

Lec 18: Access Control (2)

CSED415: Computer Security
Spring 2025

Seulbae Kim



Administrivia

- Lab 04 is due on ~~Apr 25~~ May 2!
 - Due dates shown on Lec 15 and 16 slides were incorrect

Recap

- Discretionary Access Control: Owner decides everything!
 - Owner of a resource decides who can access what and how
 - Strengths
 - Flexibility: Users can instantly share or revoke access without system's help
 - Psychological acceptability: "My file, my rules"

Problem of DAC

- Once access is granted, the owner cannot prevent recipients from re-sharing the data
 - Example:
 - Alice owns file (f_A). She sets the permissions to: `rw- r-- ---`
 - Owner: Alice, Group: Friends
 - Bob is in the Friends group. Therefore, Bob can read Alice's file
 - Claire is not in that group. She cannot read the file
 - Bob duplicates Alice's file to f_b
 - Bob sets f_b 's permissions to `rw- rw- r--`
 - Now Claire can see the data in Alice's file

Mandatory Access Control (MAC)

Two problems with DAC

- Information-flow control problem:
 - You cannot stop users from copying user data to a wider audience
- Administrative issue:
 - In many organizations (e.g., a company), administrators should decide how the organization's sensitive data can be shared
 - Users should not be able to control anything

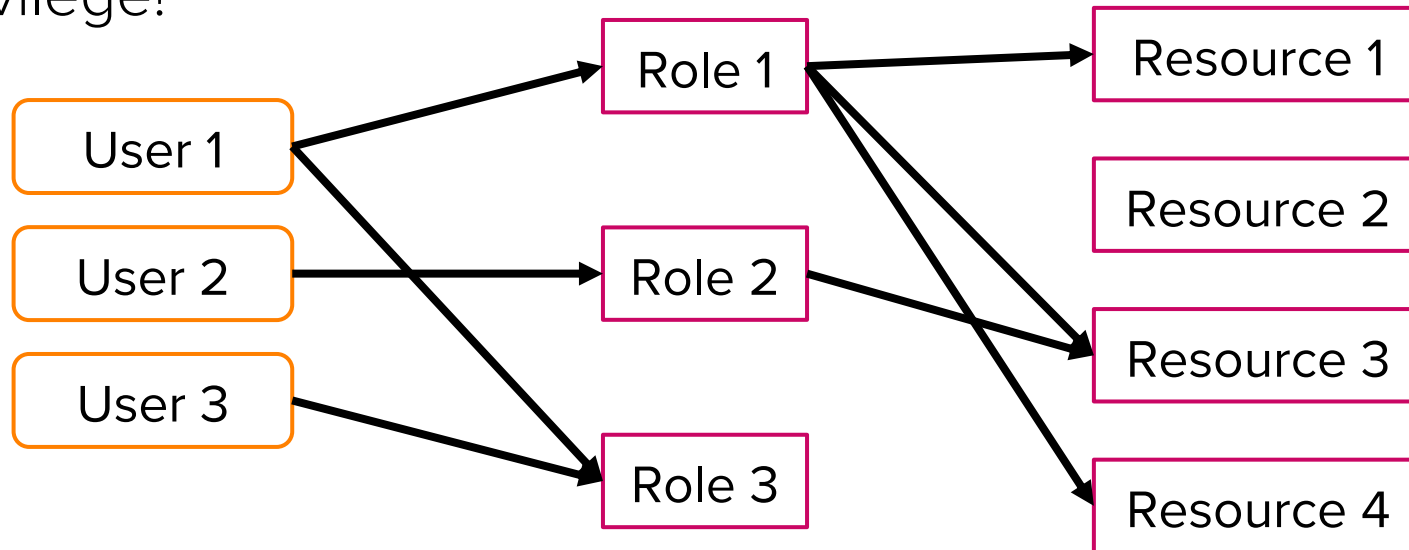
Mandatory Access Control (MAC) helps address these problems by assigning system-wide rules that users cannot override

Mandatory Access Control (MAC)

- Core idea:
 - Assign additional **attributes** to subjects and objects
 - Access is controlled based on the attributes
 - A system-wide policy checks these attributes on every access
 - Users cannot override it
 - Hence, “mandatory”

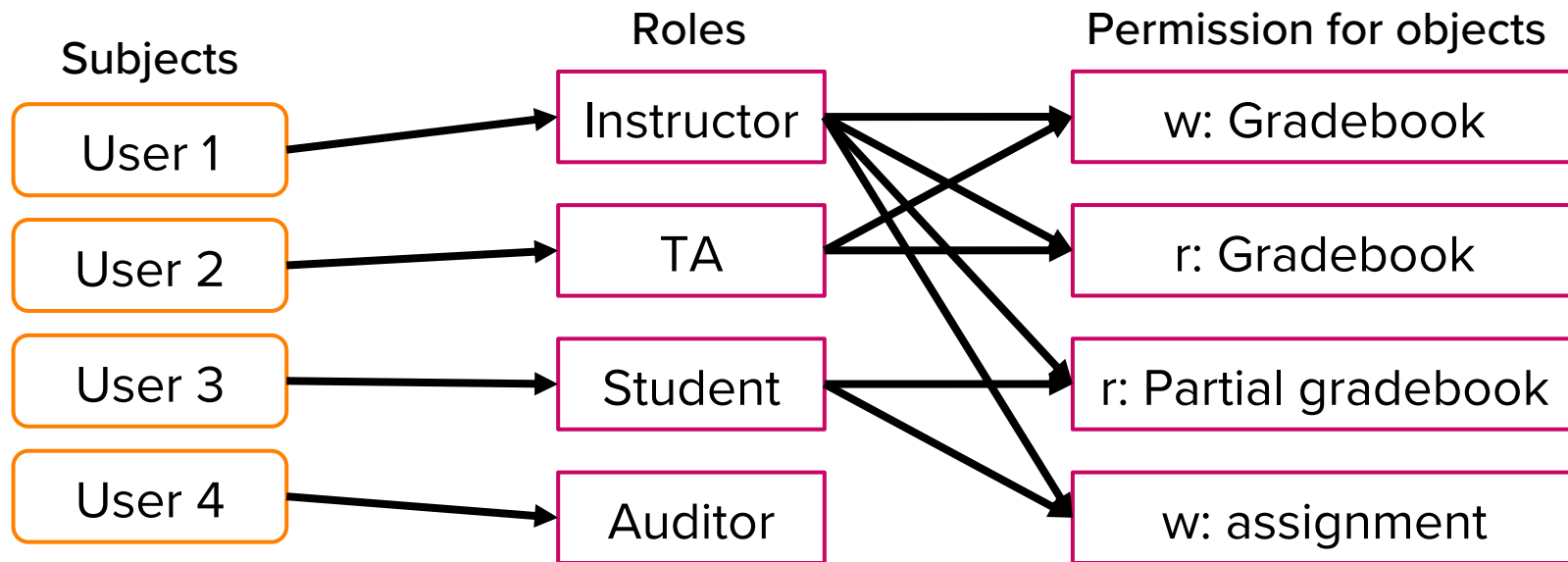
Role-based Access Control (RBAC)

