Bell-LaPadula Model

Dr. Asem Kitana

Mandatory Access Control (MAC)

Mandatory Access Control

- MAC is a type of access control systems that constrains the ability of a subject to access or generally perform some sort of operation on an object.
- In MAC, security policies are centrally controlled by a security administrator; users do not have the ability to override the policy and, for example, grant access to files that would otherwise be restricted.
- Based on Multi-Level Security (MLS)
- Bell-LaPadula model formally defines MLS and MAC.

Mandatory Access Control

Implementations of MAC:

- SELinux: Linux kernel modules available to most Linux distributions (RedHat, Debian, Ubuntu, and SuSE).
- Mandatory Integrity Control: Vista, Windows 7, and Windows 8.

Bell-LaPadula Model

- Bell-LaPadula (BLP) is a confidentiality model, thatsupports confidentiality policies.
- A **confidentiality policy**, also called an *information flow policy*, prevents the unauthorized disclosure of information (i.e. provides secrecy of information).
- Based on **Multi-Level Security** (MLS).
- MLS is the application of a computer system to process information with different security levels (i.e. security classifications), permit access by users with different security clearances.

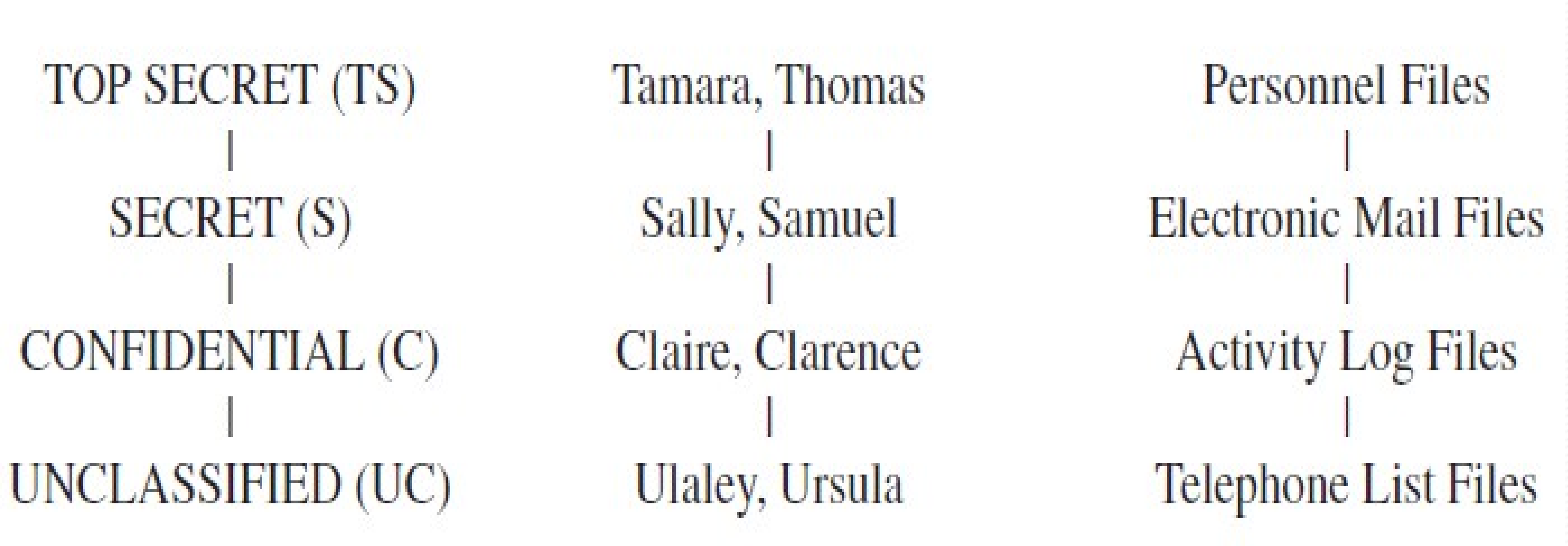
top secret > secret > confidential > restricted > unclassified

- A **security clearance** is a status granted to individuals allowing them access to classified information.

