Access control models

Access types

Physical access

- Physical contact between a subject and the object of interest
  - Facility, room, network, computer, storage device, authentication token, etc.
- Out of scope of this course …

Informatic or electronic access

- Information-oriented contact between a subject and the object of interest
  - Contact through request-response dialogs
- Contact is mediated by
  - Computers and networks
  - Operating systems, applications, middleware, devices, etc.

Access control

The policies and mechanisms that mediate the access of a subject to an object

Interaction model

Normal requirements

- Authentication
  - With some Level of Assurance **(LoA)**
- Authorization
- Accountability

Access control: Subjects and objects

Both digital entities

Subjects can be something exhibiting activity :

- Processes, Computers, Networks

Objects can be something being a target of an action:

- Stored data, CPU time, Memory, Processes, Computers, Network

An entity can be both subject and object

- E.g: Computer, Process, Network

Least privilege principle

Each subject should have, at any given time, the exact privileges required to the assigned tasks

- Less privileges than the required create unsurpassable barriers
- More privileges than the required create vulnerabilities
  - Damage resulting from accidents or errors
  - Potential interactions among privileged programs
  - Misuse of a privileges
  - Unwanted information flows
    - "need-to-know" military restrictions

Privilege:

- Authorization to perform a given task
- Similar to access control clearance

Every program and every user of the system should operate using the least set of privileges necessary to complete the job

J. H. Saltzer, M. D. Schroeder, The protection of information in computer systems, 1975

Access control models

	O1	O2	…	Om-1	Om
S1		Access rights			
S2					
…					
Sn-1					
Sn					

Access control matrix

- Matrix with all access rights for subjects relatively to objects
- Conceptual policy
- Real world: huge and sparse matrix (its bad)

Access control models

	O1	O2	…	Om-1	Om
S1		Access rights			
S2					
…					
Sn-1					
Sn					

ACL-based mechanisms

- ACL: Access Control List (matrix column)
  - List of access rights for specific subjects
  - Access rights can be positive or negative
  - Default subjects may often be used
- Usually ACLs are stored along with objects

```
$ getfacl index.php
# file: index.php
# owner: security
# group: www-data
user::rw-
user:www-data:r--
group::---
other::---
```

Access control models

Capability-based mechanisms

- Capability: unforgeable auth. token (matrix row)
  - Contains an object reference and access rights clearance
- Access granting
  - Transmission of capabilities between subjects (if allowed)
- Usually capabilities are kept by subjects

Example:

- Movie ticket: grants access to a movie session
- A door key: grants access to a specific house

"A capability is a token, ticket, or key that gives the possessor permission to access an entity or object in a computer system",
Dennis and Van Horn, 1966

Access control models

Capability Operations

Create: Creating a movie ticket

Delegate: Giving a movie ticket

Revoke: Getting the movie ticket back

Enable: Making the movie ticket valid

Disable: Making the movie ticket temporary invalid (e.g. on a day)

Delete: Permanently invalidating the movie ticket

Access control kinds: Mandatory access control (MAC)

Access control policy statically implemented by the access control monitor

- It's the basic, unchangeable, policies

Access control rights cannot be tailored by subjects

Example: System wide settings like laws or policies

- E.g.: police forces can access the CC address data

Access control kinds: Discretionary access control (DAC)

Some subjects (an individual?) can update rights granted or denied to other subjects for a given object

- Usually this is granted to object owners and system administrators
- Administrators can allow further updates by individual users

Example: Linux file permissions

- Users can set which users or groups can access files

```
$ ls –l
-rw-r--r--   1 user wheel     0 Nov 14 20:48 security.txt
$ chmod g+rw security.txt
$ ls –l
-rw-rw-r--   1 user wheel     0 Nov 14 20:48 security.txt
```

Access control kinds: Role-Based Access Control (RBAC)

