isec
INFORMATION SECURITY
lectures by Ferucio Laurentiu Tiplea

Access Control

Mandatory Access Control

Prof.dr. Ferucio Laurențiu Țiplea

Fall 2023

Department of Computer Science
"Alexandru Ioan Cuza" University of Iași
Iași 700506, Romania

e-mail: ferucio.tiplea@uaic.ro

Outline

Introduction to MAC

Mandatory access control

Basic features:

- MAC enforces access control based on regulations mandated by a central authority;
- No concept of ownership in MAC;
- MAC makes distinction between users and subjects:
  - Users are trusted (must be trusted) not to disclose secret information outside of the computer system;
  - Subjects are not trusted (they may have Trojan horses embedded into the code they execute).

MAC models:

- The Bell-LaPadula model (confidentiality);
- The Biba model (integrity);
- The Chinese wall model (separation of duty).

Information flow models

Information flow models

Information flow (IF) models were introduced by Denning (1976).

Basic features:

- IF models are concerned with the flow of information from one security class to another;
- Object = viewed as a container of information;
- Examples of objects: files or directories in an operating system, or relations and tuples in a database;
- Information flow is controlled by assigning every object a security class or security label.

Information flow models: definition

Definition 1

An information flow model is a triple $(SC, \rightarrow, \oplus)$, where:

- SC is a set of elements called security classes (or access classes or security labels);
- $\rightarrow \subseteq SC \times SC$ is a binary relation called may-flow;
- $\oplus : SC \times SC \rightarrow SC$ is a commutative and associative operator called the class combiner operator.

Meaning:

- The semantics of a security class is varied and depends on the purpose of use (e.g., for confidentiality, integrity, categories or compartments of objects and subjects, etc.);
- $A \rightarrow B$: the information may flow from A to B;
- $A \oplus B$: if information from A and B are combined, the result belongs to $A \oplus B$.

Information flow models: Denning's axioms

Denning's axioms:

Axiom 1: SC is finite;

Axiom 2: The may-flow relation $\rightarrow$ is a partial order;

Axiom 3: SC has a least element w.r.t. $\rightarrow$;

Axiom 4: $\oplus$ is a least upper bound operator.

Proposition 2

Any information flow model that satisfies the Denning's axioms is a lattice.

In what follows, all IF models we consider satisfy the Denning's axioms!

Information flow models: dominance

Definition 3

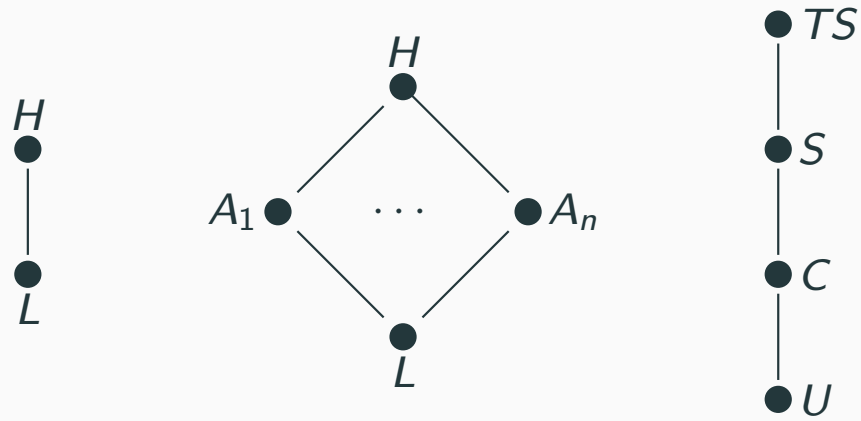
Let $(SC, \rightarrow, \oplus)$ be an information flow model and $A, B \in SC$. We say that A dominates B, denoted $A \geq B$, if $B \rightarrow A$.

Notation and terminology:

- $A > B$ (A strictly dominates B) if A dominates B and $A \neq B$;
- A and B are comparable if $A \geq B$ or $B \geq A$;
- A and B are incomparable if A and B are not comparable.

