Lec 20: Access Control (2)

CSED415: Computer Security

Spring 2024

Seulbae Kim

POSTECH

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Administrivia

- Lab 04 is due this weekend!
  - Questions?

Recap

- Discretionary Access Control: Owner decides all!
  - Owner of a resource decides how it can be shared
  - Owner can choose to give access rights to other users
  - Risk: The owner can never be sure that the sensitive data he/she shares with another user will not be further shared with others

Mandatory Access Control (MAC)

Two problems with DAC

- Information flow control problem:
    - You cannot control if someone you share a file with will not further share the data contained in it
- Administrative problem:
    - In many organizations (e.g., a company), a user should not decide how certain type of data can be shared
    - Typically, the employer decides how various types of sensitive data should be shared among employees

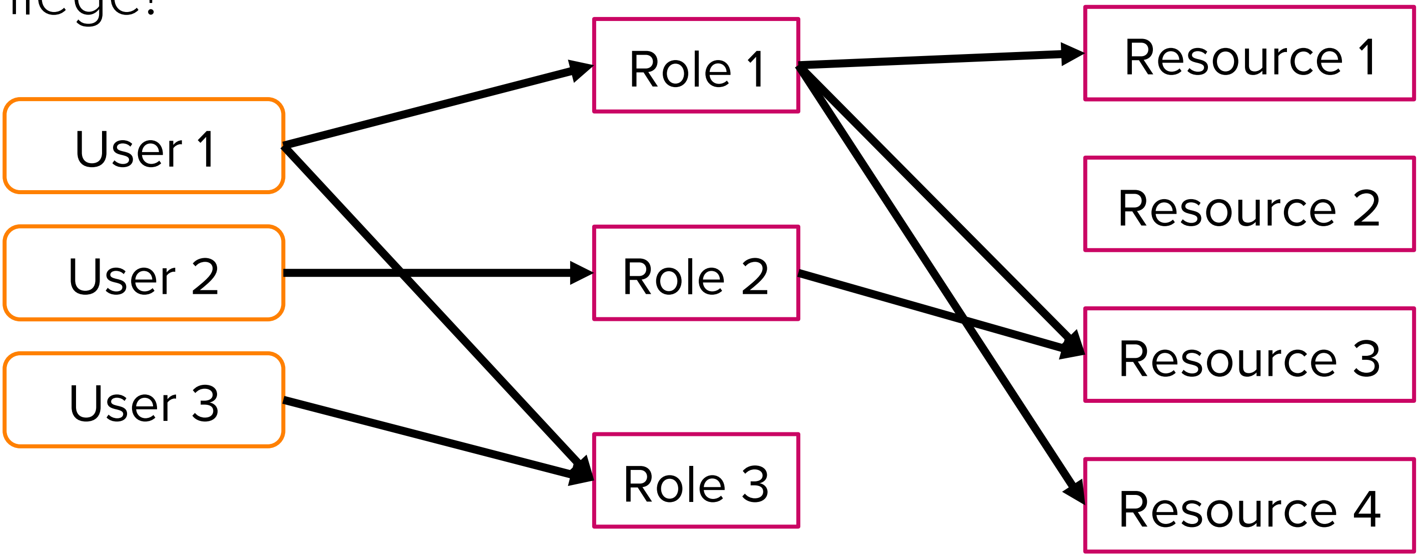
Mandatory Access Control (MAC) helps address these problems

Mandatory Access Control

- Idea:
  - Assign additional **attributes** to subjects and objects
  - Control access based on the attributes
- The system globally imposes MAC policy
  - Hence, “mendatory”
  - Subjects (or object owners) cannot change that policy

