# Authorization

## Chapter 4

## Authorization

- Definition of Authorization
  - Security Policy
  - Secure System
- How to create a Policy
- Access Control Models
- Type Enforcement
  - Access Control Matrix
- Security Models
  - Multilevel Security
    - Biba
    - Bell-Lapadula
  - Multilateral
    - China Wall
- Conclusion

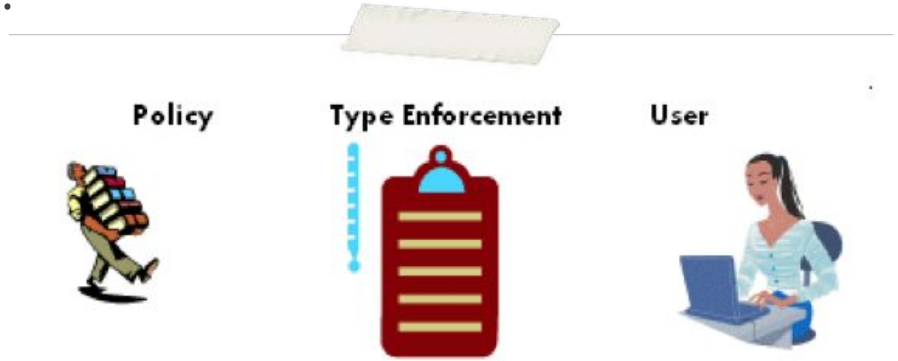
## Authorization: Definition

Honesty is the best policy
Italian Proverb

★ Unlike Authentication (which can be shared), Authorization is Application Specific.

★ Decision is based on (security) **policy**.
  - Secure in App A ≠ Security in App B
  - Analogy: Law varies among places.

★ **Type Enforcement** is to enforce the policy.
  - Analogy: Policeman to law



# Policy

★ **Policy** “It is a definite course or method of action selected from among alternatives and in light of given conditions to guide and determine present and future decisions” - [Merriam-Webster online Dictionary]

★ Aka. Don’t know how to decide, ask policy.

★ Try this:
  - Students ask “May I eat in a class”.
  - Policy: “The instructor should create supportive learning environment.”
  - Instructor answers: “If it make you learn better, then you may eat in a class.”

# Security Policy

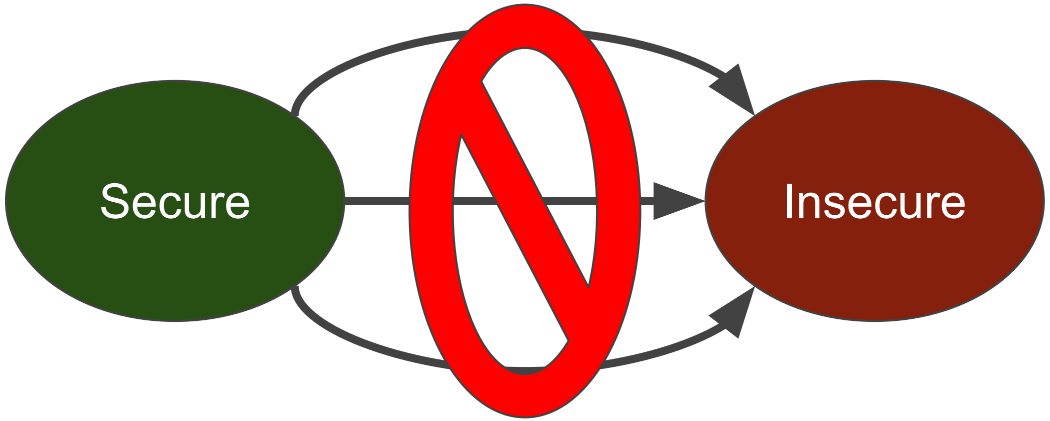
★ A **<u>security policy</u>** is a statement that partitions the states of the system into a set of authorized, or secure, states and a set of unauthorized or insecure, states. - [BISHOP].

★ Any statement that can partition the system is a security policy.

★ Try
  - Do not cross the lawn
  - No cheating in examination

## Secure System

★ Base on the security policy, we can define a secure system.
★ A secure system is a system that starts in an authorized state and cannot enter an unauthorized state. - [BISHOP]

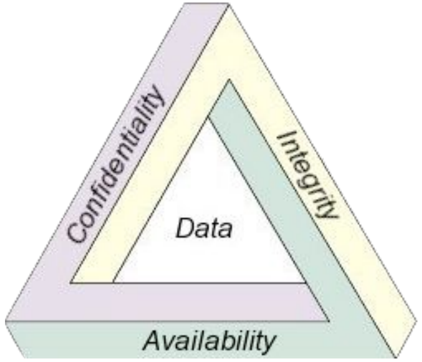


## How to create a (Security) Policy.

---

- ★ A key to authorization is a security policy.
- ★ How to create a policy?