- Attributes can be “roles”
 - Assign multiple roles to users
 - Each role is associated with a different permission
 - RBAC controls access based on the roles of the users
 - Least privilege!



Role-based Access Control (RBAC)

- Attributes can be “roles”
 - Example: CSED415



Q) Can User 2 create a copy of the gradebook and let User 3 read it?

A) No. Copied objects inherit the original role-based permissions.
User 3 (role: Student) cannot access the copy.

Attribute-based Access Control (ABAC)

- ABAC is a generalization of MAC
 - Three key elements
 - Attributes: Defined (or naturally determined) for entities
 - Age, department, login time, etc.
 - Policy: Defines access policies for attributes
 - e.g., “If **age** is over 21, allow access to **alcohol**”
 - Architecture model: Defines the relationship between policies
 - Flexible and expressive!
 - Attributes are dynamic → We will see an example

Attribute-based Access Control (ABAC)

- RBAC vs ABAC Example: Movie Rating

Movie Rating	Allowed Viewrs
R	Age 17+
PG-13	Age 13+
G	Everyone

- If using RBAC

- Roles:

- Adult, Juvenile, or Child

- Permissions:

- Can view R-rated movies, can view PG-13-rated movies, and can view G-rated movies

- Policy:

- Adult role gets assigned all three permissions
 - Juvenile role gets permissions for PG-13- and G-rated movies
 - Child role gets permission for G-rated movies

User-to-role assignment and role-to-permission assignments are manual

Management of roles is also manual (e.g., need to change role when a child turns 17)

Attribute-based Access Control (ABAC)

- RBAC vs ABAC Example: Movie Rating

Movie Rating	Allowed Viewrs
R	Age 17+
PG-13	Age 13+
G	Everyone

- If using ABAC

- Do not need explicitly defined roles

- Permissions:

- Can view R-rated movies, can view PG-13-rated movies, and can view G-rated movies

- Policy:

- if (get_age(user) >= 17) return {R, PG-13, G}
 - else if (get_age(user) >= 13) return {PG-13, G}
 - else return {G}

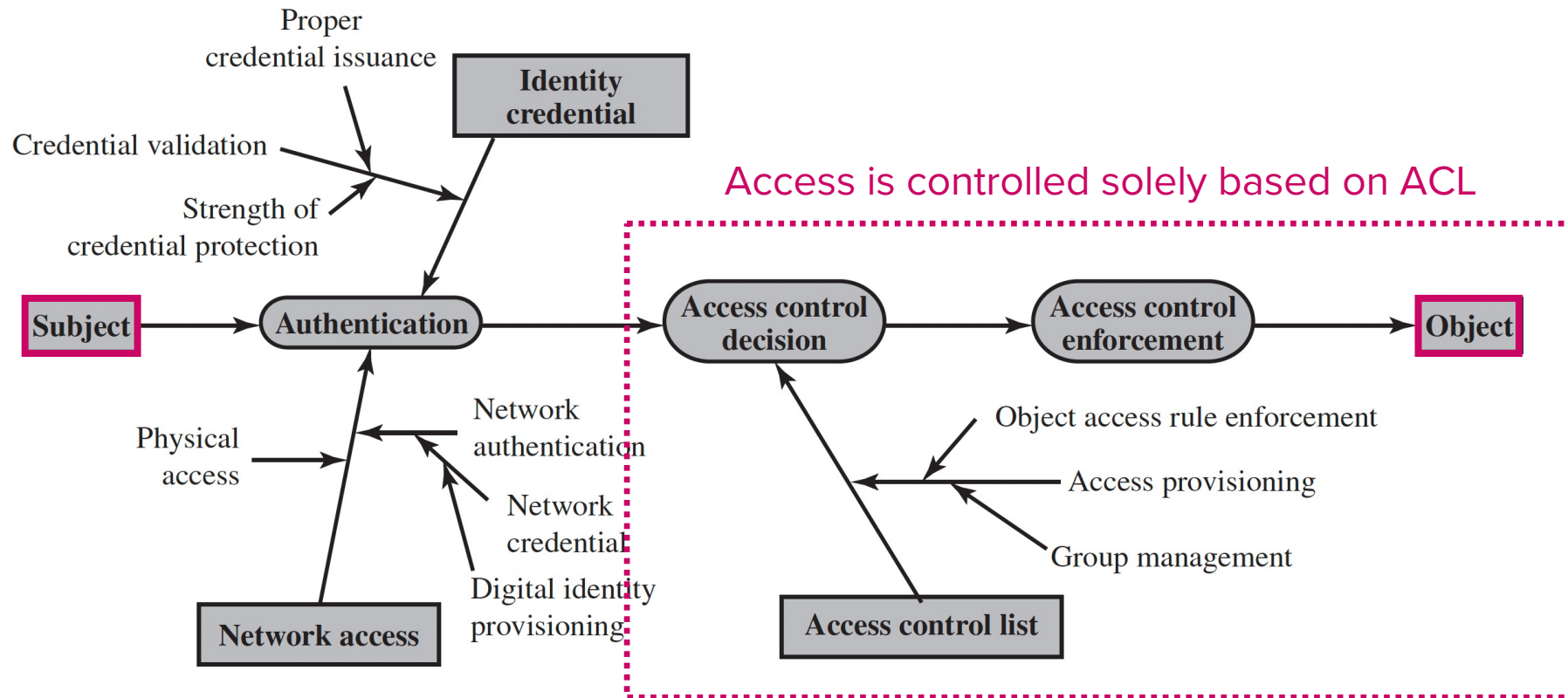
Only need to define policy with subject's attributes (age)