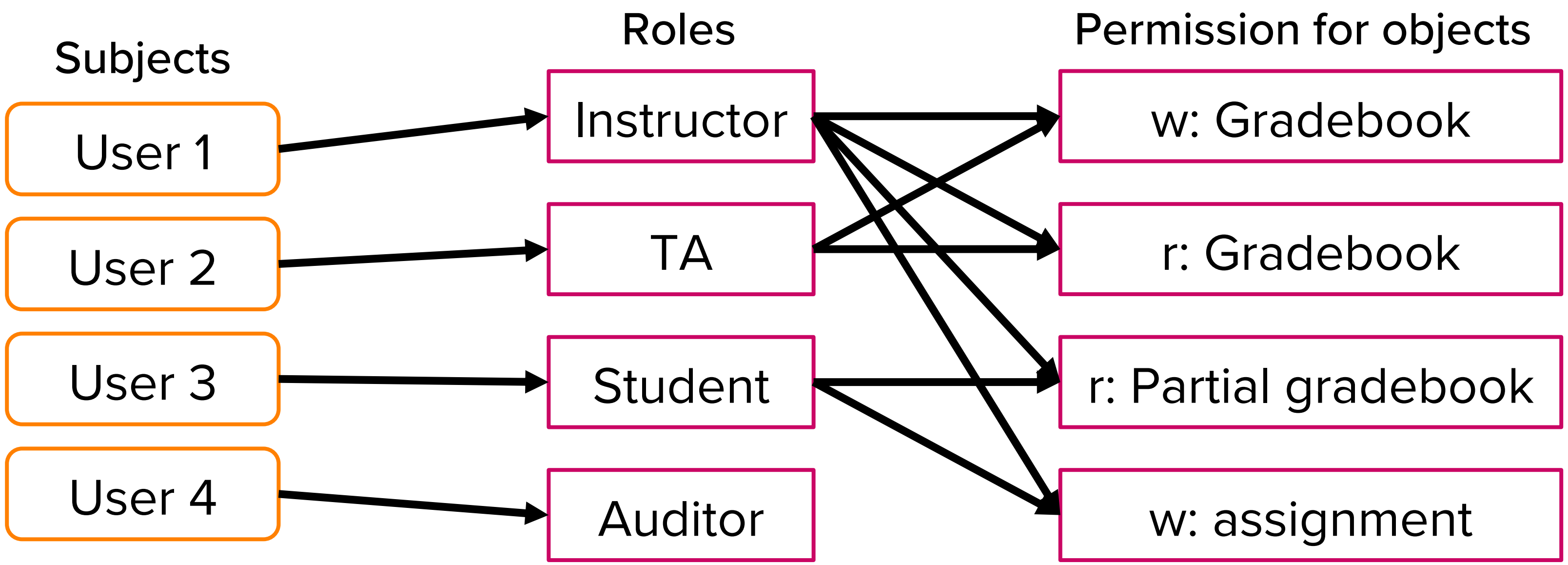
Role-based Access Control (RBAC)

- Attributes can be “roles”
  - RBAC assigns 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



Subjects

User 1

User 2

User 3

User 4

Roles

Instructor

TA

Student

Auditor

Permission for objects

w: Gradebook

r: Gradebook

r: Partial gradebook

w: assignment

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

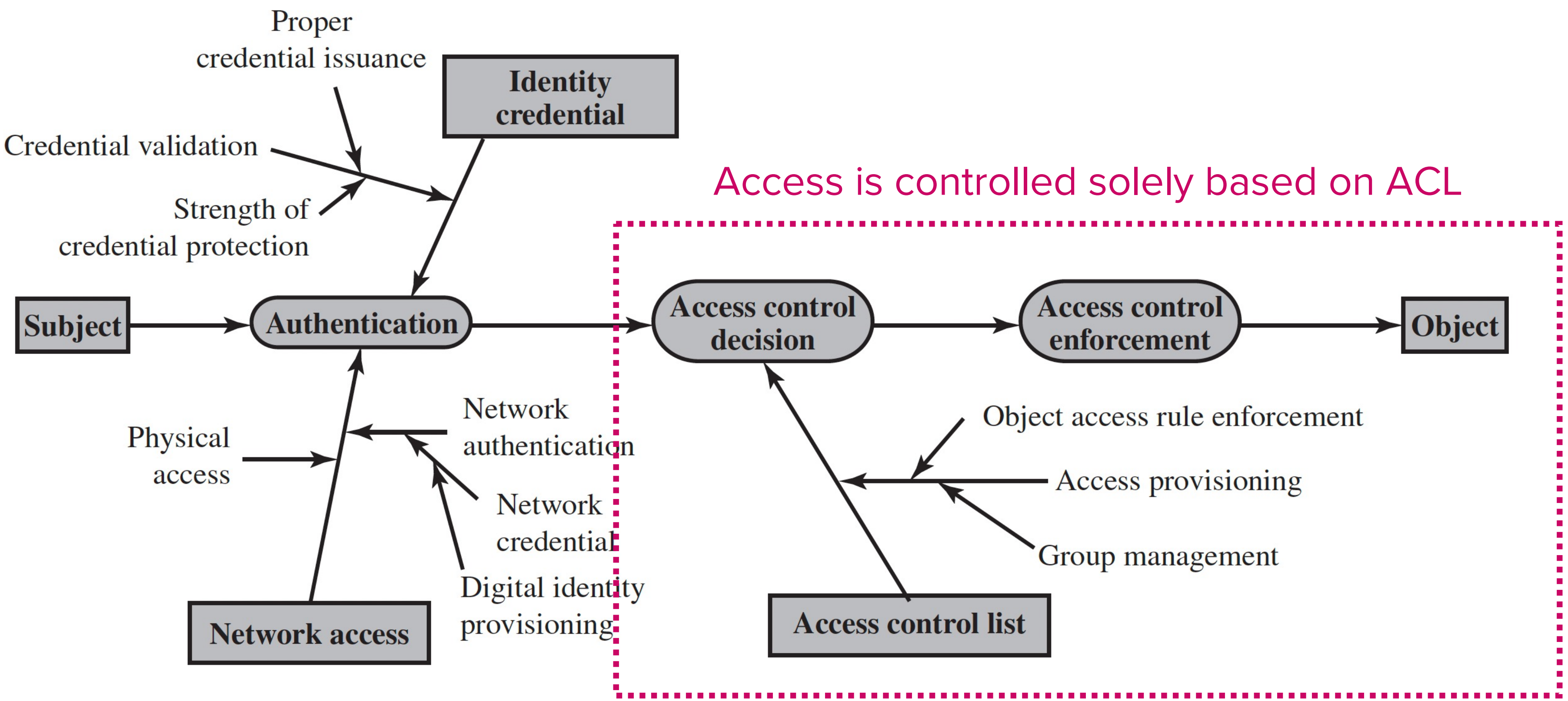
A) No. Copied objects inherit the original role-based permissions.
User 3 (role: Student) cannot access the copy.