Information flow models: examples

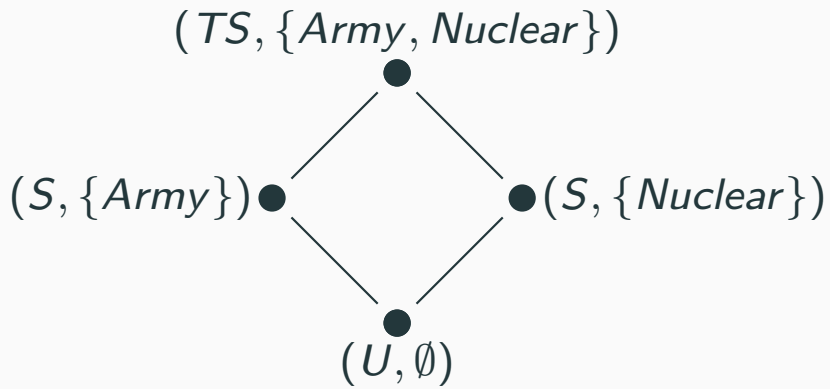
A security class may be precisely a tag/label that specifies the sensitivity degree once attached to an object or subject.



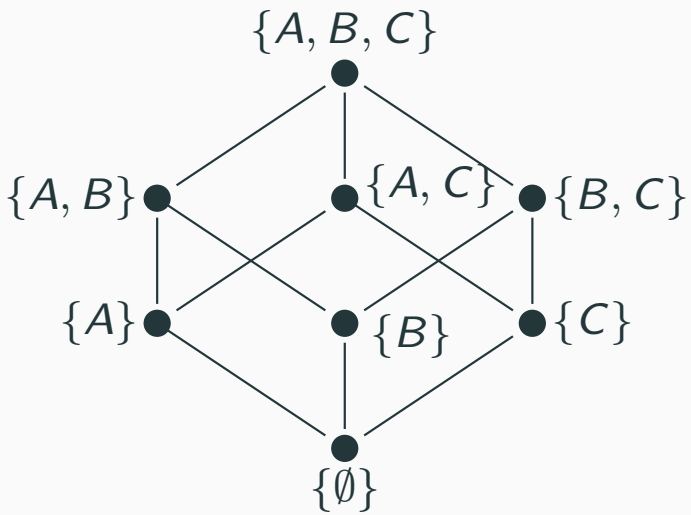
For example, if an object has the L label attached (see the first two lattices), it is considered an object with low sensitivity.

Information flow models: examples

A security class may also specify a security label and a set of categories (of objects or subjects). The security label then specifies the degree of sensitivity of all elements of one of the categories thus specified.



$(S, \{Army\})$ specifies that all items in the *Army* category have a degree of sensitivity S, as do items in the *Nuclear* category. However, $(S, \{Army\})$ and $(S, \{Nuclear\})$ are different security classes.



This lattice does not specify the degrees of sensitivity of its security classes, these being irrelevant as long as we have their lattice structure.

Confidentiality-based mandatory policies: the Bell-LaPadula model

Confidentiality-based mandatory policies

Overview :

- Aim: control the direct and indirect flows of information by preventing leakages to unauthorized subjects;
- Subjects and objects are assigned security levels (security classes);
- The security level of an object, also called security classification, reflects the sensitivity of the information contained in the object;
- The security level of a subject, also called security clearance, reflects the user's trustworthiness;
- Requests of subjects to access objects are regulated by means of their security classes.

Each user with a security class assigned to it can connect to the system (as a principal) at any security class dominated by its security class.

The Bell-LaPadula model: a minimalist approach

Proposed by Bell and LaPadula (1973) to formalize the multilevel security policy of the U.S. Department of Defense.

Overview of the Bell-LaPadula (BLP) model:

- Key idea: augment DAC with MAC to enforce information flow policies;
- Two-step approach:
  1. First, a discretionary access control matrix D is established;
  2. Second, operations must be authorized by the mandatory access control policy;
- The model was initially defined for a fixed set of rights $R = \{r, w\}$.

The Bell-LaPadula model: a minimalist approach

The MAC in the BLP model:

- Assign labels to subjects and objects by some labelling function λ;
- Rules (No Read Up – No Write Down):
  1. Simple security (ss-) property – $\boxed{s \text{ is allowed to read } o \text{ only if } \lambda(s) \geq \lambda(o)}$
  2. $*$-property – $\boxed{s \text{ is allowed to write } o \text{ only if } \lambda(s) \leq \lambda(o)}$

Initially, the BLP model assumed that the labels of subjects and objects, once assigned, could not be changed (unless a security administrator resets them). This property has been called the tranquility principle. However, it can be relaxed without losing security (Sandhu (1993)).

On the $*$-property

Remark 4

The $*$-property avoids Trojan horse attacks because subjects cannot transfer information to a security class with lower sensitivity than their security clearance.

Through the simple security property, any subject can transmit information (e.g., emails) to a higher (dominant) security class.

Users are not restricted by the $*$-property, as they can connect to the system (as principals) at any security class dominated by their security clearance.