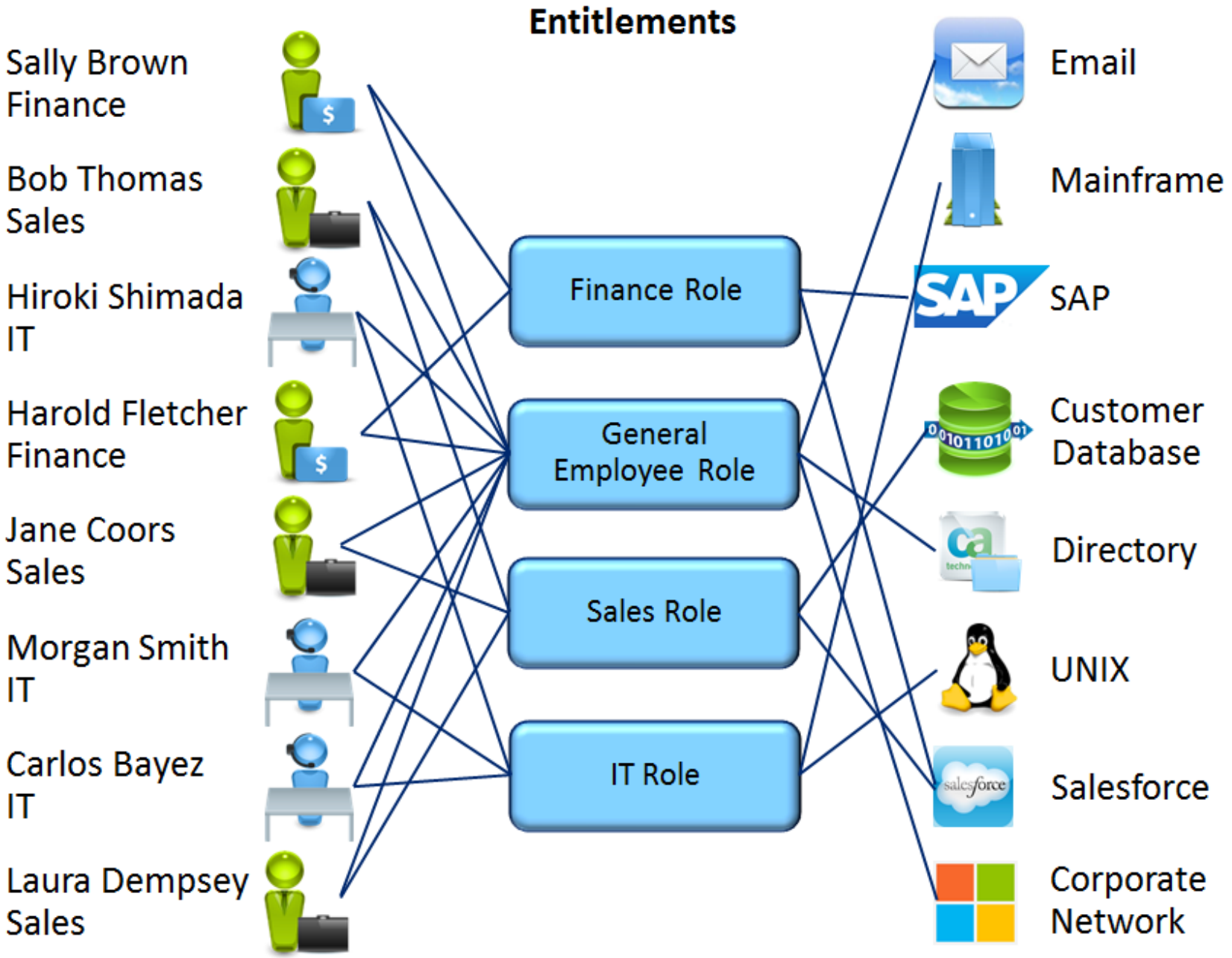
Access control binds roles to (meaningful) operations

- Operations are complex, meaningful system transactions
  - Not the ordinary, low-level read/write/execute actions on individual objects
- Operations can involve many individual lower-level objects

Not DAC or MAC

- Roles are dynamically assigned to subjects
  - For access control it matters the role played by the subject and not the subject's identity

D.F. Ferraiolo and D.R. Kuhn, "Role Based Access Control",
15th National Computer Security Conference, Baltimore, October 1992


Entitlements
Sally Brown
Finance
Bob Thomas
Sales
Hiroki Shimada
IT
Harold Fletcher
Finance
Jane Coors
Sales
Morgan Smith
IT
Carlos Bayez
IT
Laura Dempsey
Sales
Finance Role
General
Employee Role
Sales Role
IT Role
Email
Mainframe
SAP
Customer
Database
Directory
UNIX
Salesforce
Corporate
Network