Attribute-based Access Control (ABAC)

- ABAC is a generalization of MAC
  - Three key elements
    - Attributes: Defined (or naturally determined) for entities
    - Policy: Defines access policies for attributes
    - Architecture model: Defines the relationship between the policies
  - Flexible and expressive!
    - Attributes are dynamic → We will see an example

Attribute-based Access Control (ABAC)

- ABAC is a generalization of MAC

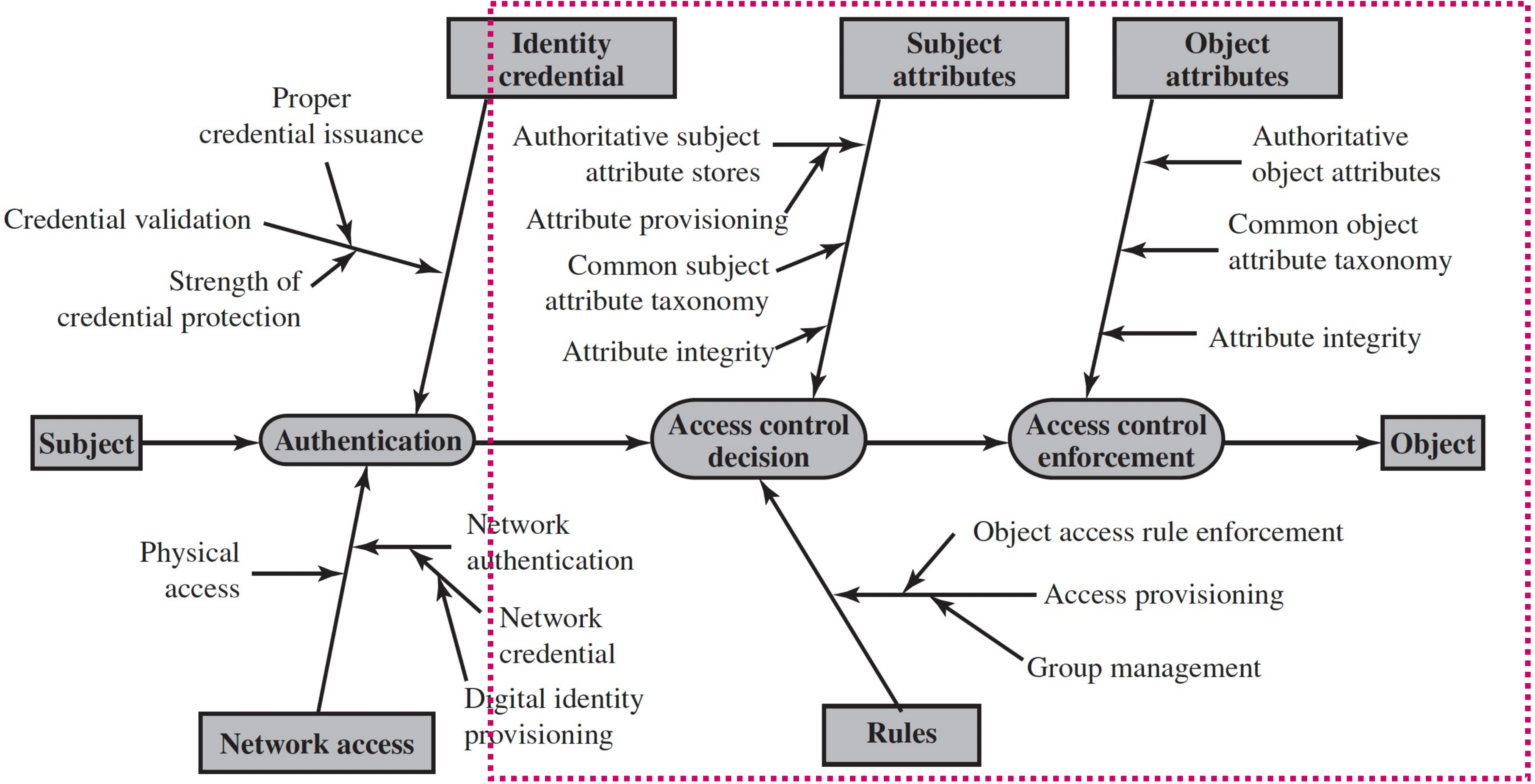


ACL trust chain for discretionary access control (DAC)