Remark 5

The $*$-property allows secret data be destroyed or damaged by unclassified subjects. To prevent this the $*$-property is sometimes used in the form

Strong $*$-property – $\boxed{s \text{ is allowed to write } o \text{ only if } \lambda(s) = \lambda(o)}$

The Bell-LaPadula model: remarks

- In some approaches, write access means "read and write", with append access provided for "write only";
- The BLP model is stated in terms of read and write operations (which suffices to illustrate the main points). Other operations may be added, such as create and destroy objects, constrained by the $*$-property because they modify the state of the object in question;
- Mandatory controls in BLP are coupled with discretionary control: if the access control matrix does not authorizes the operation, there is no need to check the mandatory controls since the operation will be rejected anyway;
- A user can login (create a subject) with any label dominated by the user's clearance.

Integrity-based mandatory policies: the Biba model

Integrity-based mandatory policies

Overview :

- Aim: control the flows of information and prevent subjects to indirectly modify information they cannot write;
- Subjects and objects are assigned integrity levels (integrity classes);
- The integrity level of an object reflects both the degree of trust of the information stored in the object and the potential damage resulting from unauthorized modifications of the information;
- The integrity level of a subject reflects the user's trustworthiness for inserting, modifying, or deleting information;
- Requests of subjects to access objects are regulated by means of their integrity classes.

The Biba model

Proposed by Biba (1977).

The MAC in the Biba model:

- associate labels to subjects and objects by some function ω;
- Rules (No Read Down – No Write Up):
  1. Simple integrity (si-) property – $\boxed{s \text{ is allowed to read } o \text{ only if } \omega(s) \leq \omega(o)}$
  2. Integrity $*$-property – $\boxed{s \text{ is allowed to write } o \text{ only if } \omega(s) \geq \omega(o)}$

Remark 6

The Biba model's rules are the dual of the BLP model's rules.

Combining the BLP and Biba models

Combining the BLP and Biba Models

- There is no fundamental difference between the BLP and Biba models: both are concerned with information flow in a lattice of security classes;
- In the BLP model, the information flows upward;
- In the Biba model, the information flows downward;
- The direction is irrelevant: it is a matter of convention in representing the highest security class (in our case, in both the BLP and Biba models the highest security class on top of the lattice).

Case 1: single label

Combination 1 : use a single label for both confidentiality and integrity.

Conclusions :

- s can read or write o only if s and o have the same security class!
- No information flow between security classes!
- Irrelevant model.

Case 2: independent labels, same directions

Combination 2 : use independent labels for confidentiality (λ) and integrity (ω) under the assumption that both lattices have the highest security class on top.

Conclusions :

- Rules:
  1. s is allowed to read o only if $\lambda(s) \geq \lambda(o)$ and $\omega(s) \leq \omega(o)$
  2. s is allowed to write o only if $\lambda(s) \leq \lambda(o)$ and $\omega(s) \geq \omega(o)$
- The model uses two lattices with information flow going in opposite directions;
- Implemented in several operating system, database, and network products.

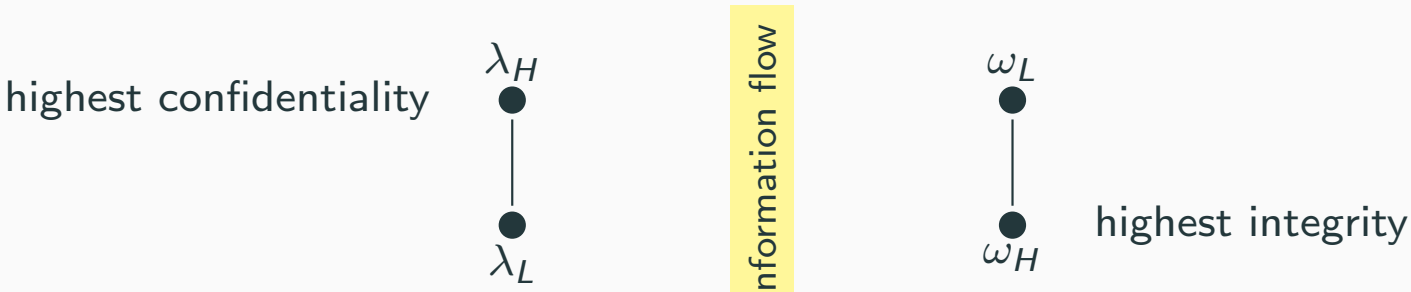
Case 3: independent labels, opposite directions