Access control kinds: RBAC rules (1/2)

Role assignment

All subject activity on the system is conducted through transactions

- And transactions are allowed to specific roles
- Thus all active subjects are required to have some active role

A subject can execute a transaction if it has selected or been assigned a role which can use the transaction

Access control kinds: RBAC rules (2/2)

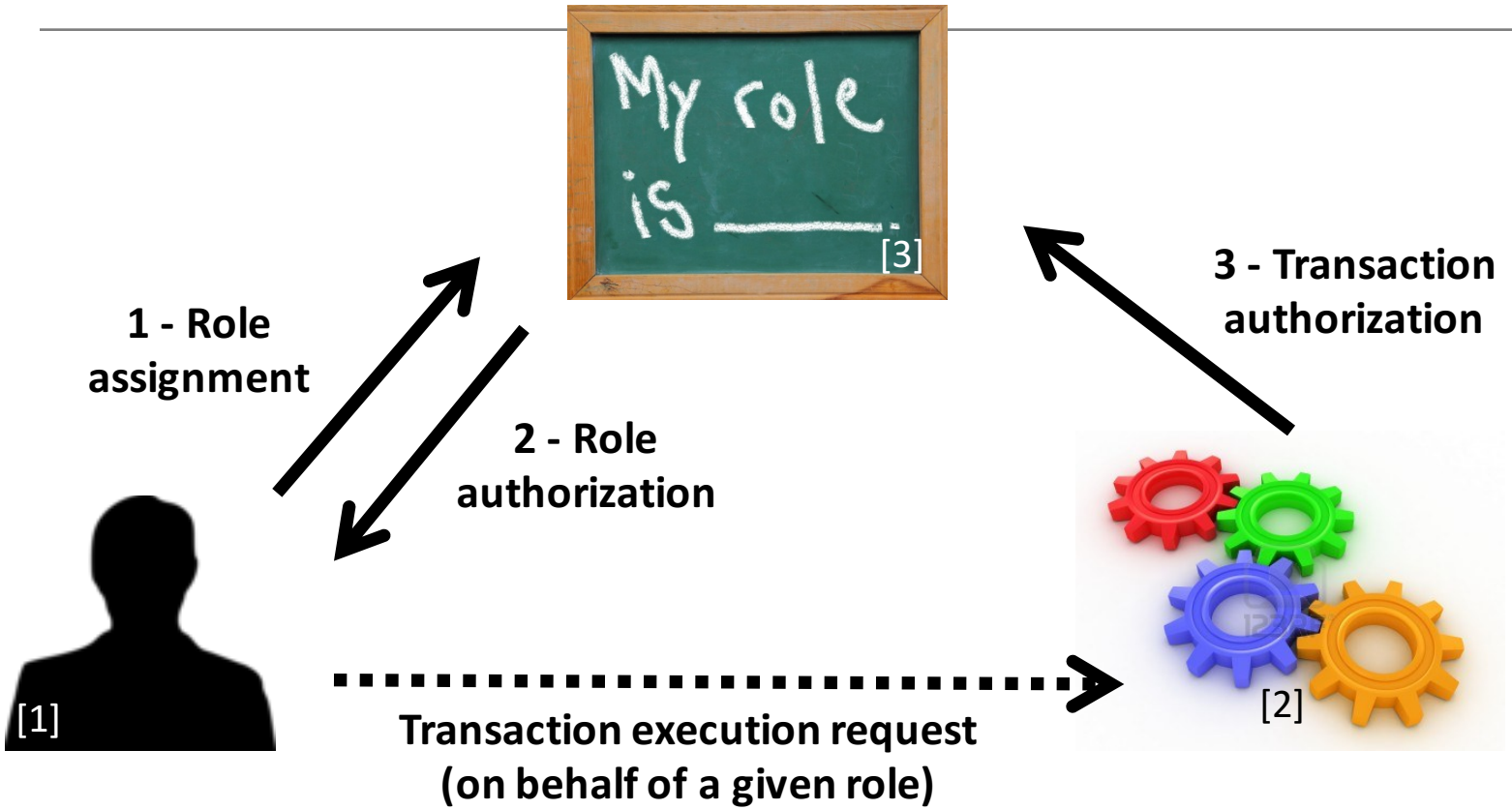
Role authorization

- A subject's active role must be authorized for the subject

Transaction authorization

- A subject can execute a transaction if
  - the transaction is authorized through the subject's role memberships
- and
  - there are no other constraints that may be applied across subjects, roles, and permissions