Attribute-based Access Control (ABAC)

- ABAC is a generalization of MAC

ACL + attributes + policy



ABAC trust chain for mandatory access control (MAC)

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., a child turns 17)

Attribute-based Access Control (ABAC)

- RBAC vs ABAC Example: Movie Rating
    - 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}
```

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

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

Do not need to redefine/manage static roles

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

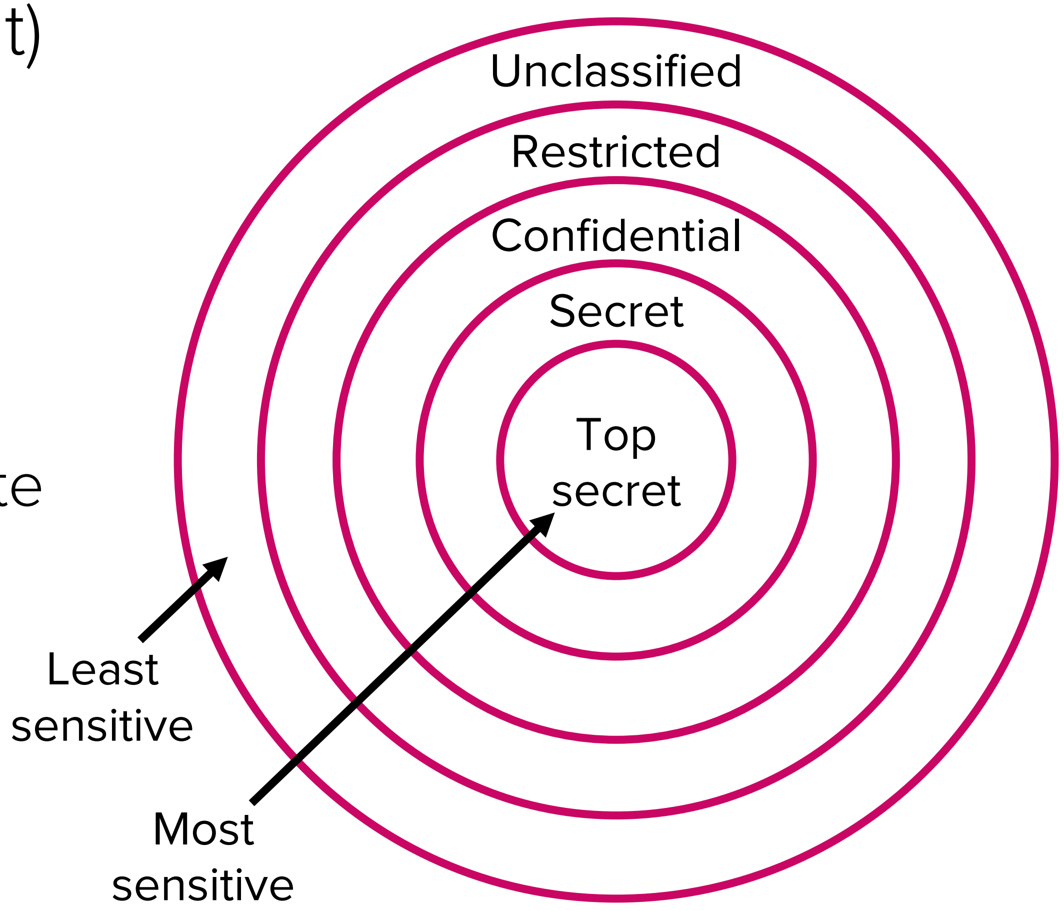
MAC in sensitive environments

- Attributes can be “labels”
- Label:
  - Metadata that describes the nature of resource
  - 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
- Compare security label with security clearance for access control
  - Label indicates how sensitive a resource is
  - Clearance indicates how eligible a subject is

Example: Military security policy

- A US DoD environment
  - Label = (sensitivity level, compartment)
  - e.g., weapon documents
    - Label 1 = (TS, {nuclear, chemical})
    - Label 2 = (S, {nuclear, missile})

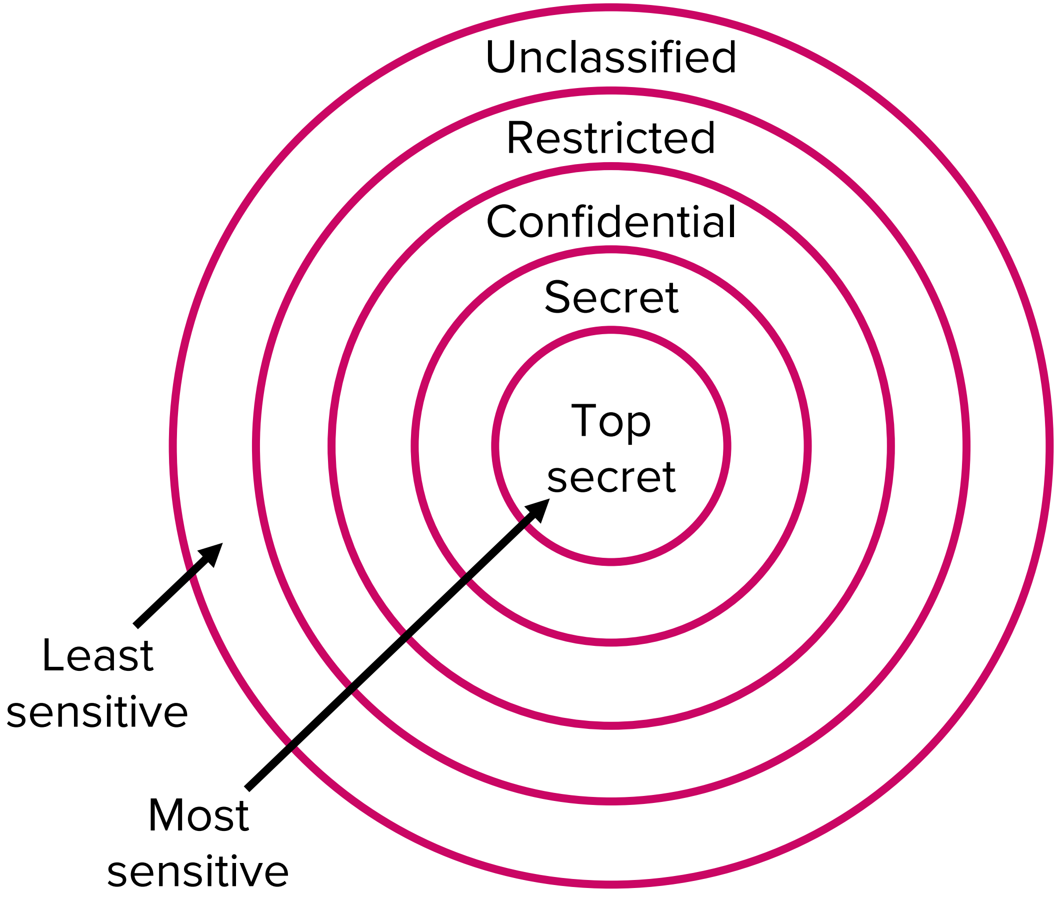
    → Based on these labels,
    users can be authorized to read or write



Comparing labels