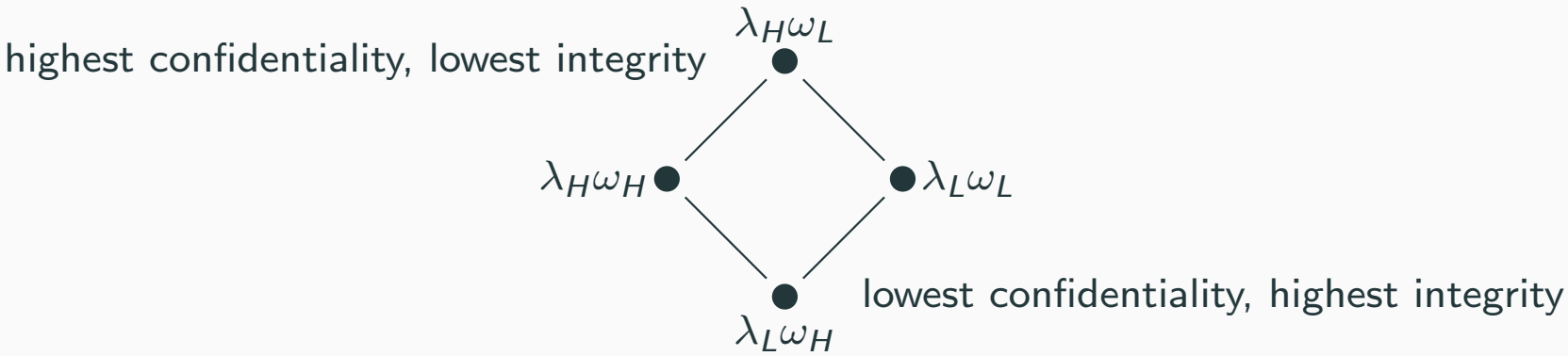
Combination 3 : use independent labels for confidentiality (λ) and integrity (ω) under the assumption that the lattices have the highest security classes on opposite directions.

Conclusions :

- Rules:
  1. s is allowed to read o only if $\lambda(s) \geq \lambda(o)$ and $\omega(s) \geq \omega(o)$
  2. s is allowed to write o only if $\lambda(s) \leq \lambda(o)$ and $\omega(s) \leq \omega(o)$
- The two lattices can be combined in just one lattice (see next slide);
- In this lattice, the entity with highest confidentiality has lowest integrity, and vice versa.

Case 3: example





Separation of duty: the Chinese wall model

Conflict of interest

The Chinese wall model was proposed by Brewer and Nash (1989).

Where it arises :

- In the commercial sector that provides consulting services to other companies.

Aim :

- Prevent information flows that result in a conflict of interest and inadvertent disclosure of information by a consultant or contractor;
- Example of conflict of interest: lawyer providing consultancy services for two airline companies.

How :

- Combines commercial discretion with legally enforceable mandatory controls.

Conflict of interest and the Chinese wall

Basic elements:

- Object = item of information concerning a single company;
- Company dataset = all objects which concern the same company;
- Conflict of interest class (CIC) = all datasets of the companies that are in competition;
- Subject = user or program that might act on behalf of a user.

Basic idea:

- In the first instance, each subject has complete freedom to access anything he cares;
- Once an object in a dataset D of some CIC is chosen, a Chinese Wall is created around D and no other dataset in CIC can be chosen by the same subject.

