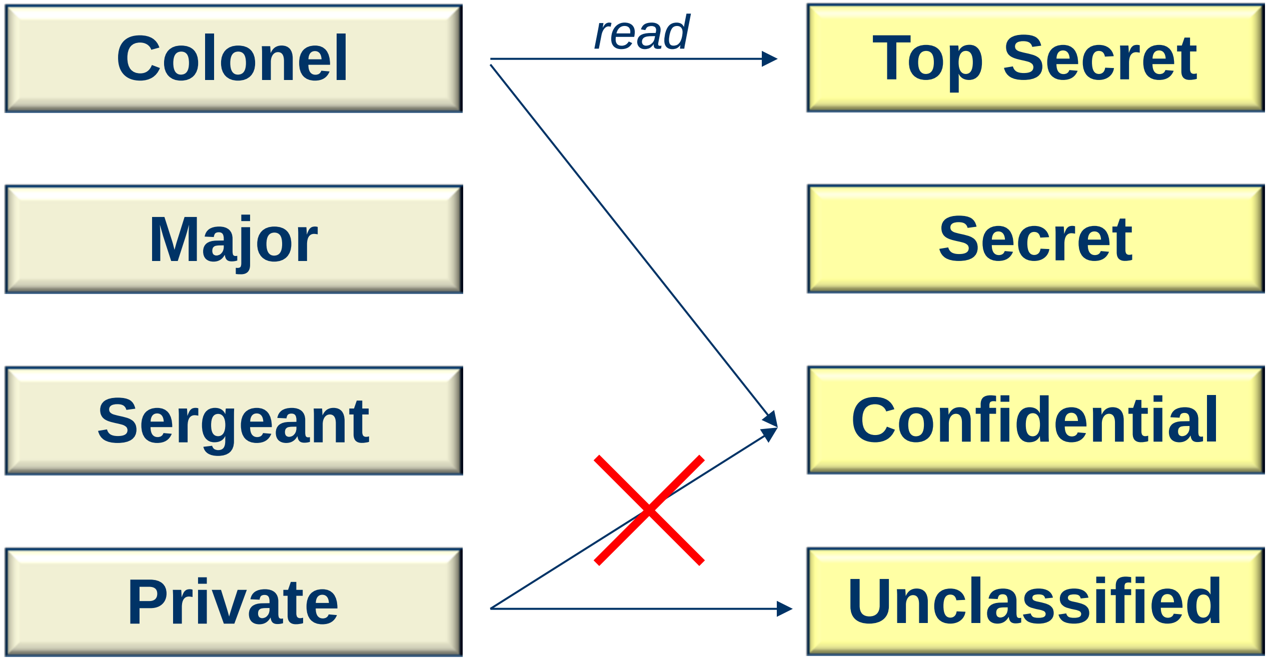
MAC: The Bell-LaPadula Model (‘76)

The main goal is to control the **confidentiality of information**

User Labels	Data Labels
Colonel	**Top Secret**
Major	**Secret**
Sergeant	**Confidential**
Private	**Unclassified**