RBAC rules



[1] From http://www.clker.com/clipart-24011.html
[2] From http://www.123rf.com/photo_12115593_three-dimensional-colored-toothed-wheels.html
[3] From http://www1.yorksolutions.net/Portals/115255/images/MyRoleIs.jpg

RBAC: Roles vs. groups

Roles are a collection of permissions

- The permissions are granted to the subjects that, at a given instant, play the role
- A subject can only play a role at a given time

Groups are a collection of users

- And permissions can be granted both to users and groups
- A subject can belong to many groups at a given time

The session concept

- Role assignment is similar to a session activation
- Group membership is ordinarily a static attribute

Access control kinds: Context-Based Access Control (CBAC)

Access rights have an historical context

- The access rights cannot be determined without reasoning about past access operations
- Example:
  - Object locking allows file modification
  - Stateful Packet Filter firewall allows traffic related to other accepted traffic (e.g FTP)

Chinese Wall policy

- Isolate conflict groups
- Access control policies need to address past accesses to objects in different members of conflict groups

D.F.C. Brewer and M.J. Nash, "The Chinese Wall Security Policy ",
IEEE Symposium on Security and Privacy, 1989

Separation of duties

Fundamental security requirement for fraud and error prevention

- Dissemination of tasks and associated privileges for a specific business process among multiple subjects
- Often implemented with RBAC

Damage control

- Segregation of duties helps reducing the potential damage from the actions of one person
- Some duties should not be combined into one position

R.A. Botha, J.H.P. Eloff, "Separation of duties for access control enforcement in workflow environments", *IBM Systems Journal, 2001*

Segregation of duties:
ISACA (Inf. Systems Audit and Control Ass.) Matrix guideline

Exhibit 2.9—Segregation of Duties Control Matrix

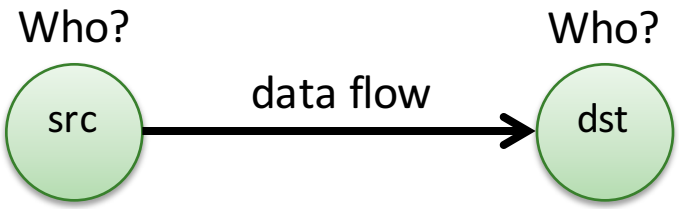
	Control Group	Systems Analyst	Application Programmer	Help Desk and Support Manager	End User	Data Entry	Computer Operator	Database Administrator	Network Administrator	Systems Administrator	Security Administrator	Systems Programmer	Quality Assurance
Control Group		X	X	X		X	X	X	X	X		X	
Systems Analyst	X			X	X		X				X		X
Application Programmer	X			X	X	X	X	X	X	X	X	X	X
Help Desk and Support Manager	X	X	X		X	X		X	X	X		X	
End User		X	X	X			X	X	X			X	X
Data Entry	X		X	X			X	X	X	X	X	X	
Computer Operator	X	X	X		X	X		X	X	X	X	X	
Database Administrator	X		X	X	X	X	X		X	X		X	
Network Administrator	X		X	X	X	X	X	X					
System Administrator	X		X	X		X	X	X				X	
Security Administrator		X	X			X	X					X	
Systems Programmer	X		X	X	X	X	X	X		X	X		X
Quality Assurance		X	X		X							X	

X—Combination of these functions may create a potential control weakness.

Information flow models

Authorization is applied to data flows

- Considering the data flow source and destination
- Goal: avoid unwanted/dangerous information flows (e.g. Leaks)



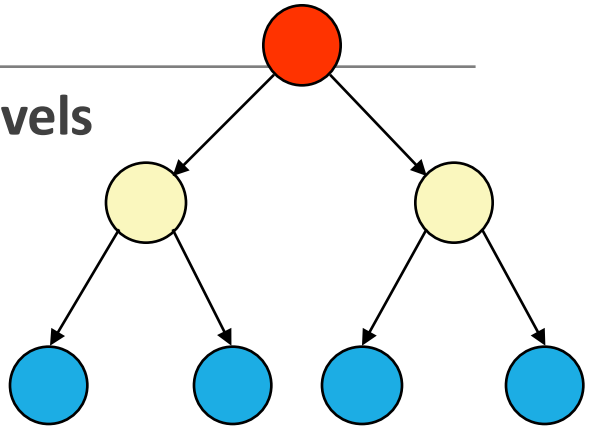
Src and Dst security-level attributes

- Information flows should occur only between entities with given **security-level** attributes
- Authorization is given based on the **SL** attributes