Do not need to redefine/manage static roles (people age automatically)

Additional ratings can be readily handled (e.g., VIP-only-rating → add one policy)

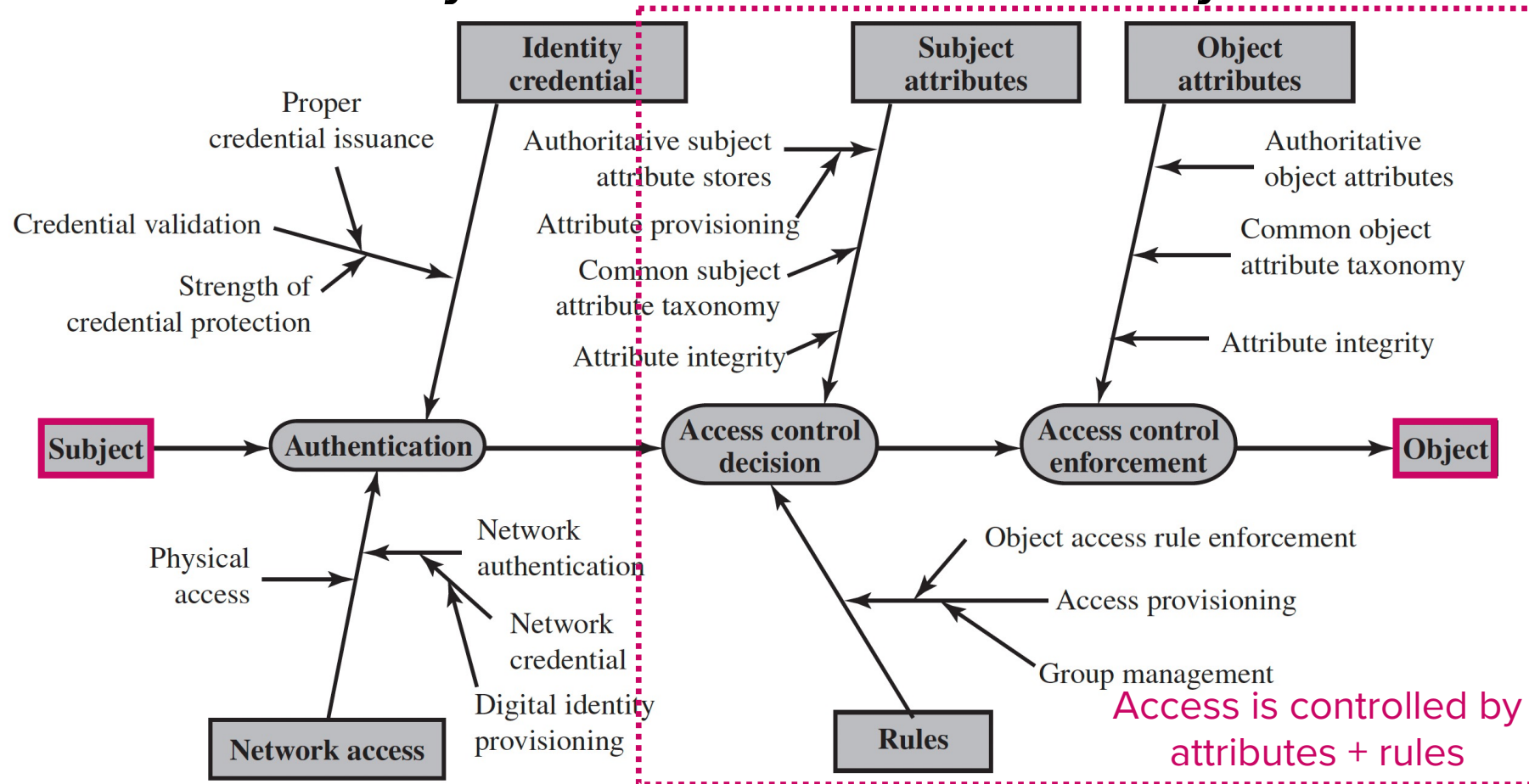
Attribute-based Access Control (ABAC)

- DAC vs ABAC – Subject wants to access Object



Attribute-based Access Control (ABAC)

- DAC vs ABAC – Subject wants to access Object



Access control using ABAC (MAC)