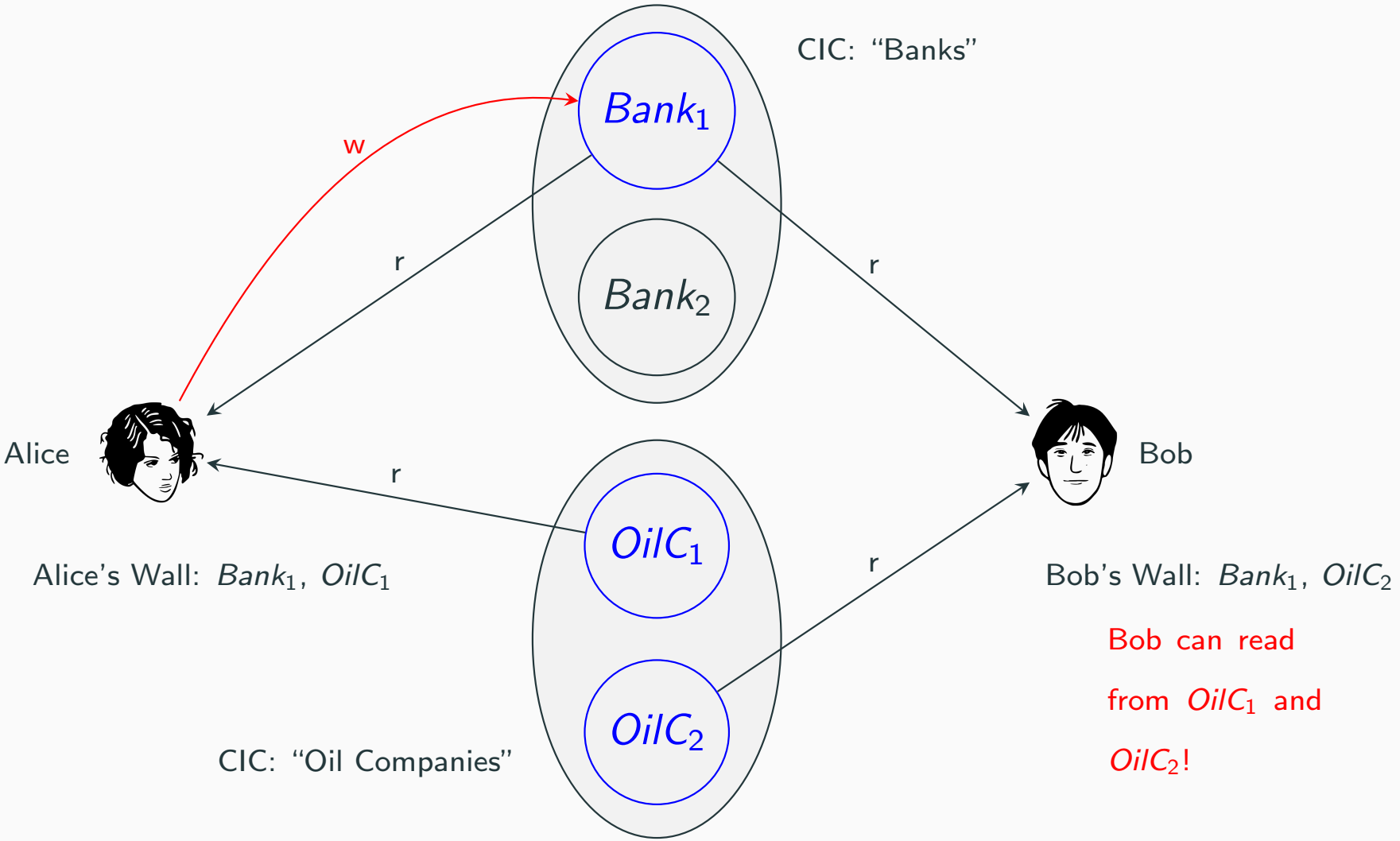
The Chinese wall model

Rules:

- (Chinese Wall) Simple Security Rule: a subject s can be granted read access to an object o only if the object:
  1. is in the same company datasets as the objects already accessed by s, that is, "within the Wall", or
  2. belongs to an entirely different conflict of interest class.
- (Chinese Wall) $*$-property: a subject s can be granted write access to an object o only if:
  1. s can read o by the simple security rule, and
  2. no object can be read which is in a different company dataset to the one for which write access is requested

$*$-property implies that either s cannot write at all or s is limited to reading from and writing to just one company dataset!