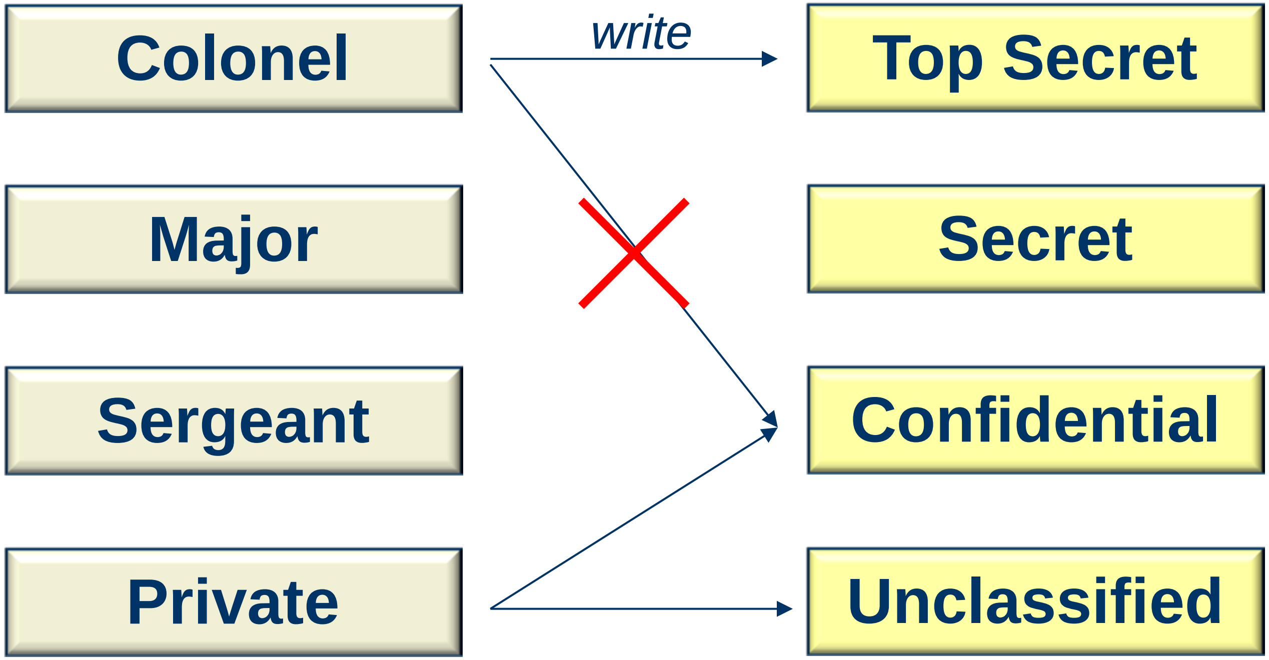
MAC Confidentiality Rules

Simple Security Property: No Read-Up



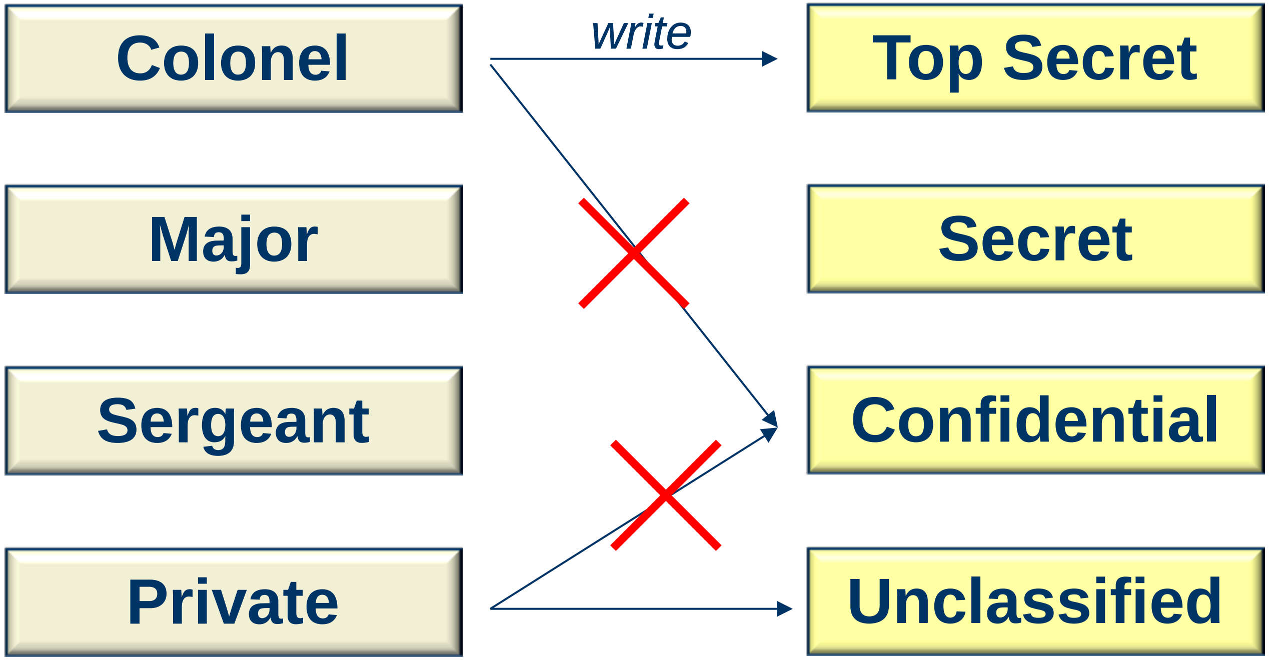
MAC Confidentiality Rules

**(Star)property: No Write-Down*



MAC Confidentiality Rules

Strong *(Star)-property: No Write-Down & No Write-up