MAC for Sensitive Data

Security labels & clearances

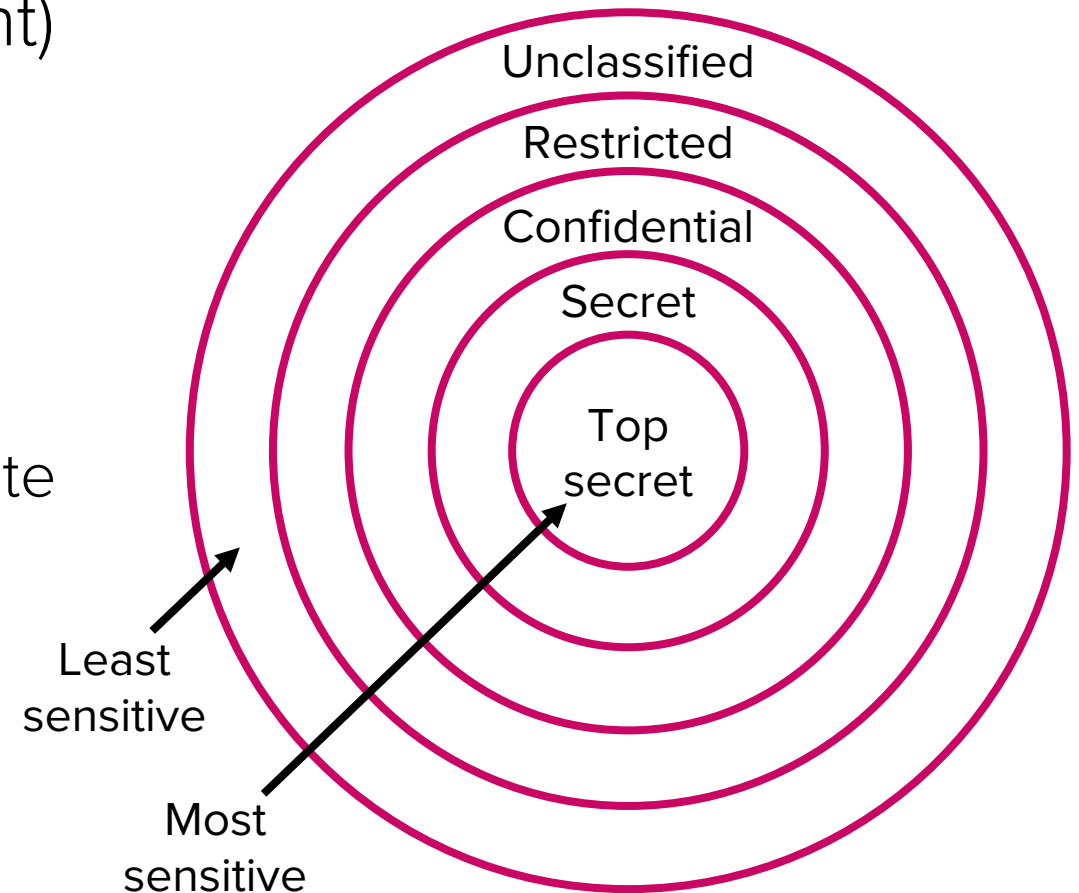
- Label:
 - Metadata that describes the nature of resource (e.g., attribute)
 - Indicate sensitivity, category, and clearance requirements of users
 - How sensitive is the data?
 - What kind of data is contained in the object?
 - OS associates labels with each user and object
- Clearance:
 - What a user/process is cleared to read/write
 - e.g., Alice has a top-secret clearance

Security labels & clearances

- Label-based access control:
 - Compare the security label of an object with the security clearance of a subject for access control
 - Label indicates how sensitive a resource is
 - Clearance indicates how eligible a subject is