Multilevel security

Subjects (or roles) act on different security levels

- Levels do not intersect themselves
- Levels have some partial order
  - Hierarchy
  - Lattice

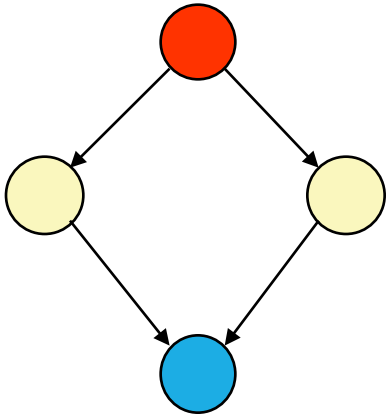


Levels are used as attributes of subjects and objects

- Subjects: security level clearance
- Objects: security classification

Information flows & security levels

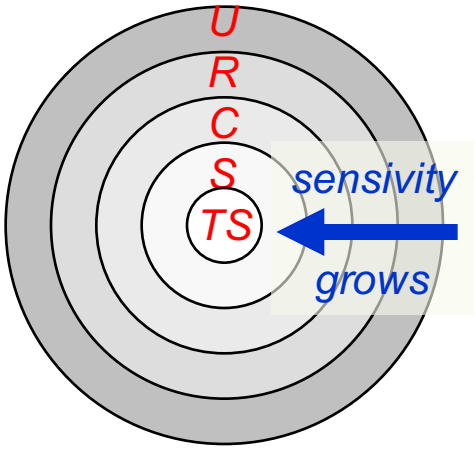
- Same security level → authorized
- Different security levels → controlled
  - Authorized or denied on a "need to now" basis



Multilevel security levels: Military environments / Int. organizations

Typical levels

- Top secret
- Secret
- Confidential
- Restricted
- Unclassified



EU example

- EU TOP SECRET
- EU SECRET
- EU CONFIDENTIAL
- EU RESTRICTED
- EU COUNCIL / COMMISSION

Portugal (NTE01, NTE04)

- Muito Secreto
- Secreto
- Confidencial
- Reservado

NATO example:

- COSMIC TOP SECRET (CTS)
- NATO SECRET (NS)
- NATO CONFIDENTIAL (NC)
- NATO RESTRICTED (NR)

Multilevel security levels: Civil organizations

Typical levels

- Restricted
- Proprietary
- Sensitive
- Public

It is vital that the organization actually enforces the levels!

- Frequently, classification is wrong, or access control is not enforced
  - E.g. https://www.google.pt/search?q=confidential+presentation+filetype%3Apdf

Security categories (or compartments)