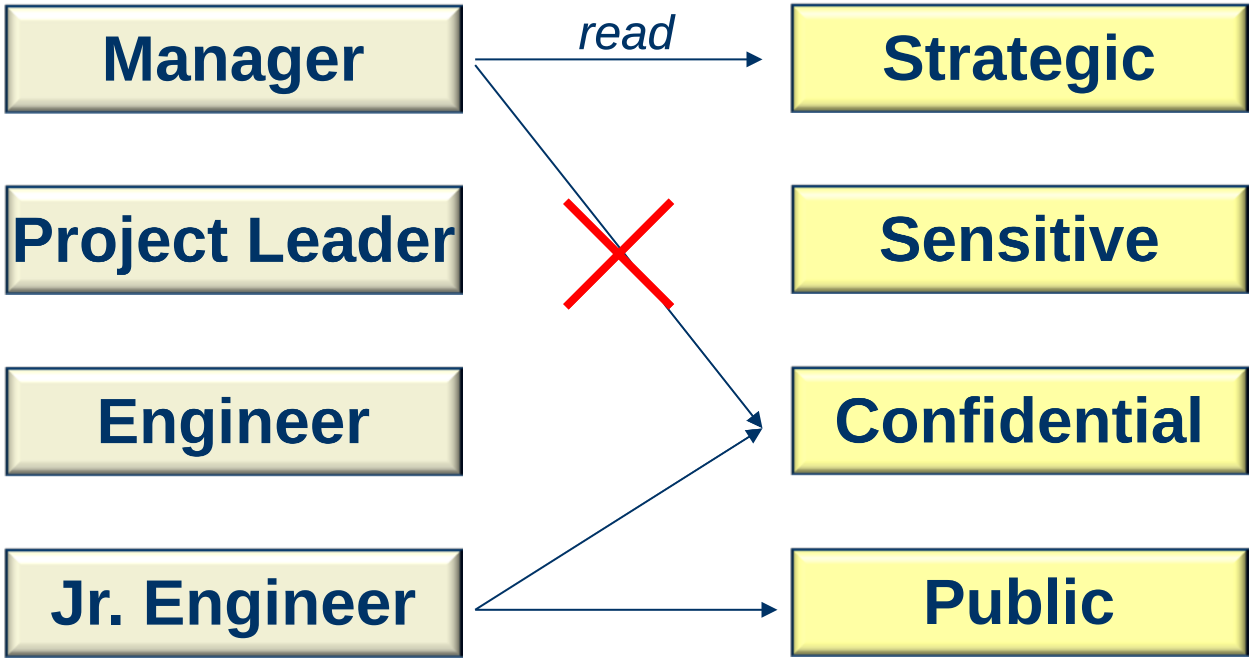
MAC: Biba Integrity Model (‘77)

The main goal is to control the **integrity of information**

User Labels	Data Labels
Manager	**Strategic**
Project Leader	**Sensitive**
Engineer	**Confidential**
Jr. Engineer	**Public**