Bell-LaPadula Model

- Subject has ***security clearance*** of a given level
- Object has ***security classification*** of a given level
- The clearances represent sensitivity levels. The higher the security clearance, the more sensitive the information (and the greater the need to keep it confidential).
- Clearance and classification is determine by security administrator; users cannot override security policy.

BLP Model



TOP SECRET (TS)	Tamara, Thomas	Personnel Files
SECRET (S)	Sally, Samuel	Electronic Mail Files
CONFIDENTIAL (C)	Claire, Clarence	Activity Log Files
UNCLASSIFIED (UC)	Ulaley, Ursula	Telephone List Files

At the left is the basic confidentiality classification system. The four security levels are arranged with the most sensitive at the top and the least sensitive at the bottom (linear order). In the middle are individuals grouped by their security clearances, and at the right is a set of documents grouped by their security levels.

BLP Model

- In the previous figure, Claire's security clearance is ***C*** (for CONFIDENTIAL), and Thomas' is ***TS*** (for TOP SECRET). An object has a *security classification*; the security classification of the electronic mail files is ***S*** (for SECRET), and that of the telephone list files is ***UC*** (for UNCLASSIFIED).

BLP Model

- Security levels arranged in linearordering
  - Top Secret: *highest*
  - Secret
  - Confidential
  - Unclassified: *lowest*
- Levels consist of *security clearance L*(*s*)
  - Objects have *security classification L*(o)

Example

Security level	***Subject***	***Object***
Top Secret	Tamara	Personnel Files
Secret	Samuel	E-Mail Files
Confidential	Claire	Activity Logs
Unclassified	James	Telephone Lists

- Tamara can read all files
- Claire cannot read Personnel or E-Mail Files
- James can only read Telephone Lists

Notation

- L(S)=l_s security clearance of subject S
- L(O)=l_o security classification of object O
- For all classification $l_i = 0, \ldots, k\text{-}1, l_i < l_{i+1}$

Properties of BLP Model

The two main properties of BLP are:

- **No read up** Subject can only read an object of less or equal security level, a.k.a. **simple security property (ss-property)**.
- **No write down** Subject can only write into object of greater or equal security level, a.k.a. ***- property (star property).**

Simple Security Property

- Reading information
- Information flows down, not up
- “Reads up” disallowed, “reads down” allowed
- Subject ***s*** can read object ***o*** iff $L(o) \leq L(s)$ and ***s*** has permission to read ***o***

*- property

- Writing Information
- Information flows up, not down
- "Writes up" allowed, "writes down" disallowed
- Subject ***s*** can write object ***o*** iff $L(s) \le L(o)$ and ***s*** has permission to write ***o***