Self-contained information environments

- May span several security levels

Military environments

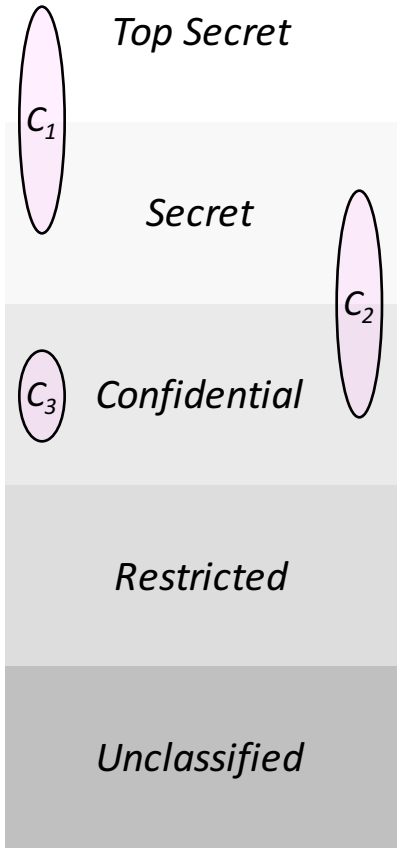
- Military branches, military units

Civil environments

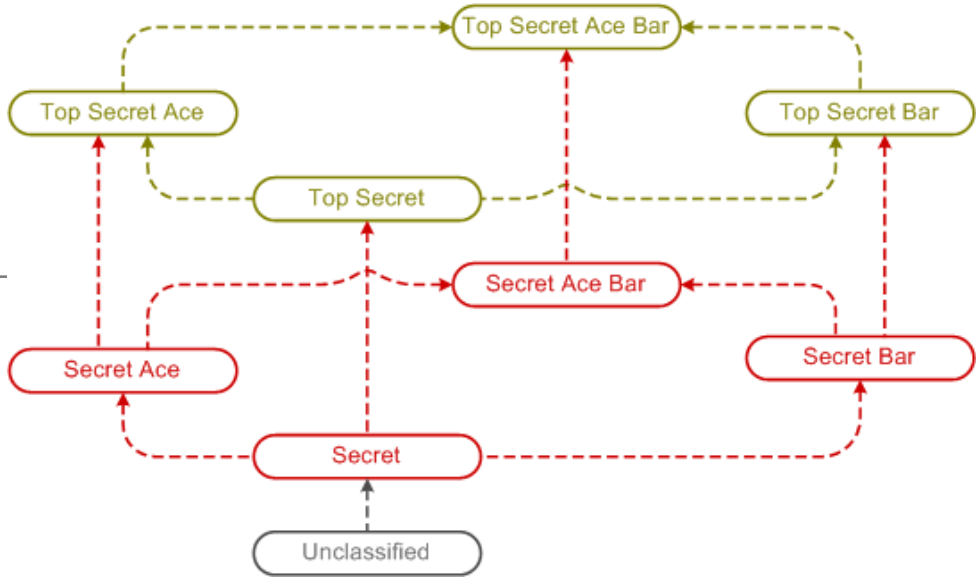
- Departments, organizational units, projects

An object can have belong to different compartments and have a different security classification in all them

- (top-secret, crypto), (secret, weapon)



Security labels



Label = Category + Level

- Lb1 = Secret Ace
- Lb2 = Top Secret Ace Bar

Relative order between labels

$Lb1 \leq Lb2 \Rightarrow C1 \subseteq C2 \wedge Lv1 \leq Lv2$

Labels form a lattice

- Secret Ace
  - Access: Secret, Secret Ace, Unclassified
  - Doesn't access: Secret Bar, all others

Bell-LaPadula MLS Model

Access control policy for controlling information flows

- Addresses data confidentiality and access to classified information
- Addresses disclosure of classified information
  - Object access control is not enough
  - We need to restrict the flow of information from a source to authorized destinations

Uses a state-transition model

- In each state there are subjects, objects, an access matrix and the current access information
- State transition rules
- Security levels and clearances
  - Objects have a security labels
  - Subjects have security clearances
  - Both refer to security levels (eg. CONFIDENTIAL)

D. Elliott Bell, Leonard J. La Padula, "Secure Computer Systems: Mathematical Foundations", *MITRE Technical Report 2547, Volume I, 1973*

Bell-LaPadula MLS Model: Secure state-transition model

Simple security condition (no read up)

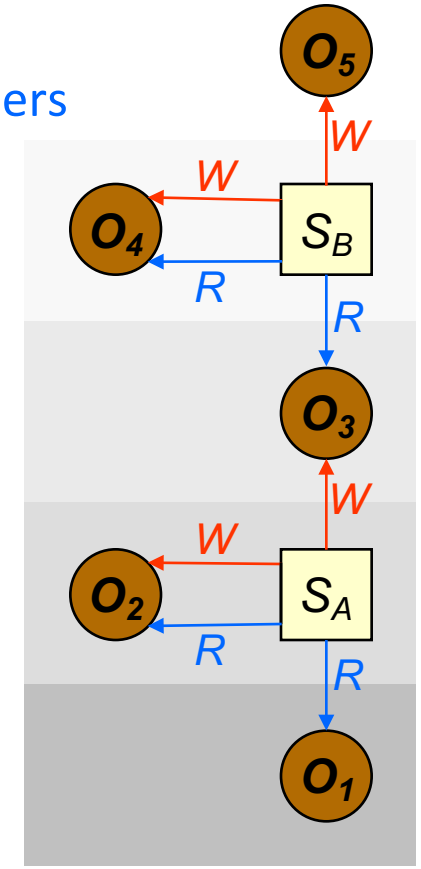
- S can read from O **if** L(S) ≥ L(O)
- … Team Leaders can access documents from programmers
- … but cannot access information from managers