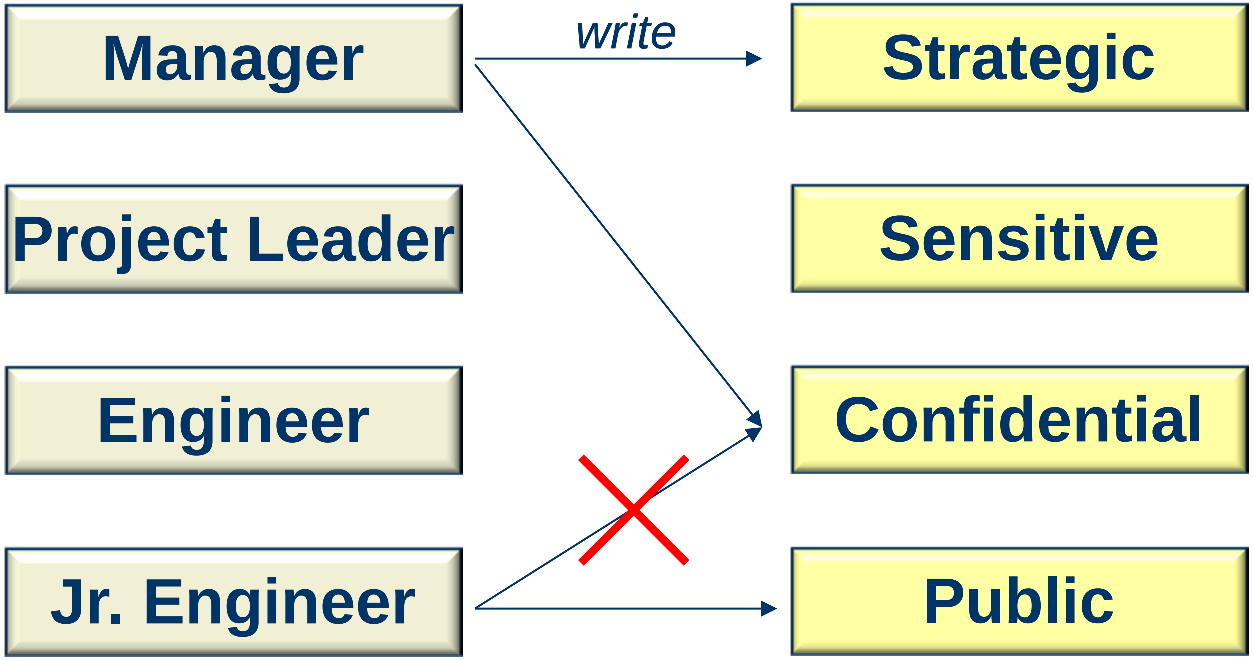
MAC Integrity Rules

Simple Integrity Axiom: No Read Down



MAC Integrity Rules

*(Star)-Integrity Axiom: No Write Up



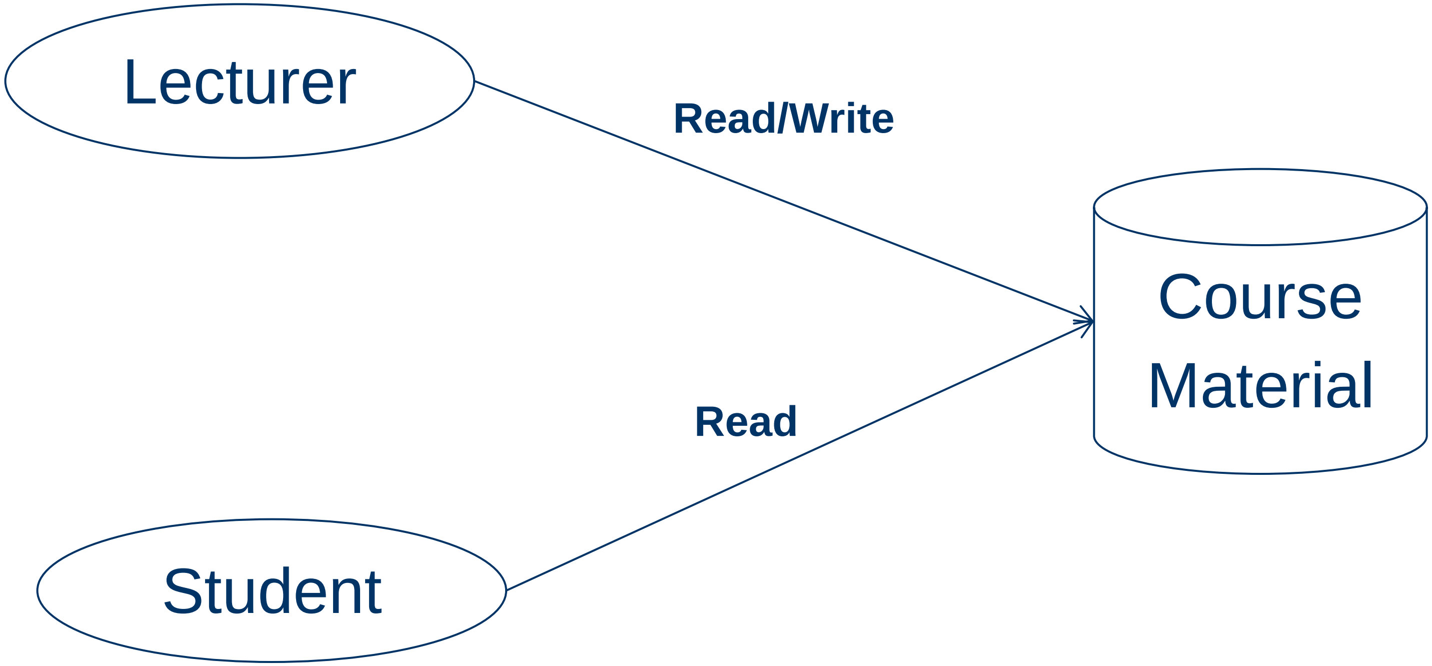
Where is MAC used

- BLP: Implemented the multi-level security policy for US Department of Defense
- BIBA: Implemented in the FreeBSD MAC policy
- A combined versions of BLP and BIBA is used in Android!

Role Based Access Control

- Enterprises organise employees in different roles
- RBAC maps roles to access rights
- Access rights are assigned to roles
- After Subjects are authenticated they are assigned to roles

A simple example



User to Role

	Lecturer	S Lecturer	Ass Prof	Prof
Giovanni	X			
Ulrich		X		
Clark				X

Role to Access Rights

	File1	File2	File3	File4
Lecturer	Own Read Write		Own Read Write	
S Lecturer	Read	Own Read Write	Write	Read
Professor	Read Write	Read		Own Read Write

Extensions to the Model

- A user can be in more than one role
  - Robert Amor is both Prof. and HoD
- Roles can be organised in Hierarchies
  - Prof>Ass Prof>Sen Lect>Lect
  - Top Roles inhered access rights of Lower Roles
- Constraints to enforce enterprise-specific requirements

RBAC Constraints

- Separation of Duties (SoD)
  - Protecting the organisation from frauds
- Chinese Wall CW)
  - Conflict of interests between different domains

SoD Details

SoD are used when an activity involves more than one roles:

Purchase order needs to be **prepared** by a **clerk** and then **authorised** by a **manager**

To avoid a fraud, the user that prepares the order should not be the same that authorises it

Static SoD

- In Static SoD, the same subject cannot be member of two mutually exclusive roles
  - **clerk** and **manager** are mutually exclusive
- Too restrictive: the user might get assigned to both roles as long as she is not working on the same order!