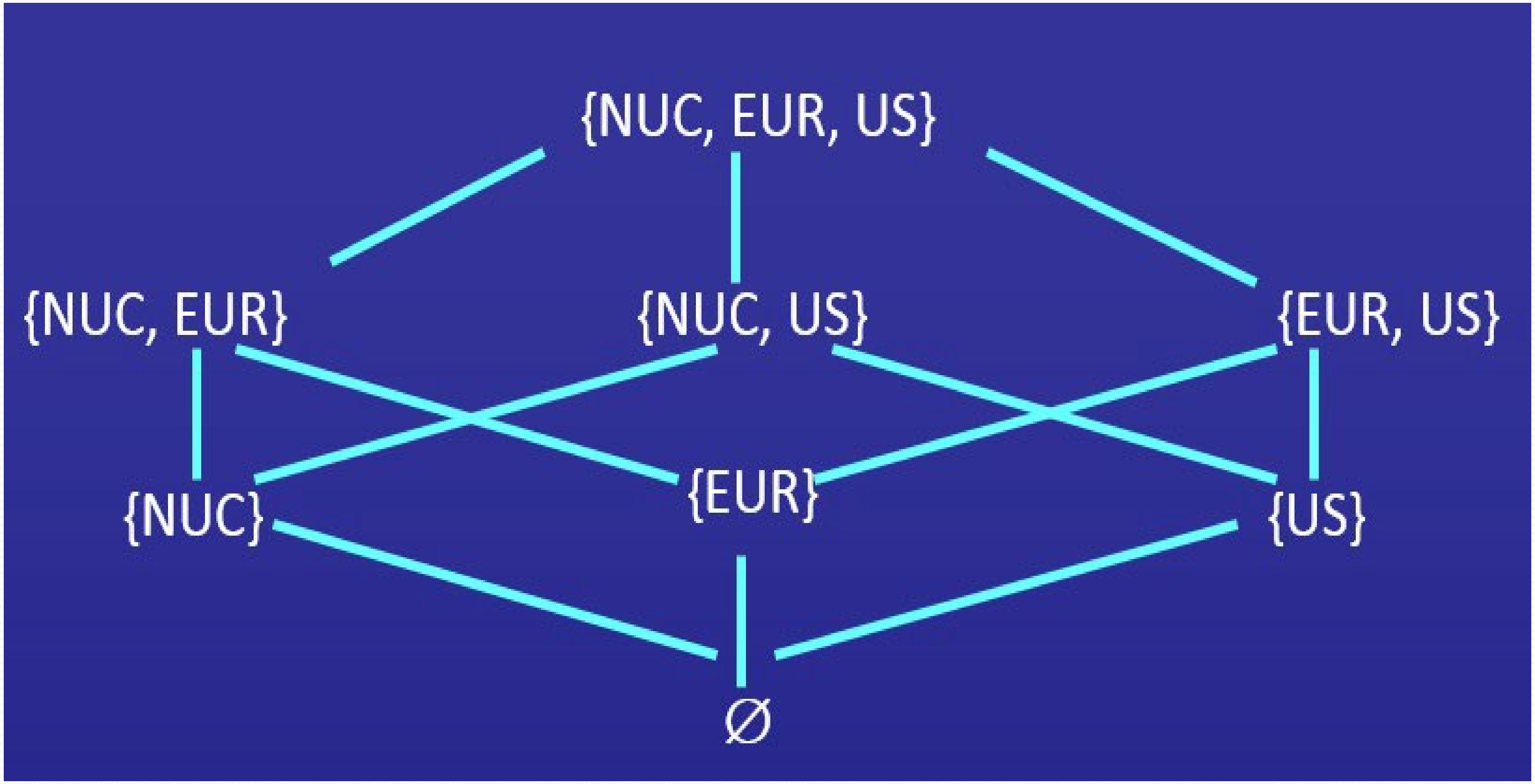
Basic Security Theorem

- Defines a *secure system*
- If a system is initially in a secure state, and every transition of the system satisfies the simple security property and the *-property, then every state of the system is secure.

Set of *Categories*

- The BLP model is expanded by adding a set of categories to each securityclassification.
- Each category describes a kind ofinformation.
- Objects placed in multiple categories have the kinds of information in all of thosecategories.
- Expand notion of security level to includecategories
- Security level is (*clearance*, *category set*)
- Examples
  - (Top Secret, {NUC, EUR, ASI})
  - (Confidential, {EUR, ASI)
  - (Secret, {NUC, ASI })

The Lattice Hierarchy



- **If the categories are NUC, EUR, and US, someone can have access to any of the following sets of categories.**
- **Lattice generated by the categories NUC, EUR, and US.**
- **The lines represent the ordering relation induced by ⊆ (subset of).**

Dominate Relationship (dom)

- Captures the combination of *security classification* and *category set.*
- *Security level = classification level + category set*
- The dominate relationship states that the security level (L, C) *dominates* the security level (L', C') if and only if $L' \leq L$ and $C' \subseteq C$.
- (L, C) *dom* (L', C') iff $L' \leq L$ and $C' \subseteq C$
- We write (L, C) $\neg$*dom* (L', C') when (L, C) *dom* (L', C') is false

Dom Relationship

- Examples
  - (Top Secret, {NUC, ASI}) *dom* (Secret, {NUC})
  - (Secret, {NUC, EUR}) *dom* (Confidential,{NUC, EUR})
  - (Top Secret, {NUC}) ¬*dom* (Confidential, {EUR})

An Example of dom Relationship

Alice is cleared into security level (SECRET, { NUC, EUR}), FileA is classified as (CONFIDENTIAL, { NUC }), FileB is classified as (SECRET, { EUR, US}), and FileC is classified as (SECRET, { EUR }). Then:

- Does Alice dominate FileA ?
- Does Alice dominate FileB ?
- Does Alice dominate FileC ?