- ★ Commonly partitioned using two properties of
  - confidentiality
  - Integrity
- ★ Additional property
  - Availability (e.g. Fault Tolerant)
- ★ Data is sensitive information, secrecy, and privacy.



the **CIA** of Security Policy

## Confidentiality and Integrity

---

- ★ **Confidentiality** is the obligation to confine or protect data from being access by unauthorized person. In another word, only the right person can access the data. (**Who can read** the data?)
- ★ **Integrity** is the condition of being unimpaired. In this context, it simple means that data is not being altered by unauthorized user. (**Who can alter/write** the data?)

## Food for Thought

Let’s put exams in an envelope and burn it.
**So, both confidentiality and integrity are preserved.**

---

- ★ What should a set of policy related to examination paper.
- ★ Hint.
  - ○ Confidentiality: Who can read? When?
  - ○ Integrity: Who can write? When?
- ★ If a student see the examination paper before the examination period, **exam leaks**?
- ★ If someone can swap/change the examination paper, **legally wrong**? (The exam. is a legal document?)

**Availability**

## Access Control Models

---

- ★ Access Control Models are models that govern who is responsible for creating security policy (based on CIA).
- ★ 3 models
- ★ Mandatory Access Control (MAC)
  - Principle: Users are untrustworthy and must be controlled.
- ★ Discretionary Access Control (DAC)
  - Principle: Users are responsible for their own data.
- ★ Role-Based Access Control (RBAC)
  - Principle: least privilege.

## Mandatory Access Control

---

★ **Admins** create policy to **control** users.

★ Multilevel systems

  ○ e.g. classified government and military information

★ Sensitivity labels (for all subjects and objects)

★ Controlling the import of information from other systems and export to other systems

★ Example.

  ○ Students must be controlled. If not, students would grade themself.

## Discretionary Access Control

★ **Owner** is **responsible** for their own data.

★ File and data ownership

★ Access rights and permissions

★ Example.
  - If is my file, I can control who can see it.

## Role-Based Access Control

- ★ RBAC is neither MAC or DAC. It is based on **least privileges**.
- ★ **Access only** when needed **to perform** a certain **function**.
- ★ Primarily granted permission based on purpose.
- ★ Example
  - RunAS (Windows), super-user do (SUDO in Unix-like OS.)
  - A doctor can see your information only when you are seeing that doctor. Unless it is emergency, the doctor can see everything.
  - A professor cannot put grade for a student in the related class.

## Food for Thought

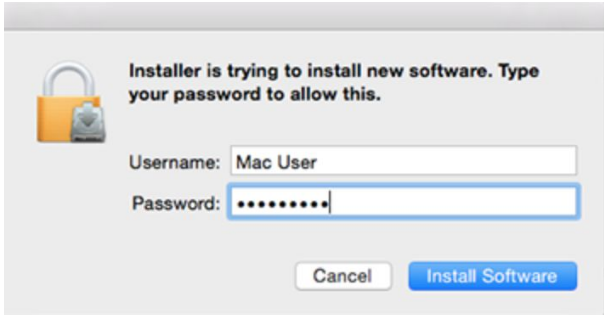
---

★ A russian spy was promoted to be a high-rank CIA. So, he was able to access every cases even though the case was not his. -- issue with MAC

★ A physician can always see all medical records of every patients in the hospital. -- issue with MAC

★ The system permission prevents parent from seeing children with inappropriate materials. -- issue with DAC

## Role-Based Access Control (ctd.)


Installer is trying to install new software. Type your password to allow this.

Username: Mac User

Password: •••••••••

★ Most application is going for Role-Based Access Control.

★ However, there is no clear definition on how to define a role.
  - Unix/Linux may create a user for each service (map role to user). (eg. www-data for web server, ftp-data for ftp)
  - An application may associate role with a method. (e.g. Object-Oriented Programming)
  - Some frameworks may ask the designer to create roles and associate permission to role. Some may even ask a user to type password to change role.

★ In a modern system, you have to type password to gain admin role for software installation.

## Pop Quiz

---

★ What is the access control model behind this system?

★ Facebook Profile
★ Google Drive
★ Unix/Linux/Windows filesystem
★ University’s registration system

## Filesystem permission

- ★ Basic Unix/Linux
  - r - read
  - w - write
  - x - execute (program)
  - x - search (directory)
- ★ Note that modern unix/linux supports extensive permissions. However, they are not enable by default.