Reason for the $*$-property



Criticisms of the model

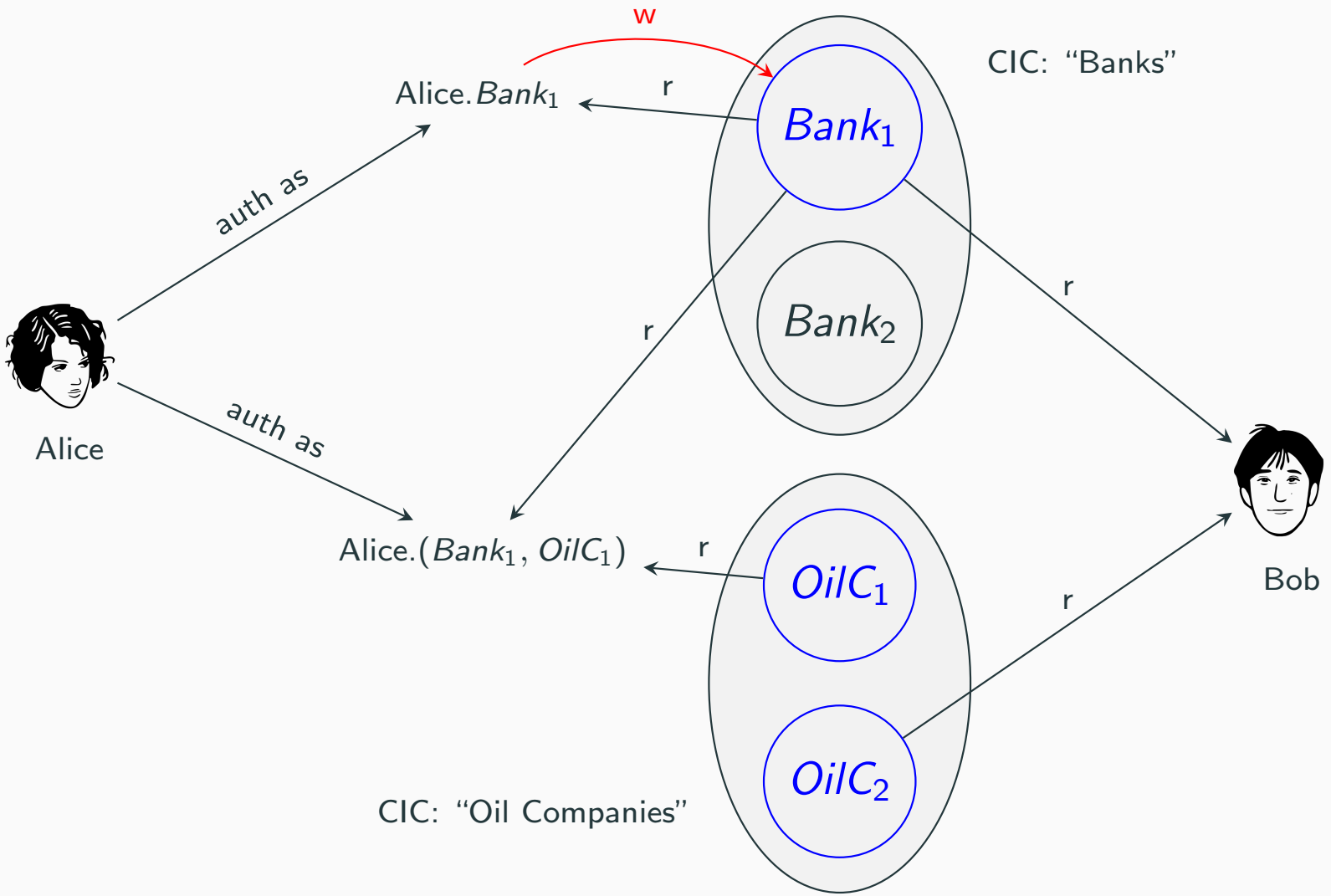
The $*$-property of the Chinese wall model protects against Trojan horse-like attacks. However, the price is unacceptable:

1. A user that reads two company datasets cannot write at all;
2. A user that is allowed to read just one company dataset can write only to that dataset.

A typical activity of a user requires reading and writing operations. The Chinese wall model thus limits the activity of users only to working with a single company. Sandhu (1992) highlighted this.

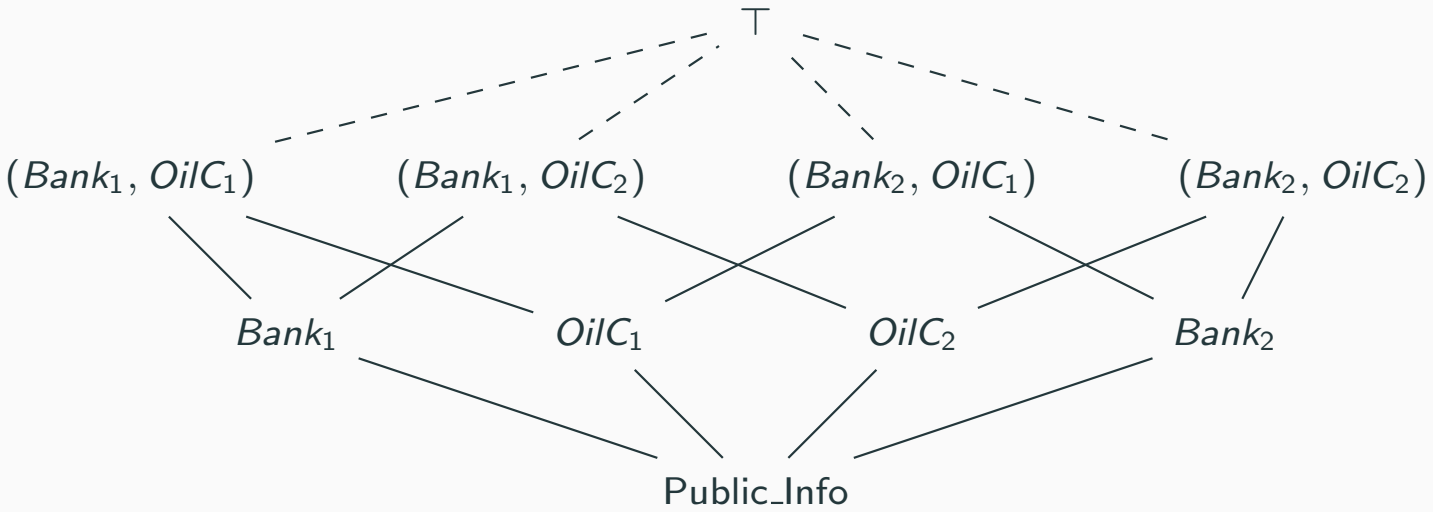
This anomaly occurs because the model does not distinguish between rules applied to users and subjects (see the concept of subject in this model).