★-property (no write down)

- S can write to O **if** L(S) ≤ L(O)
- … Team Leaders can produce documents to managers
- … but cannot disclose information to programmers
- aka Confinement property

Discretionary Security Property

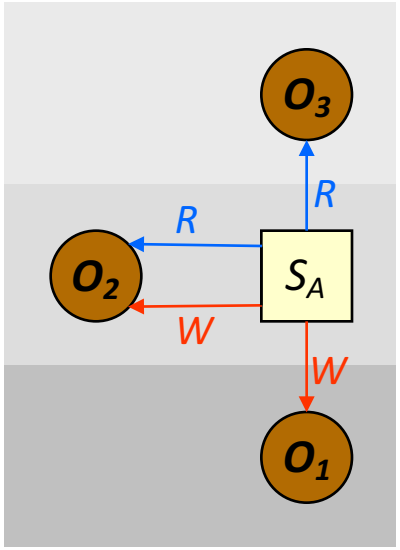
- DAC-based access control at the same level



Biba Integrity Model

Access control policy for controlling information flows

- For enforcing data integrity control
- Uses **integrity levels**, not security levels
- **Focus on Integrity**, not confidentiality



Similar to Bell-LaPadula, with inverse rules

- Simple Integrity Property (no read down)
  - S can read O if I(S) ≤ I(O)
  - … One can only read content from layers of higher integrity
- Integrity ✯-Property (no write up)
  - S can write O if I(S) ≥ I(O)
  - … One can only write content to levels of lower integrity

K. J. Biba, "Integrity Considerations for Secure Computer Systems",
MITRE Technical Report 3153, The Mitre Corporation, April 1977

Biba and Bell-LaPadula

Biba: No read down, no write up to maintain **integrity**

LaPadula: No write down, no read up to keep **confidentiality**

Combining both models in parallel => Same level access only?

Compromises either leak information or threaten integrity...
Which usually is required in real world.

Wrong! Models are orthogonal!

Clark-Wilson Integrity Model

Addresses information integrity control

- Uses the notion of transactional data transformations
- Separation of duty: transaction certifiers ≠ implementers
- Developed for commercial activities

Terminology

- Data items
  - Constrained Data Item (CDI)
    - Can only be manipulated by TPs
  - Unconstrained Data Item (UDI)

D. D. Clark, D. R. Wilson, "A Comparison of Commercial and Military Computer Security Policies", IEEE Symposium on Security and Privacy, 1987

Clark-Wilson Integrity Model

Terminology (cont…)

- Integrity policy procedures
  - Integrity Verification Procedure (IVP)
    - Ensures that all CDIs conform with the integrity specification
  - Transformation Procedure (TP)
    - Well-formed transaction
      - Take as input a CDI or a UDI and produce a CDI
    - Must guarantee (via certification) that transforms all possible UDI values to "safe" CDI values

Key terms

- Tampered: Data cannot be tampered
- Logged: Everything is logged
- Consistent: CDIs are always consistent

D. D. Clark, D. R. Wilson, "A Comparison of Commercial and Military Computer Security Policies", IEEE Symposium on Security and Privacy, 1987

Clark-Wilson Integrity Model: Certification & Enforcement

Integrity assurance

- Certification
  - Relatively to the integrity policy
- Enforcement

Two sets of rules

- Certification Rules (C)
  - Certify that a specific CDI was properly verified and is consistent
- Enforcement Rules (E)
  - Defines how the system should maintain the consistent of the CDI in future transactions

Clark-Wilson Integrity Model: Certification & Enforcement rules

Basic rules:

- **C1**: when an IVP is executed, it must ensure that all CDIs are valid
- **C2**: for some associated set of CDIs, a TP must transform those CDIs from one valid state to another
- **E1**: the system must maintain a list of certified relations and ensure only TPs certified to run on a CDI change that CDI

Separation of duty (external consistency)