- ★ Windows (NTFS)
  - Read
  - Write
  - Execute
  - Delete
  - change permission
  - change ownership
- ★ Unless you are an administrator of Windows Server, you are unlikely to see them.

## Access Control Matrix

---

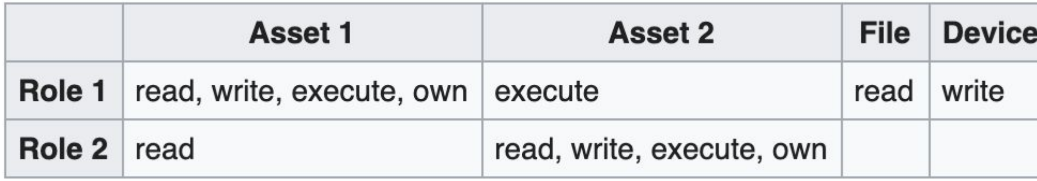
★ Eventually, policies (regardless of MAC, DAC, RBAC) are translated into a table (Access Control Matrix).

★ Access Control Matrix
- Label subjects (or roles) on rows
- Label resources on columns
- Specify the rules (based on policies)

★ Two ways to store it

★ Access Control List (by column)

★ Capability-based system (by row)



| | Asset 1 | Asset 2 | File | Device |
|---|---|---|---|---|
| **Role 1** | read, write, execute, own | execute | read | write |
| **Role 2** | read | read, write, execute, own | | |

Access Control Matrix, taken from wikipedia

## ACL vs. Capability

- ★ ACL
- ★ Associate with object
- ★ Object got a list of who can do what with the object.

- ★ Capability
- ★ Associate with subject
  Think of capability as a token (or key).
- ★ Granting a permission is passing a token/key to another person.

**Who can enter this classroom?**

**ACL**
My name is on a **list** in front of the room.

**Capability**
I got a **key** (capability) to this room.

## ACL vs. Capability (ctd.)

- ★ ACL
- ★ Easy to implement
- ★ Which files do I have access on this system?

- ★ Capabilities
- ★ ?
- ★ ?

## Fact

ACL based OS

- ★ Most (if not all) operating systems use ACL.
- ★ Why?
- ★ Note that some capability-based systems exist (in a museum).

# Fact

Windows ACL
SE Linux
Unix/Linux/MacOSX

- ★ You may access Windows ACL with the **acls** command from the command line.
- ★ Try **ls -le** on your mac os X to see full ACL.
- ★ Modern Linux, try **getfacl** and **setfacl.**
- ★ Security-Enhanced Linux (SELinux) is capable of being MAC, DAC, and RBAC. However, configuration is not trivial.

## Food for Thought

---

Will a capability-based system solve these issues?

★ What do you think?

★ A superuser can access every file in a system (including files that he/she does not create).
Do you think it is correct?

★ Your favorite word processor can open arbitrary files that are accessible by you (including binary files).
Thus, your word processor is basically you.
Does it sound right?

## Food for Thought (ctd.)

---

A **practical** capability-based system will solve a lot of security issues.
However, there is none.

★ With capability-based system, we can have a system without administrator.

★ Analogy
★ When you buy a car, the car is yours right after the dealer passes the key to you.

★ This is not true for a computer. Apple, Microsoft, Googles may access your system without permissions.

Now, you have graduated. Seeing that you have taken this class, your boss ask you to make (security) policy for company.

What will you do?

Resign?

## Security Models

---

- ★ Though we know everything, making security policies are not an easy task.
- ★ So we should have security models as guidelines

## Security Models (ctd.)

★ a high-level concept to ease understanding and specifying policy
★ a generic framework for dictating a policy system
★ Two general frameworks
    ○ Multilevel Security
    ○ Multilateral Security

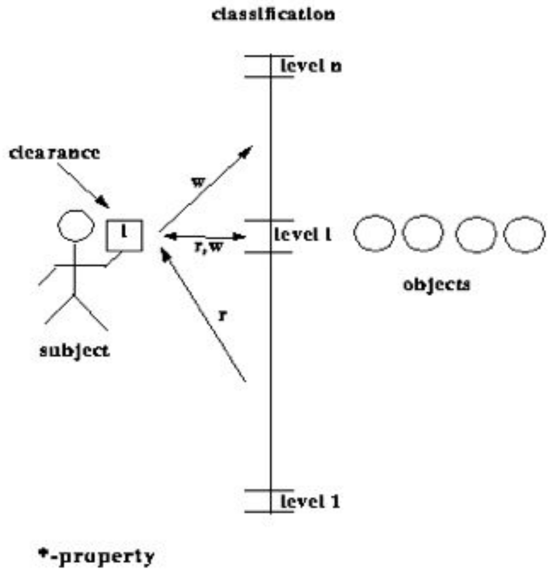
## Multilevel Security

- ★ Within an organization
- ★ Mandatory Access Control (MAC)
- ★ Two proposed models
  - ○ Bell Lapadula
  - ○ Biba

