UC: Segurança de Redes e Computadores

TP2 Report – Access Control – Bell-LaPadula in academic context exercise

Students (Nº / Name):

António Lourenço (68452)

Jorge Ribeiro (60027)

Pedro Alves (61893)

Introduction

The main function of access control is to limit the damage that can be done by particular groups, users and programs whether through error or malice.

In this exercise was proposed to build the access control model for an academic context using the Bell-LaPadula model.

Purpose of the Model

The purpose of this model is to achieve what is meant to be a "*secure system state*" in **how subjects in the system access objects**. These actions should be in accordance with a security policy. This "*secure state*" means that each state transition preserves security by moving from secure state to secure state.

Two-step approach:

1. Operations authorized by MAC policy, over which users have no control
2. Discretionary access matrix

We will start with the MAC policy. To determine whether a specific access mode is allowed, the clearance of a subject is compared to the classification of the object. This clearance is expressed in terms of a lattice.

Step 1 – MAC Policy, Building the lattice

The lattice is represented by a diagram (graph), which defines the levels of security that an object may have and that a subject may have access to.

- different **sensitivity levels**, are assigned to information, translating into a hierarchy
- a **clearance** is assigned to individuals, which reflects their trustworthiness

In this exercise there are 3 security levels: P (public), C (confidential) and SC (strictly confidential). (**Levels: P, C and SC; SC > C > P**). And the **Categories AS** (Academic Services) and **ScS** (Scientific Services).

There are two types of **subjects** in the system:

- Students: Clearance: C (Confidential); Categories: {AS} (academic services).
- Teachers: Clearance: C (Confidential); Categories: {AS, ScS} (academic services and scientific services).

While drawing the lattice is important to consider that it will represent the information flow as an ordering relation. Instead of a list of axioms governing user's accesses, it simply requires that information transfers obey the ordering relation. There is a finite set of security classes and categories and the diagram represents the "may flow" relations between them.