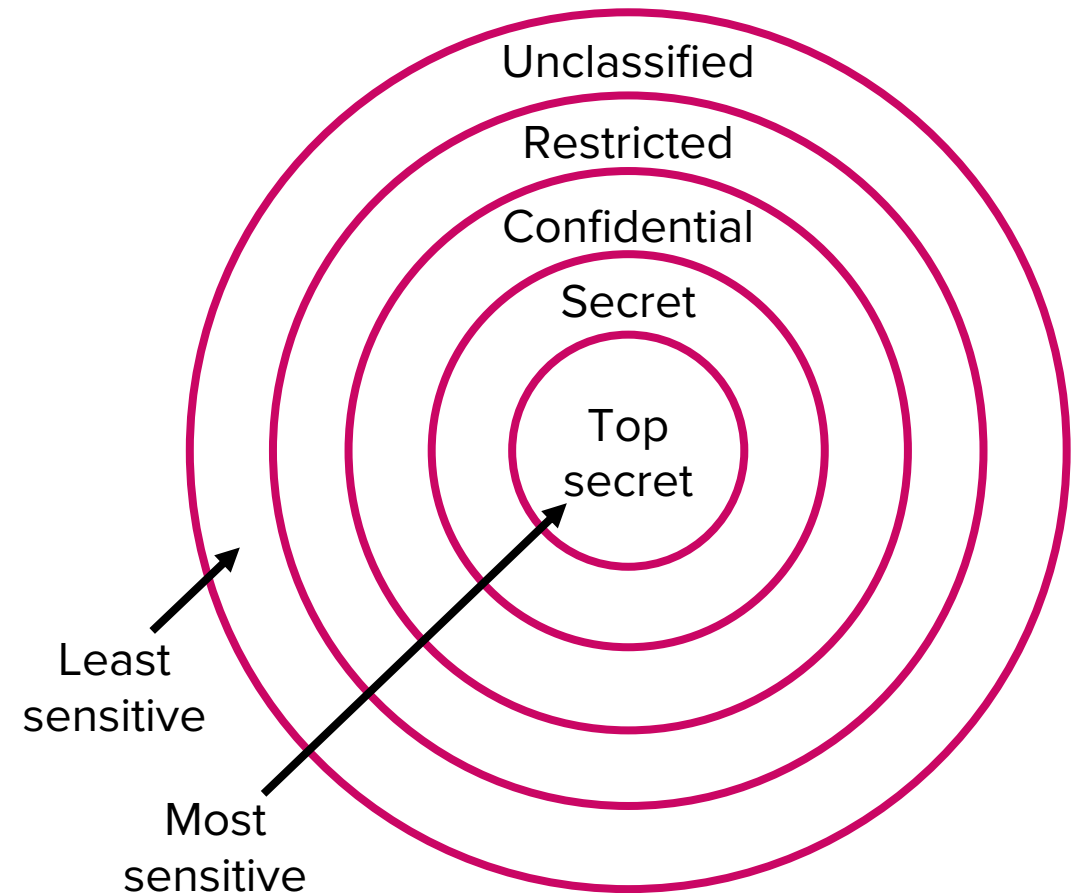
Example: Military security policy

- US Department of Defense environment:
 - Label = (sensitivity level, compartment)
 - e.g., weapon documents
 - Label 1 = (TS, {nuclear, chemical})
 - Label 2 = (S, {nuclear, missile})
 - Based on these labels, users can be authorized to read or write



Comparing labels

- Sensitivity levels are totally ordered
 - i.e., $TS > S > C > R > U$
- Compartments are sets that are partially ordered
 - nuclear \in weapon
 - chemical \in weapon
 - nuclear ?? chemical



Comparing labels

- levels (l_i) are compared by their orders
- Compartments (c_j) are compared using containment
- Example: $L_1 = (l_1, c_1)$, $L_2 = (l_2, c_2)$

L_1 dominates L_2	$l_1 > l_2$ and $c_1 \supseteq c_2$
-----------------------	-------------------------------------

L_1 is dominated by L_2	$l_1 < l_2$ and $c_1 \subseteq c_2$
-----------------------------	-------------------------------------

L_1 is equivalent to L_2	$l_1 = l_2$ and $c_1 = c_2$
------------------------------	-----------------------------

L_1 and L_2 are not comparable	All other cases
------------------------------------	-----------------

Ordering labels

$TS > S > C > R > U$

POSTECH

- By comparing the labels, we can order them

- Example:

- $L_1 = (TS, \{CSE, EE, ME\})$
- $L_2 = (S, \{CSE, EE\})$
- $L_3 = (S, \{EE, PHY\})$
- $L_4 = (C, \{CSE, PHY\})$

→ Q) Find all dominations? L_1 dominates L_2

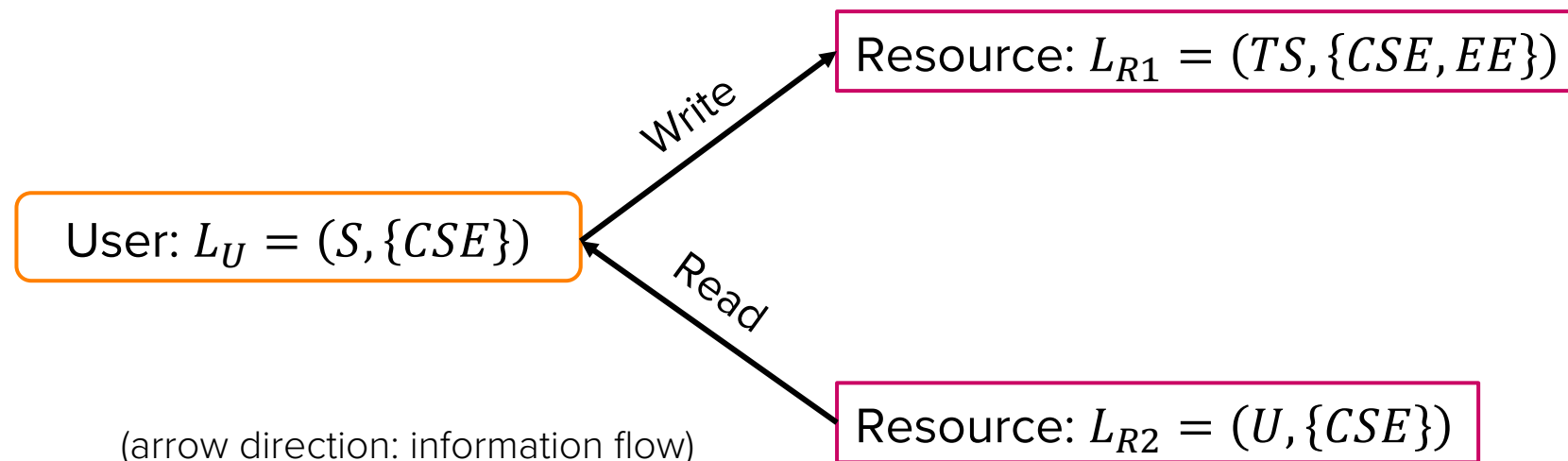
All other pairs of labels are not comparable

Bell-LaPadula (BLP) model

$TS > S > C > R > U$

POSTECH

- Access control model focusing on confidentiality
 - RDWU rules
 - **Read-down rule:** User with L_1 clearance can read document with label L_2 only when L_1 is equivalent to or dominates L_2
 - **Write-up rule:** User with L_1 clearance can write document with label L_2 when L_1 is equivalent to or is dominated by L_2