A variation of the model: user-principal-subject distinction



A lattice representation: object labeling

The security labels are represented in the lattice bellow (Sandhu (1992)).



Object labeling :

- First level: public information;
- Second level: companies datasets;
- Third level: allowed combinations of companies datasets;
- Top level: combinations not allowed (this level might be empty).

A lattice representation: user, principal, and subject labeling

Labeling rules for users, principals, and subjects:

- Each user U may have associated any security label, except for $\top$;
- Each user U with the security label X has associated the principals $U.Y$, for any Y dominated by X;
- Each principal $U.Y$ has associated the fixed security label Y. Each subject created by the principal inherits this label.

Example 7

If Alice has the security label $Bank_1$, then she has associated two principals, $Alice.Public\_Info$ and $Alice.Bank_1$. So, she can read from $Public\_Info$ and $Bank_1$, and write to $(Bank_1, OilC_1)$ and $(Bank_1, OilC_2)$. The conflict of interest is thus avoided.

This new model behaves as the BLP model!

MAC implementations

MAC early implementations

Early implementations (started out in the eighties, military-oriented):

- Honeywell Secure Communications Processor (SCOMP);
- Strategic Air Command DIgital Network (SACDIN) of the US Air Force (USAF);
- Boeing Multi-level Secure Local Area Network;
- etc.

MAC recent implementations: SELinux

Security-Enhanced Linux (SELinux):

- Linux kernel security module that provides a mechanism for supporting access control security policies, including MAC;
- The key concepts can be found in some earlier projects by United States National Security Agency (NSA) `https://www.nsa.gov/selinux/`.

Subjects and objects:

- Subject security level = domain;
- Object security level = type;
- Type of an object = class.

MAC recent implementations: SELinux

SELinux has two types of rules: access rules and labeling rules:

- Access rules:
  - Example: `allow sshd.t shell.exec.t:file execute`
  - Meaning: when a subject of `sshd.t` accesses an object of `shell.exec.t` of class `file`, it has the `execute` permission;
- Rules for the type of a new object (labeling rules):
  - Example: `type.transition sshd.t tmp.t: devfile.class.set cardmsg.dev.t`
  - Meaning: when sshd daemon creates a device file in the tmp directory, the new file is labeled with `cardmsg.dev.t`.

More on SELinux: Kuliamin et al. (2019); Vermeulen (2020).

MAC recent implementations: AppArmor

Application Armor (AppArmor):

- Linux kernel security module that supplements the traditional Unix DAC model by providing MAC;
- Originally developed by Immunix (1998-2005), then by SUSE (2005-2009), and currently by Canonical from 2009;
- Included in the mainline Linux kernel since version 2.6.36 (Oct 2010);
- A set of MAC rules in AppArmor is known as a profile;
- SELinux labels the files (according to the main theory on MAC), while AppArmor works with file paths;
- Two types of rules:
  - path entries: what files an application can access in the file system;
  - capability entries: privileges that are allowed to use.

MAC recent implementations: MIC

Mandatory Integrity Control in Windows operating systems:

- Starting with Windows Vista and Windows Server 2008, Microsoft adds a Mandatory Integrity Control (MIC) (Microsoft (2021));
- MIC is a form of the Biba model ensuring integrity to writes and deletions: to write to or delete an object, the subject's integrity level must be greater than or equal to the object' integrity level;
- There are six integrity levels: Untrusted, Low (everyone), Medium (standard and authenticated users), High (local or network services, elevated users), System (system services), and Trusted Installer;
- Subjects' integrity level: when a user logs on, Windows Vista assigns an integrity SID to the user's access token;
- Objects' integrity level: files, pipes, threads, registry keys, printers etc., are assigned an integrity SID which is stored in the System Access Control List (SACL).