- Sensitivity levels are totally ordered
  - i.e., TS > S > C > R > U

- Compartments are sets that are partially ordered
  - nuclear ∈ weapon
  - chemical ∈ 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

- By comparing the labels, we can order them
  - Example:
    - $L_1 = (TS, \{CSE, EE, ME\})$
    - $L_2 = (S, \{CSE, EE\})$
    - $L_3 = (S, \{EE, PHY\})$
    - $L_4 = (C, \{CSE, PHY\})$

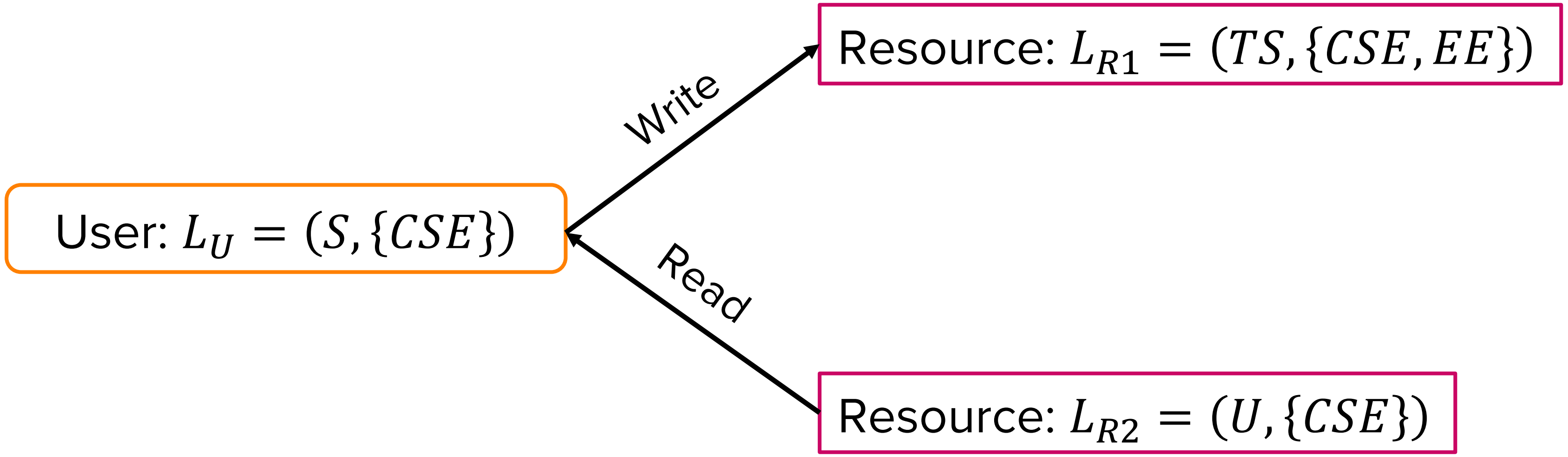
→ Q) Find all dominations?

L_1 dominates L_2

All other pairs of labels are not comparable

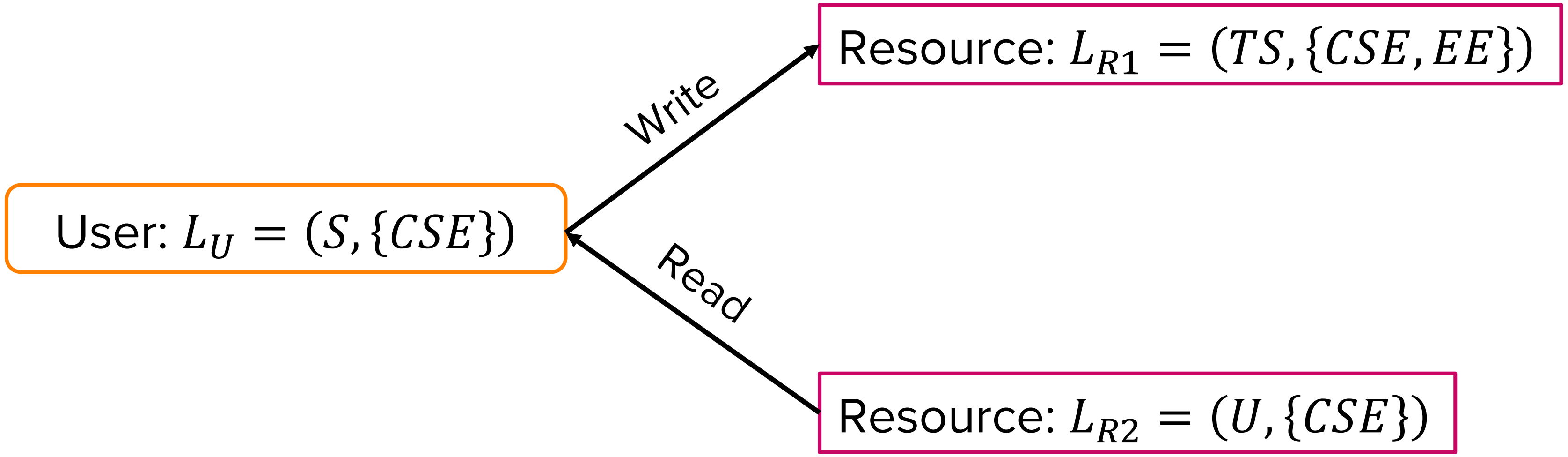
MAC in practice: Bell-LaPadula (BLP) model

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


Resource: $L_{R1} = (TS, \{CSE, EE\})$
Write
User: $L_U = (S, \{CSE\})$
Read
Resource: $L_{R2} = (U, \{CSE\})$

MAC in practice: Bell-LaPadula (BLP) model

- 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


Resource: $L_{R1} = (TS, \{CSE, EE\})$

Write

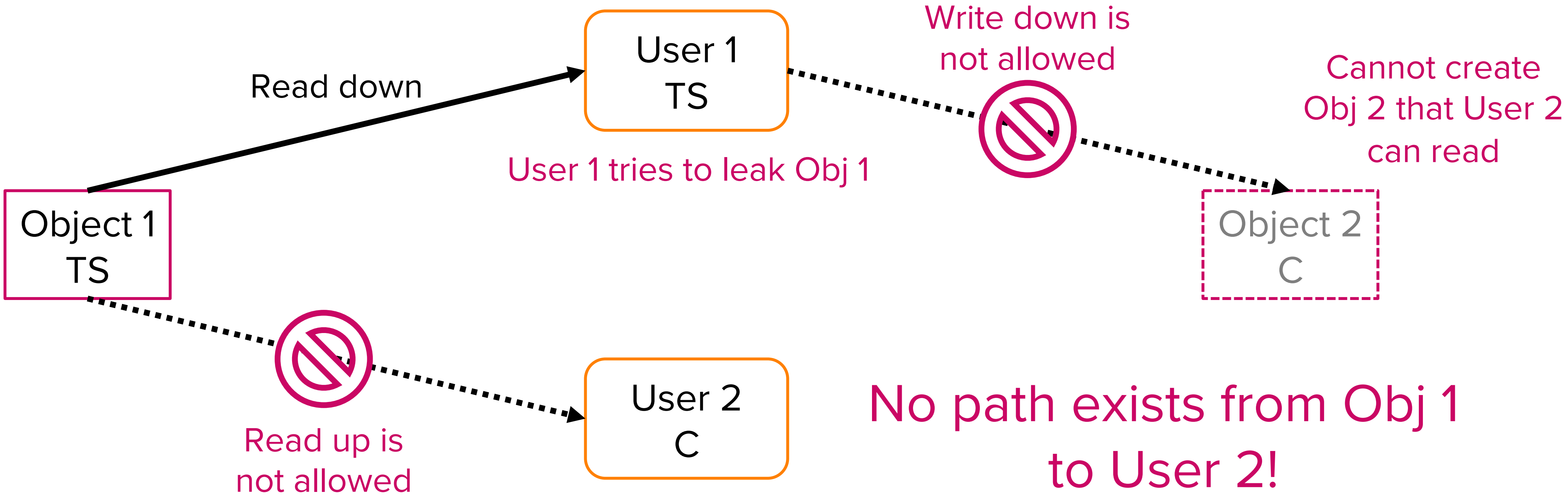
User: $L_U = (S, \{CSE\})$

Read

Resource: $L_{R2} = (U, \{CSE\})$

MAC in practice: Bell-LaPadula (BLP) model

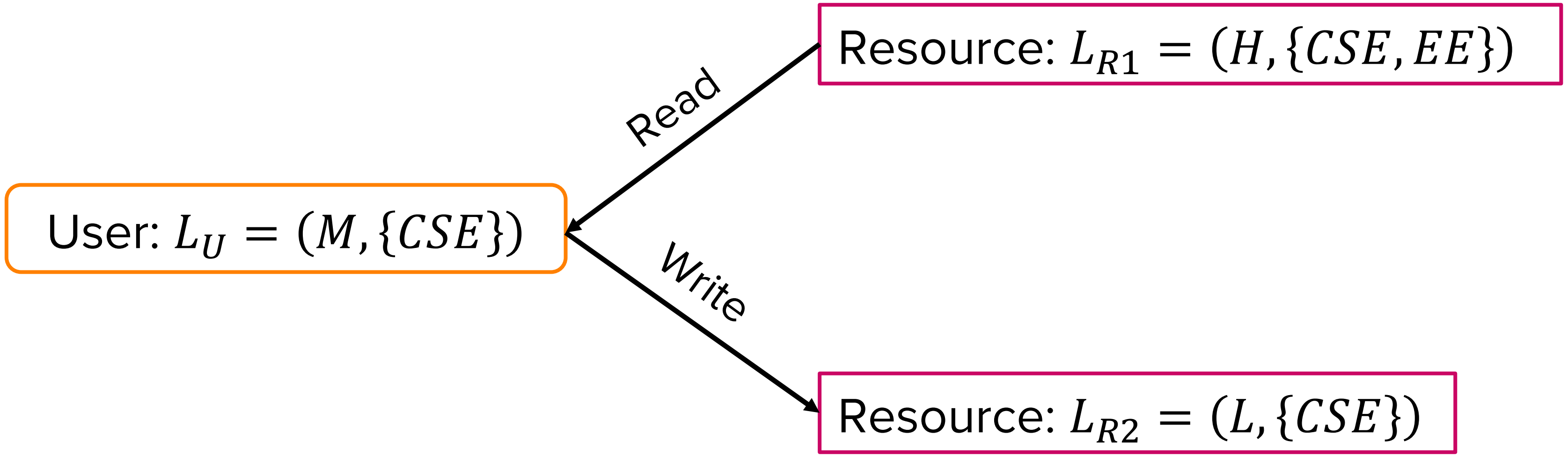
- Solving information flow control problem with BLP
  - Goal: Prevent top secret information in Object 1 to confidential User 2


Write down is not allowed
Cannot create Obj 2 that User 2 can read