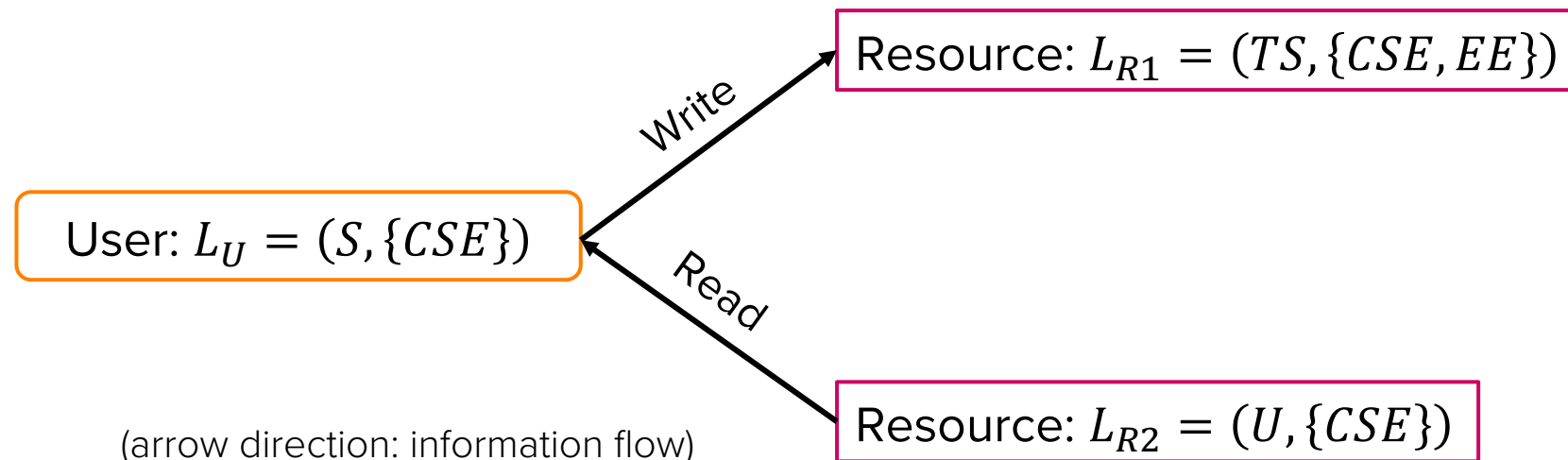


Bell-LaPadula (BLP) model

$TS > S > C > R > U$

POSTECH

- Access control model focusing on confidentiality
 - RDWU rules
 - Rationale: More sensitive information should not flow to users who do not clearance for that level
 - If write-down is allowed, user with high security clearance can leak information to lower security level users

