Protection and Security

- Protection is any mechanism for *controlling the access* of *processes to the resources of a computer system*. This mechanism must provide a means for specification of the control to be imposed and a means of enforcement.

- Security ensures user authentication preventing malicious destruction or alteration of the information stored in a computer system.

Protection

• A general protection system may be subdivided in three levels

- ❑ models
- ❑ policies
- ❑mechanisms

Models

• The protection model defines the subjects, the objects to which the subjects may access and the access rights, that is the operations by which subjects can access to objects.

• The subjects are the active part of the system (processes)

• The objects are the passive part of the system (physical and logical resources).

•A subject may have access rights either to objects or other subjects (a process can control other processes) .

• A process may have different access rights to objects, depending on what task is currently doing.

• The particular set of rigts a process has at any given time is referred to as its protection domain.

Policies

The protection policy defines the rules by which the subjects can access to the objects:

• Discretional Access Control (DAC). The owner of an object *controls the access rigths* for that object. (UNIX).

• Mandatory Access Control (MAC). The access rigths are centrally managed (ex, hospital organizations). Rules are defined in order to establish the access rights of the users. *The rules cannot be modified by the users.*

• Role Based Access Control (RBAC). Specific access rights are assigned to useres depending to their role in the organization. *A user can belong to different roles.*

Mechanisms and policies are different concepts

Example:

• UNIX provides a *mechanism* to define for each file three bits (*read, write and execute)* for the *owner* of the file, for the *group* and for the *others*

• The user defines the value of the three bits (*policy*)

Changing the protection state

Standard dual mode (monitor-user mode)

• Two protection state:user mode and monitor o kernel mode.

• Domain changing associated to the system calls

• When a process must execute a privileged instruction (access to files, I/O operations…) a change of domain happens.

•*It is not possible to have protection among users*.

Unix

• The domain is associated with the user. The change of domain corresponds to the *temporary change of the identiy among users.*

• To each file are associated the *owner identification* (user-id) and a *domain bit* (set-uid*)*.

• When a user A *(user-id=A*) begins the execution of a file, whose owner is B (*user-id=B*) and the *set-uid=on for the file*, the *user-id* of A is set temporary to B .

• When the execution completes the initial conditions are restored.

Access matrix

	O1	O2	O3
S1	read,write	execute	write
S2		execute	read,write

Access matrix implementation

:

- Matrix dimension
- Sparse matrix

Access Control List (ACL).
The matrix may be decomposed by columns: to each object an access control list is associated. It contains all the subjects that can access the object and the permitted access rigths.

Capability List
The matrix is decomposed by rows: to each subject is associated a list that contains the objects accessible by the the subject and the relative access rights.

Access list

•The access list for each object is represented by the set of informations

<subject, set of rigts >

only for the subjects with a not empty set of rights for the object.

• When an operation M must be executed on an object Oj by the subject Si, the access list is examined looking for

<Si,Rk> with M belonging to Rk

If the condition is not satisfied, *a default list* is examined. If the answer is negative , a error codition occurs.

• The default list contains the access rigths that can be executed by every subject.

•User groups .

•SID(subject identifier) , GID (group identifier). The ACL entry :

SID_1, GID_1: < set of rigths>
SID_2, GID_2: < set of rigths >

•Role concept. A user can belong to different groups and then with different set of rigts.

•When the user access the object he must specify the belonging group.

CAPABILITY LIST

• The capability list, for each subject, is the object list together with the allowed accesses for each of them to which the subject can access.

• Every element of the list is named capability.

• Every capability guarantees to the owner some access rigts on an object.

•In order to execute an operation M on an object O it is necessary that :

- The subject can access the object
- M is one of allowed rigths

	F1	F2	F3	F4	F5	F6	PR1	PR2
subject	-	-	R	RWE	RW	-	W	-

Type	**rigths**	**object**
File	R	Pointer to F3
File	RWX	Pointer to F4
File	RW	Pointer to F5
Printer	W	Pointer to printer PR1

capability list for the subject

Comparison

A protection system realized using only one of two methods, ACL or capability list, normally presents efficency problems

- **ACL**. The information about the access rigths owned by a subject is dispersed in the various ACL relative to the system objects.
  every access to the same object by a subject implies a search in the list.

- **Capability list**. In order to remove an object, whose access rights belong to more subjects, it is necessary to search in all the relative capability lists.

- Adopted solution: Utilization of a **combination** of the two methods.

Example: Unix O.S

fd=open(<nome file>, < access rigths >)

- ACL: the access is allowed?

- If the answer is positive a new entry is created in the *open files table* associated to the process, constituted by fd and the access rights (capabilty).