Read down
User 1
TS
User 1 tries to leak Obj 1
Object 1
TS
Object 2
C
User 2
C
Read up is not allowed

No path exists from Obj 1 to User 2!

MAC in practice: Biba Integrity model

- Access control model focusing on integrity
  - Opposite of BLP: RUWD rules
    - Integrity level (i.e., quality of information) could be high (trustworthy), medium, or low (untrustworthy)
    - **Read-up rule**: Low integrity users can access high integrity information
    - **Write-down rule**: High integrity users can produce low integrity information



Resource: $L_{R1} = (H, \{CSE, EE\})$
Read
User: $L_U = (M, \{CSE\})$
Write
Resource: $L_{R2} = (L, \{CSE\})$

Models are contradictory

- Bell-LaPadula: Read-down, write-up for confidentiality
- Biba Integrity: Read-up, write-down for integrity
- What if we want both confidentiality and integrity?
    - The only way is allowing read and write at the same level / clearance

MAC in Commercial Context

Models for commercial environments

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

Clark-Wilson model

- Focus heavily 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 to 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 exectued 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

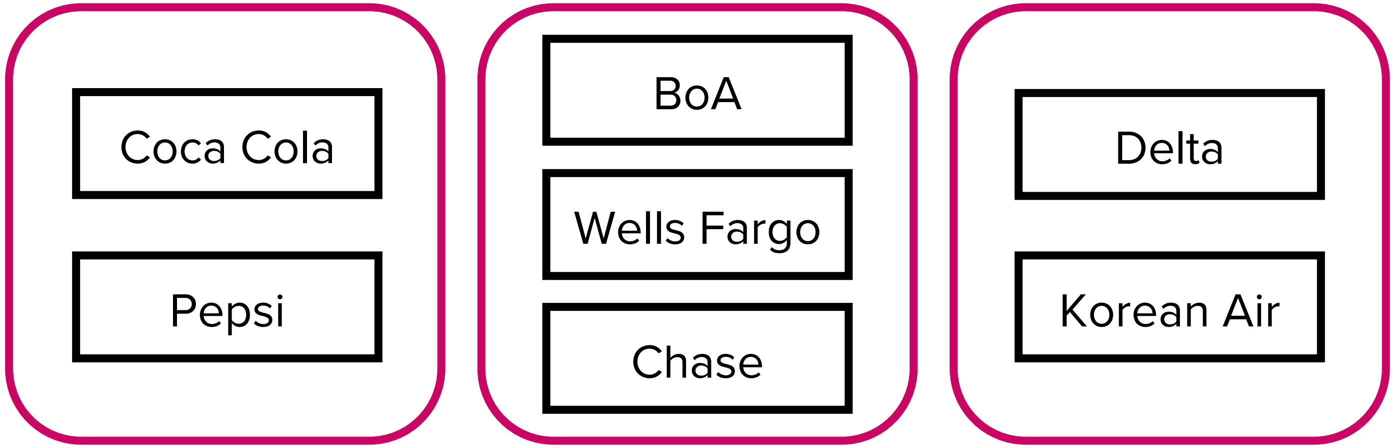
- Example: Placing an order