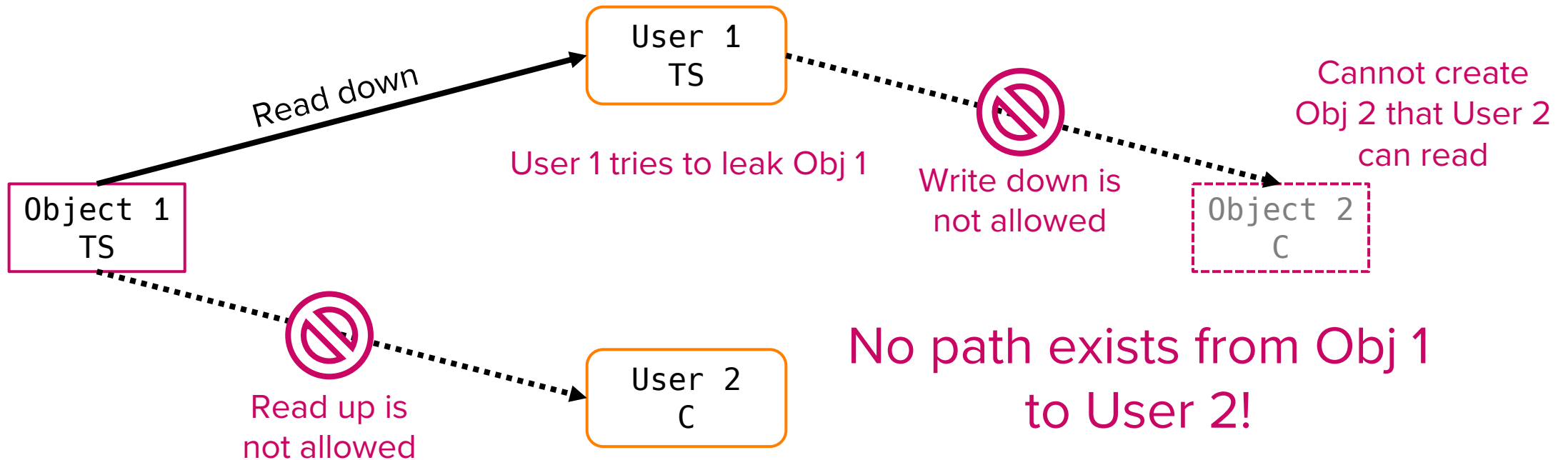


Bell-LaPadula (BLP) model

$TS > S > C > R > U$

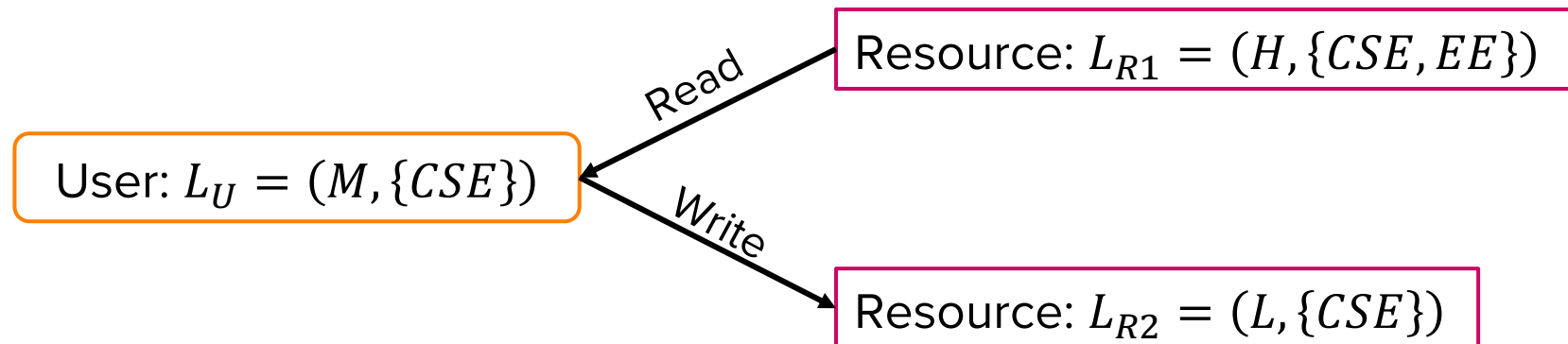
POSTECH

- BLP solves the information flow control problem
 - Confidential-level user (User 2) cannot read top secret Object 1
 - Top secret-level user (User 1) cannot create confidential Object 2



Biba Integrity model

- Access control model focusing on integrity
 - Integrity level defines the quality of information
 - Levels: High (trustworthy), medium (mediocre), or low (untrustworthy)
 - RUWD rules (opposite of BLP)
 - **Read-up rule:** Low integrity users can access high integrity information
 - **Write-down rule:** High integrity users can produce low integrity information



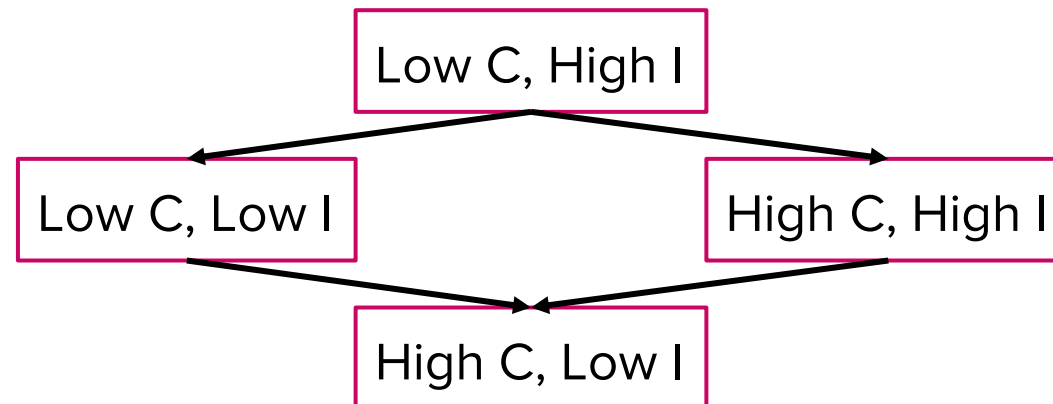
(arrow direction: information flow)

Biba Integrity model

- Access control model focusing on integrity
 - Example: You are a junior employee at a stock trading firm
 - RU: You may read market reports (high integrity) from your firm
 - No WR: You may not write or edit the market reports your firm publishes
 - No RD: You may not consult Reddit posts (low integrity) or tweets for market research
 - WD: You may tweet about markets on your twitter account

BLP and Biba are contradictory

- Bell-LaPadula: Read-down, write-up for confidentiality
- Biba Integrity: Read-up, write-down for integrity
- Can we combine the BLP and Biba models?
 - If a single label is used for both confidentiality and integrity, then the two models conflict
 - We need to use independent labels for confidentiality and integrity



MAC in Commercial Context

Models for commercial environments

- BLP / Biba Integrity models are intended for use in military settings where users (soldiers and officers) have clearances (labeled), and documents are classified (also labeled)
- MAC is also needed in commercial settings
 - Companies should limit how information can be shared
 - Challenges
 - Users usually do not have clearances
 - Labeling information is challenging

Clark-Wilson model

- Focus on integrity in commercial setting
 - “No user of the system, even if authorized, may be permitted to modify data items in such a way that assets or accounting records of the company are lost or corrupted”

Clark-Wilson model

- Two principles for data integrity
 1. Well-formed transaction
 - A user cannot manipulate data arbitrarily
 - Users are only allowed to make “transactions”
 - Transactions constrain the ways in which users can modify the data
 - Correspond to high-level operations that could be performed on data
 - e.g., `add_employee()`, `set_salary()`, `pay_salary()`, ...
 - All transactions are recorded in a write-only log

Data can only be manipulated through trusted code!

Clark-Wilson model

- Two principles for data integrity
 2. Separation of duty
 - Responsibilities are divided among different users
 - All operations are divided into subparts
 - Each subpart must be executed by a different person
 - e.g., Two-person rule for critical operations (such as launching a missile)
 - One person inserts a launch key
 - Another person types in a password



Clark-Wilson model

- Example: Placing an order
 1. A purchasing agent creates an order for a supply
 - The agent sends copies of the order to both the supplier and the logistics agent
 2. The supplier ships the goods to the logistics agent
 - The logistics agent conducts an integrity check to verify the correctness of the shipment (amount, quality, ...)
 3. A delivery confirmation is sent to finance department agent
 - The finance agent pays the supplier after reviewing both the order and delivery confirmation

