CS505 Intermediate Topics in Databases Fall 2012 Project One page

The purpose of this project is to test the understanding of two basic security models for databases, DAC, and MAC. As such, it relates to the Outcome 1 of the course (Understanding of security issues in Databases.)

Background

This project builds a database to support MSM (mixed security model), MSM is different from both Discretionary Access Control (DAC) and Mandatory Access Control (MAC), but has elements of both. MSM is an extension of DAC to include two lists: a *list of forbidden privileges* and a *list of permitted privileges*. Here is the idea. Given the user (say)*marek*, we maintain a list of *forbidden privileges* for *marek*. For instance we may decide that the user *marek* can not have SELECT privileges on the table *clients*. MSM works just like DAC, but if someone tries to grant *marek* SELECT privileges on *clients*, then

1. The grant operation must be rejected (with an appropriate message sent to the grantor and the security officer).
2. The failed attempt must be logged.

We also maintain the list of *permitted privileges*. If somebody attempts to give *marek* privileges that are not on that list, we do *not* give *marek* the privilege and warn the grantor (note the subtle difference of the treatment of the two cases.) For our purposes, the only privileges are INSERT and SELECT. Neither INSERT nor SELECT has attribute lists, that is the granularity is on the table level. Further, the INSERT privilege implies the SELECT privilege (unlike the standard in DAC).

We do not have access to the system files that control security, so we will simulate them. Your project should have a database table *my_permissions* where you store the privileges. Of course, you will need to design a meaningful metadata for this table.

The scheme for forbidden_list is *forbidden_list(user, table_name, privilege, grant_option)*. The data types of attributes could be strings (and Boolean, or similar if needed). The scheme for permitted_list is *permitted_list(user, table_name, privilege, grant_option)*.

Such two lists could be inconsistent (how?), so you need to be able to *check for consistency*.

Implementation requirements

- All access to the facilities listed here must be through a graphical user interface (GUI) with dialog boxes.
- One facility must display the *forbidden_list*.

- One facility must display the *permitted_list*.
- Your GUI must allow insertion to and deletion from the *forbidden_list*.
- Your GUI must allow insertion to and deletion from the *permitted_list*.
- You need to demonstrate that you check for consistency of the access control.
- Your GUI must allow granting and revoking SELECT and INSERT privileges.
- Error messages must appear in the GUI.
- You must populate the *forbidden_list* with at least 10 entries.
- You must populate the *permitted_list* with at least 10 entries.
- You must write documentation explaining how to use your GUI and giving examples of queries that show the behavior of the system.

This project is worth 30% of the credit for the part I of this course. Devote sufficient effort to this project!

You can implement this project either on your own laptop (which you may bring it to my office for the demonstration), or on MultiLab machines. You can choose the implementation language: C++, C#, JAVA, PCP, Python and Perl are acceptable; if you want to use any other, please speak to the instructor first.

Submission requirements

- You must submit the code, documentation, and test cases in hard copy (not by email) by **Friday, October 12, 2012**.
- I might ask you to demonstrate your program *at my discretion*. If so, I will let you know by Monday, October 15. I will base my decision on your code and documentation.
- I will try to grade your project by Friday, October 19.

Alternative projects Students have an option to define and implement their own projects. However, such projects need to be specified and approved by me not later then Thursday, September 20, 2012.

Set up: September 13, 2012

Last updated: September 13, 2012