Mandatory Access Control Models

Prof.Dr. Ferucio Laurenţiu Ţiplea

Department of Computer Science
Alexandru Ioan Cuza University of Iaşi
Iaşi, Romania
E-mail: `ferucio.tiplea@uaic.ro`

Outline

Mandatory Access Control

Basic features:

- MAC enforces access control on the basis of regulations mandated by a central authority
- No concept of ownership in MAC
- MAC makes distinction between users and subjects

MAC models:

- The Bell-LaPadula model
- The Biba model
- The Chinese-wall model

Information Flow Models

D. E. Denning. *A Lattice Model of Secure Information Flow*, Communications of the ACM, vol.19, no. 5, 1976, 236–243.

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
- $\rightarrow \subseteq SC \times SC$ 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:

- $A \rightarrow B$: the information may flow from the security class A to the security class B
- $A \oplus B$: if information from the two security classes A and B are combined, the result belongs to the security class $A \oplus B$

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 1

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

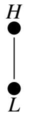
Definition 2

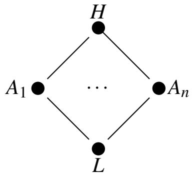
Let $(SC, \rightarrow, \oplus)$ be an information flow model and $A, B \in SC$. We say that A dominates B, denoted $A \geq B$, if $B \rightarrow A$.

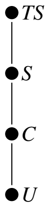
Notation and terminology:

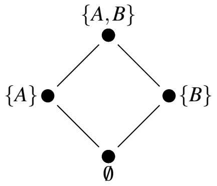
- $A > B$ (A strictly dominates B) if A dominates B and $A \neq B$
- A and B are comparable if $A \geq B$ or $B \geq A$
- A and B are incomparable if A and B are not comparable

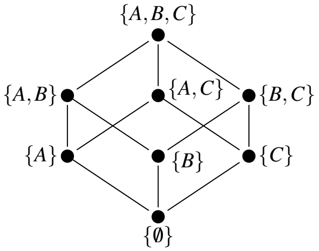
Information Flow Models: Examples











Confidentiality-based Mandatory Policies

- 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

The Bell-LaPadula Model: A Minimalist Approach

D. E. Bell, L. J. LaPadula. *Secure Computer Systems: Mathematical Foundations*, MITRE Corporation, 1973.

Overview:

- Key idea: augment DAC with MAC to enforce information flow policies
- Two-step approach:
  1. First, an access control matrix D is established
  2. Second, operations must be authorized by the mandatory access control policy

The Bell-LaPadula Model: A Minimalist Approach

The MAC in the BLP model:

- associate labels to subjects and objects by some function λ (once assigned, labels cannot be changed – this is called tranquility)
- 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 – $\boxed{s \text{ is allowed to write } o \text{ only if } \lambda(s) \leq \lambda(o)}$

Remark 1

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 – $\boxed{s \textit{ is allowed to write } o \textit{ only if } \lambda(s) = \lambda(o)}$

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

Integrity-based Mandatory Policies

- 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

K. J. Biba. *Integrity Considerations for Secure Computer Systems*, MTR-3153, The Mitre Corporation, April 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 2

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

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

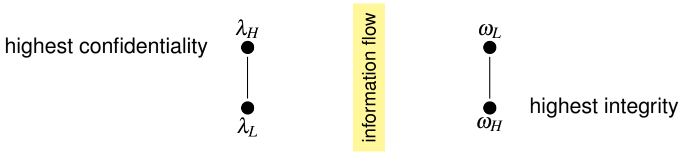
Case 3: Independent Labels, Opposite Directions

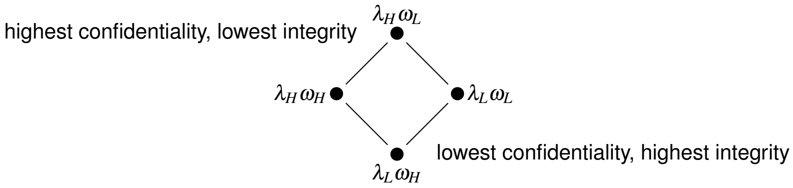
Combination 3: use independent labels for confidentiality (λ) and integrity (ω) under the assumption that the lattices have the highest security classes on opposite directions

Conclusions:

- Rules:
  1. $\boxed{s \text{ is allowed to read } o \text{ only if } \lambda(s) \geq \lambda(o) \text{ and } \omega(s) \geq \omega(o)}$
  2. $\boxed{s \text{ is allowed to write } o \text{ only if } \lambda(s) \leq \lambda(o) \text{ and } \omega(s) \leq \omega(o)}$
- 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




highest confidentiality, lowest integrity
$\lambda_H\omega_L$
$\lambda_H\omega_H$
$\lambda_L\omega_L$
$\lambda_L\omega_H$
lowest confidentiality, highest integrity

The Chinese Wall Model

D. F. C. Brewer, M. J. Nash. *The Chinese Wall security policy*, IEEE Symposium on Security and Privacy, 206–214, 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

The Chinese Wall Model

Basic elements:

- Object = item of information concerning a single corporation (company)
- Company dataset = all objects which concern the same company
- Conflict of interest class = 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 conflict of interest class CoI is chosen, a Chinese Wall is created around D and no other dataset in CoI 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

The Chinese Wall model can be states as an information flow model !

MAC Implementations

Early implementations of MAC (started out in the eighties):

- 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.
- These are focused to protect military-oriented security classification levels

More recent implementations of MAC:

- Security-Enhanced Linux (SELinux) : Linux kernel security module, incorporated into Linux kernels from 2.6, to provide the mechanism for supporting access control security policies
- Mandatory Integrity Control (MIC) : incorporated by Microsoft starting with Windows Vista and Windows Server 2008, adds integrity levels
- etc.

Mandatory Integrity Control (MIC)

- MAC implementation in Windows Vista is called Mandatory Integrity Control (MIC), which is a form of the Biba model
- It ensures integrity by controlling 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 users, authenticated users), High (local services, 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)

Security-Enhanced Linux (SELinux)

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

Concluding Remarks

- MAC provides protection against indirect information leakages
- MAC is still vulnerable to covert channels
  - covert channels were introduced by Lampson in 1973 as
    "channels not intended for information transfer at all, such as the service program's effect on system load."
  - Example: a low level subject requires a resource which is busy by a high level subject. If the system signal this to the low level subject, then there is a flow of information from the high level subject to the low level subject
  - Covert channels can exist in any MAC system that restrict information flow
  - Covert channels are hard to detect and control