  - `<User, transaction, {data}>` triples:
    - `<Purchasing agent, place_order, {order book}>`
    - `<Logistics agent, receive_delivery, {inventory}>`
    - `<Finance agent, make_payment, {account balance}>`
  - Separation of duty:
    - Different agents for subparts of “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 (CoI) 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 CoI between competitors

Chinese Wall policy

- Focus on confidentiality
  - Conflicting groups



Coca Cola

Pepsi

BoA

Wells Fargo

Chase

Delta

Korean Air

  - Policy: A user U can access an object O belonging 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 (ref: Lec 19)
    - Ownership (user, group, and other) with access rights
    - 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 DACs still applies (owner, group, others, rwx, …)
  - SELinux then evaluates accesses against its own security policies
    - Additional access control layer

- Policy
  - Targeted (default): Only targeted processes are protected
  - MLS (Multi-level security): BLP model is applied

How does SELinux work?

- Labeling and Type enforcement
  - Labeling
    - Files, processes, ports, ets., are labeled with an SELinux context
    - Labels are stored as extended attributes on the filesystem
    - Labels are in format “user:role:type:level”
      - User: Individual users
      - Role: A group of specific users
      - Type: Abstract domain assigned to subjects and objects
      - Level: Sensitivity level

How does SELinux work?

- Labeling and Type enforcement
  - Type enforcement
    - Each subject is assigned a type (domain)
    - Only certain operations are permitted for each type
    - e.g., Apache’s HTTPD web server
      - 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
```

      - Web server configuration directory is of type httpd_config_t

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

How does SELinux work?

- Labeling and Type enforcement
  - Type enforcement
    - e.g., Apache’s HTTPD web server
      - 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
```

      - Web server configuration directory is of type httpd_config_t

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

      - Access control policy:
        - `allow httpd_exec_t httpd_config_t: file { read }`

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

- Mandatory access control applies global control policy for subjects and resources
- Deals with information flow control problem
- Bell-LaPadula and Chinese Wall policy aim to provide confidentiality
- Biba and Clark-Wilson model aim to provide integrity

Questions?