- **fd** is returned to the process (*i-node* corrisponding to the new open file).

- All the next operations on the file are executed by using **fd** and verifiyng that the access rigths are allowed.

Multilevel security

• The majority of the operating systems allow single users to define users that can read or write their files and their objects.
(Discretionary Access Control, DAC).

• In some environments a more stringent security is required (commonly found in military,hospitals, companies..). Rules are defined relatively the possibilty to have access to informations and these rules *cannot be modified without special permissions.*
(Mandatory Access Control, MAC)

The model is equally *applicable in other areas*, where information can be organized into gross categories and users can be granted authorization to access certain categories of data.

Por example, the highest level of security might be for *strategic corporate planning documents and data,* accessible by only corporate officers and their staff; next might come sensitive financial and personnel data, accessible only by administrators personnel, corporate officers, and so on.

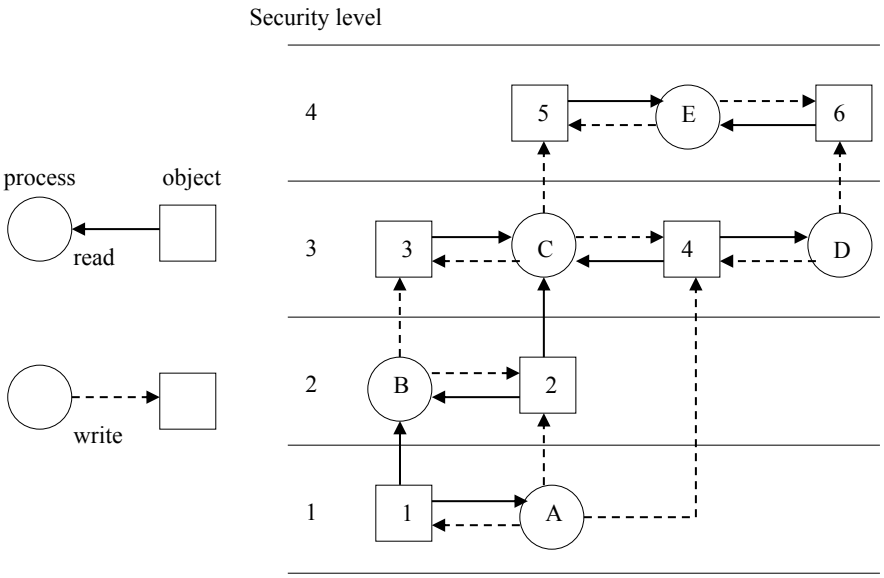
A multilevel secure system *must enforce* the following rules:

•No read up: A subject can only read an object of less or equal security level. (simple security property)

•No write down: a subject can only write into an object of greater or equal security level (*-property)

•The subjects can read downward and write upward, but the contrary is not allowed.

•The two rules, if properly enforced, provide multilevel security



Trojan Horse Defense

• In the example a troian horse is used to get around the standard security mechanism based on *the access control list* (ACL).

• A user named Bob created a file containing the critical sensitive character string "CPE1704TKS", with read/write permission provided only to processes belonging to Bob.

• A hostile user, Alice, gains legitimate access to the system and installs both a troian horse program and a private file used in the attack as a "back pocket".

•Alice gives read/write permission to himself for this file and gives to Bob write only permission.

• Alice now induces Bob to invoke the troian horse program, perhaps by advertising it as *a useful utility.*

• When the program detects that it is being executed by Bob, it reads the sensitive character string from Bob' file and copies it into Alice' back pocket file.

• Both the read and write operations satisfy the constraints imposed by Access Control List.

• Alice then has only to access Bob' file at a later time to learn the value of the string.

• Use of a multilevel protection. Two security levels, *confidential and public*. To Bob' processes and files the confidential security level is assigned. To Alice' processes and files the public level is assigned.

• When the trojan horse program is activated by Bob , the Bob security level is assigned to the program and it is possible for it to see the sensitive string of characters (no read up rule).
When the program tries to store it in a public file (*back pocket file*) the *property is violated and the attempt is not allowed.

• The security policy takes priority over ACL mechanism

Steganography

• Is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message, a form of *security through obscurity.*

• Generally, messages will appear to be something else: images, articles, shopping lists, or some other *covertext* .

•*Images utilization*. Each pixel is composed by three 8 bit numbers, one for each red, green and blu intensity . The pixel color is obtained by the linear overlapping of the three colors.

• The coding method utilizes the less significant bits as a hidden channel. Impossibility to distinguish 7 bit colors from 8 bit colors.

• Example: 1024x768 pixel image (294912 byte). The text to be transmitted is compressed (if it is necessary) encrypted and inserted into the less significative bits of each color.