Dynamic SoD

- In Dynamic SoD, the same subject can be member of two mutually exclusive roles
- **However,** it requires extra checks that need to be done at runtime to avoid undesired behaviour
- Simple DSoD, Object DSoD, Operational DSoD, History DSoD

Simple DSoD

- Users cannot be active in mutually exclusive roles at the same time
- For instance, a user can be assigned to both clerk and manager roles as long as she is not active on both at the same time

Object DSoD

- Users can be active in mutually exclusive roles at the same time as long as she is not operating on the same object instance for the entire business process
- For instance, a user can act in either clerk or manager role for a purchase order
- Let say that there is another operation: sending the order to depot. The user cannot execute this action even if it is not in conflict

Operational DSoD

- Users can be active in mutually exclusive roles at the same time but cannot perform all the operations of business process
- For instance, a user can activate both clerk and manager roles but cannot execute both the prepare and authorise operations (even for different objects!!)

History DSoD

- Users can be active in mutually exclusive roles at the same time as long as she is not authorised to execute all the operation for the same object instance
- For instance, a user can be activate in both clerk role for a purchase order and manager role for another order

Chinese Wall

- It applies to accesses in multiple domains with conflict of interest
- For instance, a consultant company offering services to both Microsoft and Apple. CW makes sure that an employee of the company will not get access to documents of both companies

Resources

- Chapter 8 in Mark Stamp, *Information Security: Principles and Practice*, Wiley 2011.
- Matt Bishop, *Computer Security: Art and Science*, Addison-Wesley 2003.