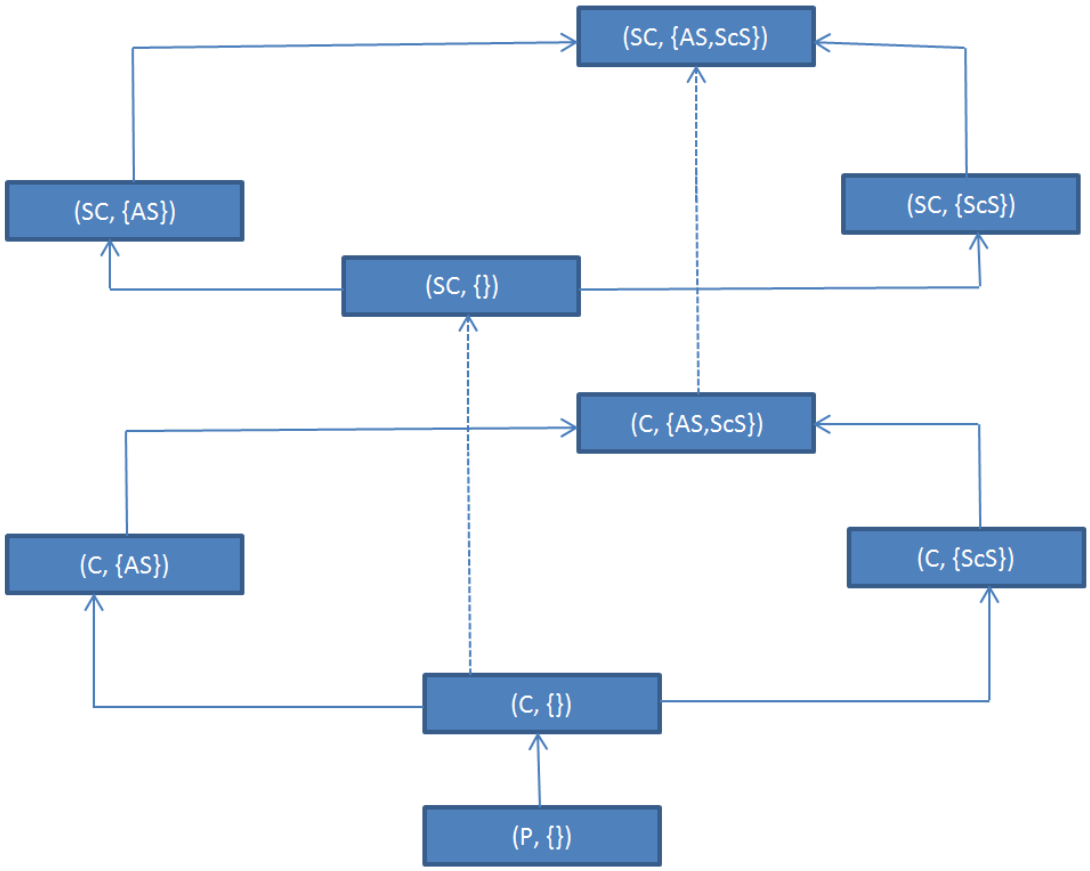

Image 1 - Lattice of access control

We tried to describe that inside the security level C, information labeled within the category ScS cannot be accessed by subjects in the category AS, but information labeled within the category AS can be accessed by subjects in the category {AS,ScS}, so the problem with professors interfering with student's work could not be solved here.

Properties important to note in information flow (Rules):

- Rule 1: **No read-up**; Simple Security property

- Rule 2: **Star Property**; no write down

In this model that means subjects labeled SC or C cannot write bellow their security level, that subjects in P cannot read in SC or C, and that subjects in C cannot read in SC.

Image illustrating this:

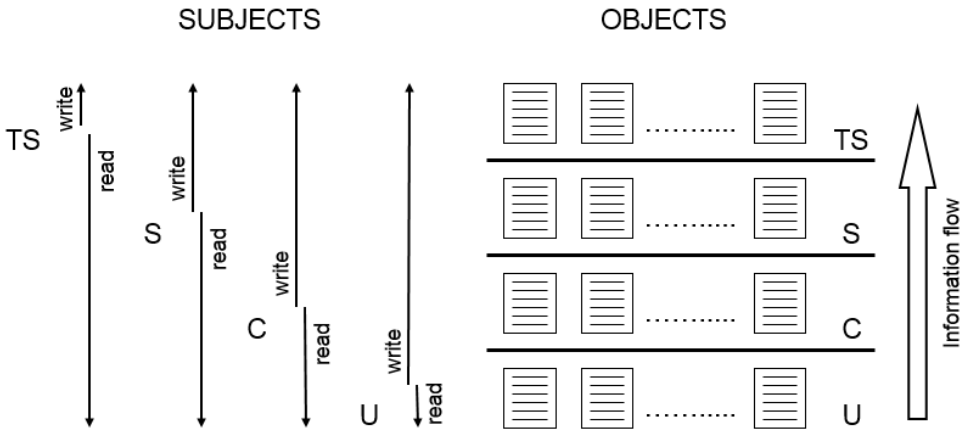

Image 2 - MAC model

Step 2 - Discretionary access matrix

For the model to be complete is needed to define a matrix which describes explicit access rules that establish **who can execute which actions on which resource.**

To do this we need, besides Subjects that we have defined before, **objects and actions.**

The **objects** of the system may vary according to the analysis. The complexity here could go on and on. We considered that we would only need two kinds, **Work** (be it homework, test, etc.) and **Grade**.

The **actions** can simply be *read* and *write*. This way we could theoretically **remedy the integrity problem** with teachers possibly interfering with student's work.

Subjects/Objects	Work	Grade
Student	W	R
Teacher	R	W

Considerations about the model's security

Tranquility property: security labels never change during system operation. This overcomes one problem with using DAC all alone, where users can pass on their rights to other users. Anyway this brings other **bigger problems**, this is not flexible and not practical in real life, as any change required while the system is operating is hard to implement, be it in objects or users.

In general this model is **too complex** and **not much usable** un real life. It focus on confidentiality, but not much on the other requirements.

A known **vulnerability** in Bell-laPadula is the covert channel exploitation (Covert channels transfer information in violation of a security policy).