## Bell-LaPadula

---

- ★ Principle:
  The system may leak the information.
- ★ classification of military and intelligence data
- ★ High Watermark
- ★ Properties
  - ○ The simple property: no process may read data at a higher level. This is also known as **no read up** (NRU).
  - ○ The ⋆-property: no process may write data to a lower level. This is also known as **no write down** (NWD).
  - ○ The tranquility property states that the security level of an object cannot be changed while it is being processed by a computer system. (Declassify)



## Bell-LaPadula (ctd.)

---

★ We rarely create our software.
  In your computer, which software is created by yourself.
  The system should prevent software from leaking data.

★ During the war, a general may create a battle plan by asking for information from private. However, the plan may not go down to prevent information leakage.
  Once the plan is ready, the plan can be declassify to all soldiers for battle.

## Biba

---

★ Principle: Information flowing from lower level may be malicious to the system.

★ integrity, ignore confidentiality

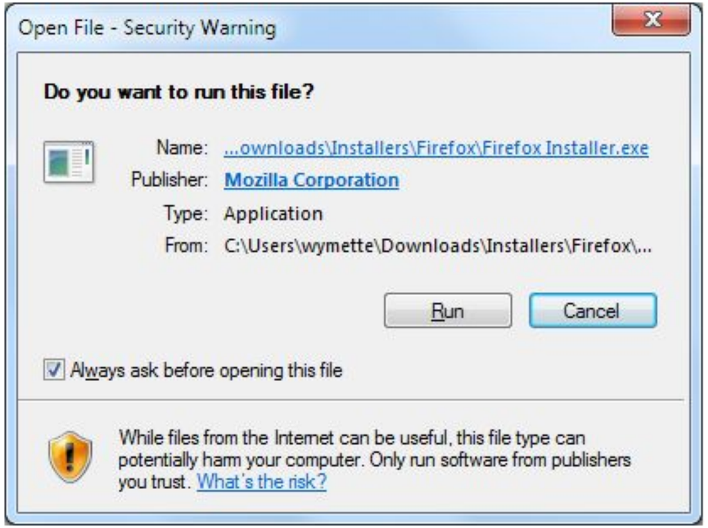
★ Low Watermark

★ Bell-LaPadula upside down

  ○ **Never read down or write up**

# Biba (ctd.)

★ See low-watermark security in your system.



Open File - Security Warning

**Do you want to run this file?**

Name: ...ownloads\Installers\Firefox\Firefox Installer.exe
Publisher: Mozilla Corporation
Type: Application
From: C:\Users\wymette\Downloads\Installers\Firefox\...

[✓] Always ask before opening this file

While files from the Internet can be useful, this file type can potentially harm your computer. Only run software from publishers you trust. What's the risk?

## Can we use both Biba and Bell Lapadula in our system at the same time?

## Multilateral Security

- ★ Between departments
- ★ China Walls model
  - ○ internal rules in a financial firm. The idea is that if a person is recently working for a company in certain sector, this person then cannot work for another company in this sector for a certain period of time.
  - ○ In several countries, financial workers rarely change jobs. (This rule does not apply to Thailand.)
- ★ Privacy.

## China Wall Model and Reverse Engineering

---

- ★ Reverse Engineering of Software
  - ○ Team A studies the software and writes a specification.
  - ○ Team B uses specification to create a new software.
- ★ Team B does not see the original software at all.

# Fact

Microsoft was (is) known as an evil empire in reverse engineering popular software.

- ★ Once, the best word processor was **Word Perfect**. When **Word Perfect 5** was released, Microsoft went to the press conference to take screenshots and list of features. Few months later, Microsoft released **Word 6.0** (jumping from 2.0, skipping 3.0-5.0) with all features found in Word Perfect 5.
- ★ Similar case on Lotus 1-2-3 4.0 vs. Excel 5.0, and others

## Conclusion

★ Let's say you have to make security policies.
★ Here are the steps.
   - Analyze a security model (e.g. Bell-Lapadula, Biba, China wall, etc.) of your system.
   - Use the model to create an access control Model (e.g. MAC, DAC, Role-based)
   - Translate it into Access Control Matrix
   - Implement a security using the appropriate type enforcement. (e.g. ACL, Capability)

Now, you have graduated. Seeing that you have taken this class, your boss ask you to make (security) policy for company.

What will you do?

Resign?

# End of Chapter 4