- **E2**: the system must associate a user with each TP and set of CDIs. The TP may access CDIs on behalf of the user if authorized
- **C3**: allowed user-TP-CDI relations must meet "separation of duty" requirements

Identification gathering

- **E3**: the system must authenticate every user attempting a TP (on each attempt)

Audit trail

- **C4:** all TPs must append to a log enough information to reconstruct operations

UDI processing

- **C5**: a TP taking a UDI as input may only perform valid transactions for all possible values of the UDI. The TP will either accept (convert to CDI) or reject the UDI

Certification constraints

- **E4**: only the certifier of a TP may change the associated list of entities

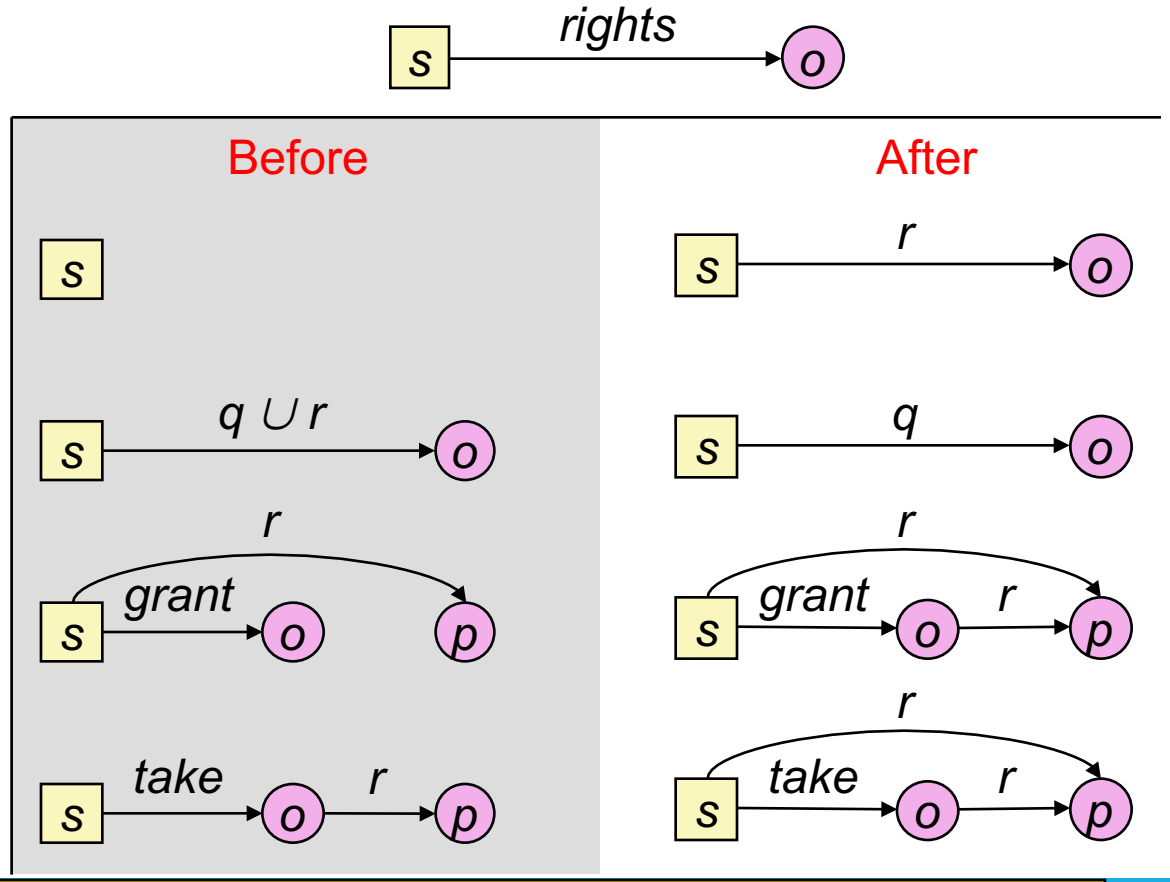
Take-grant system

4 primitive operations

- Create (o,r)
- Revoke (o,r)
- Grant (o,p,r)
- Take (o,p,r)

Special rights

- Grant right
- Take right



A. Jones, R. Lipton, L. Snyder, "A Linear Time Algorithm for Deciding Security",
Proc. 17th Annual Symposium on the Foundations of Computer Science, October 1976