An Example of dom Relationship

- Alice is cleared into security level (S, {NUC, EUR})
- `FileA` is classified as (C, {NUC})
- `FileB` is classified as (S, {EUR, US})
- `FileC` is classified as (S, {EUR})

➢ Alice dom `FileA`

Alice *dom* FileA as CONFIDENTIAL ≤ SECRET and { NUC } ⊆ { NUC, EUR }

➢ Alice ¬dom `FileB`

Alice ¬ *dom* FileB as { EUR, US } ⊄ { NUC, EUR }

➢ Alice dom `FileC`

Alice *dom* FileC as SECRET ≤ SECRET and { EUR } ⊆{ NUC, EUR }

Extended Simple Security Property

- Reads up" disallowed, "reads down" allowed
- Let *C*(*S*) be the category set of subject *S*, and let *C*(*O*) be the category set of object *O*. Then extended simple security property states:
- *Subject S* can read Object *O* if and only if *S dom O* and *S* has discretionary read access to *O*.
  - Subject *s* can read object *o* iff *L*(*s*) *dom L*(*o*) and *s* has permission to read *o*
  - Note: combines mandatory access control (relationship of security levels) and discretionary access control (the required permission)

Example of extended ss property

- In the previous example, Alice can read FileA and FileC but not FileB (assuming that thediscretionary access controls allow suchaccess).

Extended *- Property

- “Writes up” allowed, “writes down” disallowed.
- *S* can write to *O* if and only if *O dom S* and *S* has discretionary write access to *O*.
  - Subject *s* can write object *o* iff *L*(*o*) *dom L*(*s*) and *s* has permission to write *o*
  - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)

BLP Example:

- Suppose Paul is cleared into security level (SECRET, { EUR, US, NUC }) and has discretionary read access and write access to the files. Also, FileA is classified as (CONFIDENTIAL, { NUC }), FileB is classified as (SECRET, { EUR, US}), and FileC is classified as (SECRET, { EUR }). Then, what files Paul can read and what files can write?

Example of extended *- property

- Suppose Paul is cleared into security level (SECRET, { EUR, US, NUC }) and has discretionary read accessto FileB. Then, Paul can read FileB (because Paul dom FileB is true), but he cannot write to FileA, because FileA *dom* Paul is false.

<u>Note:</u>

- `FileA` is classified as (C, {NUC})
- `FileB` is classified as (S, {EUR, US})
- `FileC` is classified as (S, {EUR})

Problem

- Sometimes, a subject must communicate with another subject at a lower level. This requires the higher-level subject to write into a lower-level object that the lower level subject can read.
- EXAMPLE:

A manager with (SECRET, { NUC, EUR }) clearance needs to send a message to an assistant with (SECRET, { EUR }) clearance. The manager must write a document that has at most the (SECRET, { EUR }) classification. But this violates the *-property, because (SECRET, { NUC, EUR }) *dom* (SECRET, { EUR }).

Problem

- Manager has (Secret, {Nuc, Eur}) clearance
- Assistant has (Secret, {Eur}) clearance
    - Assistant can talk to Manager (“write up” or “read down”)
    - Manager cannot talk to Assistant (“read up” or “write down”)

Solution

- The model provides a mechanism for allowing this type of communication. A subject has a *maximum security level* and a *current security level*. The maximum security level must dominate the current security level. Asubject may (effectively) decrease its security level from the maximum in order to communicate with entities at lower security levels.

Solution

- Define maximum, and current security levelsfor subjects
  - *maxlevel*(*s*) *dom curlevel*(*s*)
- Example
  - Treat Assistant as an object (Manager is writing to him/her)
  - Manager has *maxlevel* (Secret, {Nuc, Eur})
  - Manager sets *curlevel* to (Secret, { Eur })
  - Now *L*(Assistant) *dom curlevel*(Manager)
    - Managercan write to Assistant without violating “no writes down”