MAC recent implementations: MAC and MIC in IBM AIX 7.2

IBM AIX 7.2. Security:

"Mandatory Access Control is enforced any time a process attempts to open a file system object, retrieve the attributes of a file system object, send a signal to a process, transfer data through a STREAM, or send or receive a packet through a network interface. Access to any file system object is only possible if both MAC and DAC criteria are met. When a user attempts to access a file, MAC restrictions are enforced before DAC restrictions, such as permission bits or ACLs, are checked."

"Trusted AIX uses a system of labels to enforce MIC. On a Trusted AIX system, all named objects have integrity labels (TLs) to identify the object's integrity level. Processes also have TLs. Process TLs indicate which level of information integrity the process is allowed to access. The higher the TL, the more trustworthy the object or process is."

MAC and covert channels

Covert channels

Definition 8 (From NIST SP 800-53 Rev.5)

A covert channel is an unintended or unauthorized intra-system channel that enables two cooperating entities to transfer information in a way that violates the system's security policy but does not exceed the entities' access authorizations.

- They were signaled by Lampson (1973);
- Trusted Computer System Evaluation Criteria (TCSEC), also called the Orange Book, defines two kinds of covert channels:
  - Storage channels – modify storage location to communicate;
  - Timing channels – use a delay between packets transmitted over a network;
- Covert channels require a cooperating sender and receiver, are hard to detect and control, and can exist in any MAC system. MAC cannot protect against them!

Concluding remarks on MAC models

Concluding remarks

- MAC enforces access control on the basis of regulations mandated by a central authority;
- MAC makes clear distinction between users and subjects;
- Lattice-based access control (LBAC), also called label-based access control or rule-based access control or multilevel access control, is a form of MAC;
- In LBAC, a lattice is used to define levels of security an object or subject may have;
- Mandatory policies protect the flow of information over overt channels (i.e., legitimate channels) but not over covert channels.

Readings

In addition to the materials indicated so far, I recommend:

- Chapters 6 and 7 of Conrad et al. (2016);
- Chapters 3 and 4 of Andress (2014);
- Chapter 11 of Collins (2014);
- Chapter 23 of Bertino (2012);
- Samarati and de Capitani di Vimercati (2001).

References

Andress, J. (2014). *The Basics of Information Security. Understanding the Fundamentals of Infosec in Theory and Practice*. Syngress, Elsevier, Boston, 2nd edition.

Bell, D. E. and LaPadula, L. J. (1973). Secure computer systems: Mathematical foundations. Technical Report 2547, MITRE Corporation.

Bertino, E. (2012). Chapter 23 - Policies, access control, and formal methods. In Das, S. K., Kant, K., and Zhang, N., editors, *Handbook on Securing Cyber-Physical Critical Infrastructure*, pages 573–594. Morgan Kaufmann, Boston.

Biba, K. (1977). Integrity considerations for secure computer systems.

Brewer, D. and Nash, M. (1989). The Chinese wall security policy. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 206–214.

Collins, L. (2014). Chapter 11 - Access controls. In Vacca, J. R., editor, *Cyber Security and IT Infrastructure Protection*, pages 269–280. Syngress, Boston.

Conrad, E., Misenar, S., and Feldman, J. (2016). *CISSP Study Guide*. Singress, Elsevier, 3rd edition.

Denning, D. E. (1976). A lattice model of secure information flow. *Commun. ACM*, 19(5):236–243.

References (cont.)

IBM (2021). IBM AIX 7.2. Security. Technical report, IBM Corp.

Kuliamin, V., Khoroshilov, A., and Medveded, D. (2019). Formal modeling of multi-level security and integrity control implemented with SELinux. In *2019 Actual Problems of Systems and Software Engineering (APSSE)*, pages 131–136.

Lampson, B. W. (1973). A note on the confinement problem. *Commun. ACM*, 16(10):613–615.

Microsoft (2021). Windows security. Technical report, Microsoft.

Samarati, P. and de Capitani di Vimercati, S. (2001). Access control: Policies, models, and mechanisms. In Focardi, R. and Gorrieri, R., editors, *Foundations of Security Analysis and Design*, pages 137–196, Berlin, Heidelberg. Springer Berlin Heidelberg.

Sandhu, R. (1993). Lattice-based access control models. *Computer*, 26(11):9–19.

Sandhu, R. S. (1992). A lattice interpretation of the chinese wall policy. In *Proc. of the 15th NIST-NCSC National Computer Security Conference*, pages 329–339, Baltimore, MD.

Vermeulen, S. (2020). *SELinux System Administration*. Packt Publishing, third edition.