Clark-Wilson model

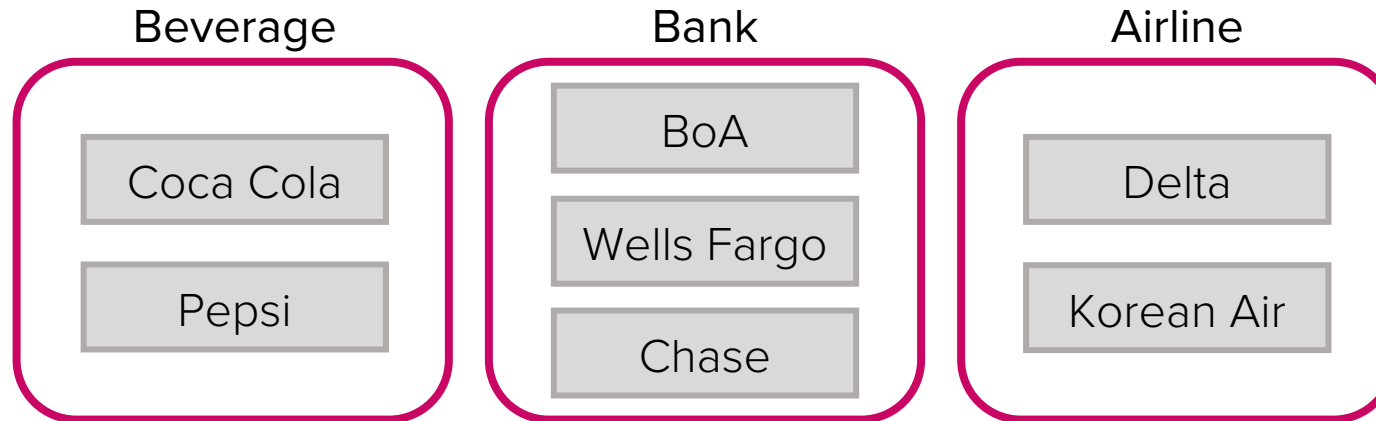
- Example: Placing an order
 - `<User, transaction, {data}>` triples:
 - `<Purchasing agent, place_order, {order book}>`
 - `<Logistics agent, receive_delivery, {inventory}>`
 - `<Finance agent, make_payment, {account balance}>`
 - Separation of duty:
 - Different agents for subparts of “order supply” operation
 - What if the same agent takes charge of the entire process?
 - Constraints:
 - The logistics agent must have the order before accepting delivery
 - The finance agent must have the delivery confirmation prior to payment

Chinese Wall policy

- Focus on confidentiality
 - Motivated by Conflict of Interest (Col) requirements
 - Example:
 - A law firm has many clients
 - Some clients have competitive relationships (e.g., Coca Cola and Pepsi)
 - Chinese Wall policy aims to avoid Col between competitors

Chinese Wall policy

- Focus on confidentiality
 - Conflicting groups



- Policy: User U can access object O that belongs to company C as long as U has not accessed any object from other companies in C 's conflicting group

MAC in practice: SELinux

What is SELinux?

- Security-Enhanced Linux is a security extension for Linux that introduces a mandatory access control (MAC) model
 - Historically, Unix-based systems have used DAC
 - Define access rights per ownership (user, group, and other)
 - Users have the ability (discretion) to change permissions of their own files
 - Nothing stops users from making bad decisions
(e.g., `$ chmod --recursive 777 /home/username`)
 - MAC policies, on the other hand, are administratively set and fixed
 - Provides better security by safeguarding (i.e., limiting the freedom of) users

How does SELinux work?

- Workflow
 - Linux kernel first applies DAC (owner, group, others, rwx, ...)
 - Then, SELinux evaluates accesses against its own security policies
 - Enforces BLP policies (RDWU)
 - SELinux acts as an additional layer of access control

How does SELinux work?

- Labeling and type enforcement
 - Labeling
 - Resources are labeled with an SELinux context: “`user:role:type:level`”
 - User: Individual users
 - Role: A group of specific users
 - Type: Abstract domain assigned to subjects and objects
 - Level: Sensitivity level (s0: least sensitive, s15: most sensitive)
 - Type enforcement
 - Each subject is assigned a type (domain)
 - Only certain operations are permitted for each type

How does SELinux work?

Label format: “user:role:type:level”

POSTECH

- Labeling and Type enforcement
 - e.g., Apache’s HTTPD web server
 - Server’s binary executable is of type `httpd_exec_t`

```
$ ls -lZ /usr/sbin/httpd
-rwxr-xr-x. root root system_u:object_r:httpd_exec_t:s0 /usr/sbin/httpd
```

- Directory with server configuration files is of type `httpd_config_t`

```
$ ls -dZ /etc/httpd
drwxr-xr-x. root root system_u:object_r:httpd_config_t:s0 /etc/httpd
```

- SELinux access control policy:
 - `allow httpd_exec_t httpd_config_t: file { read }`
(Subjects of type `httpd_exec_t` can read objects of type `httpd_config_t`)
(Even a root-privileged binary cannot access `/etc/httpd` without proper SELinux type)

SELinux Adoption

- Additional reference: SELinux coloring book
 - https://people.redhat.com/duffy/selinux/selinux-coloring-book_A4-Stapled.pdf
- Widely used in security-critical environments
 - Government, enterprise servers, ...
- OS support
 - Android, Fedora, Debian and Ubuntu, Red Hat Enterprise Linux, ...

Summary

- MAC (BLP, Biba, Clark-Wilson, Chinese Wall) enforces organization-wide policy and information-flow control
- Authentication and access control for system-level security
 - All users are authenticated first
 - Based on the authenticated identity, users can request access to a resource
 - Access control policies compares the access rights of the identity to access control policy (e.g., ACL)

Coming up next

- How does system-level security fall apart?
 - In other words, what can make strong authentication and access control meaningless?
 - Next topic: Malware and anti-malware

Questions?