COURSE PROJECT

Mandatory Access Control

Instructions

The goal of this project is for you to apply the concepts we cover in this course to an example system. You should continue working on the same system that you used in the previous three courses.

In **Part One**, you will identify which parts of your authorization relation can be implemented with multi-level security then revise your authorization mechanisms for these components.

In **Part Two**, you will identify which parts of your authorization relation can be implemented with commercial security policies or role-based access control, then revise your authorization mechanisms for these components.

In **Part Three**, you will identify which parts of your authorization relation can be implemented with credentials-based authorization then design credentials and guards for your system.

Except as indicated, use this document to record all your project work and responses to any questions. At a minimum, you will need to turn in a digital copy of this document to your facilitator as part of your project completion. You may also have additional supporting documents that you will need to submit. Your facilitator will provide feedback to help you work through your findings.

Note: Though your work will only be seen by those grading the course and will not be used or shared outside the course, you should take care to obscure any information you feel might be of a sensitive or confidential nature.

Complete each project part as you progress through the course. Wait to submit the project until all parts are complete. Begin your course project by completing Part One below. Submit your completed project on the final project assignment page online. Information about the grading rubric is available on any of the project assignment pages online. Do not hesitate to contact your facilitator if you have any questions about the project.

Part One

Multi-Level Security

For this part of the course project, you will identify which parts of your authorization relation could be implemented with multi-level security, then you will revise your authorization mechanisms for these components. You should specify the set of labels that will be used, who will assign labels to principals and objects, what restrictions will be enforced, and whether/how labels can be modified. Feel free to revise some of your designs from the previous module, if appropriate.

The system I have been working on is the electronic voting system, specifically the online voting system. The users of the system include

1. Electoral commissioners
2. System administrators
3. Eligible voters

The authorization relation for the system is the following:

1. An eligible voter can **vote** when they are authorized to do so.
2. System administrator in consultation with the head of electoral commissioners can **grant access** to the voting system for eligible voters
3. System administrator can **grant access** to the server room.
4. Senior security officers **monitor** the entrance of the server room
5. System administrator can **revoke access** to server room
6. All principals can view election results.

Eligible voters casting their votes, principals viewing election results, system administrators and electoral commissioners accessing election data from a database could be implemented with multi-level security.

Eligible voters casting their votes is write an example of a write progress since the votes will be written and stored on a the server database, where as accessing election data or viewing

election results is a read process.

- Let V_i be a voting (writing) process
- Let RR_i be accessing (reading) election results
- Let RD_i be accessing (reading) election data

Note that voting and accessing election results or data are disjoint processes.

The labels are as follows and they will be assigned by the system administrator.
Sensitivity labels: SL = {Top Secret, Secret, Unclassified}
Compartment labels: CL = {ballot_i, Election results, Election data}

For all voters, L(V_i) = (Secret, {ballot_i})
For all readers RR_i, L(RR_i) = (Unclassified, {Election results})
For all readers RD_i, L(RD_i) = (Top Secret, {Election data})

We wish to enforce:

- Voters cannot convey content to readers RD_i.
- Readers RR_i cannot convey content to V_i and RD_i.
- RD_i can convey content to V_i, RR_i and RD_i.

The labels will not be modified.

Part Two

Commercial Access Control

You now need to decide which parts of your authorization policy could be implemented with either (1) commercial security policies or (2) role-based access control. You should then design an access control scheme to authorize those components. Feel free to revise some of your designs from Part One and/or the previous module, if appropriate.

The following part of the authorization policy could be implemented using Role-Based Access Control.

- Eligible voters casting their votes,
- principals viewing election results,
- system administrators and electoral commissioners accessing election data from a database
- Senior security officers **monitor** the entrance of the server room

The Users are: All citizens of the country conducting the election, Employees, election officers

The Roles are

- Eligible voters
- Employee
- System Administrator
- Electoral Commissioner.
- Senior security officer

Role Inheritance: System administrators, Electoral commissioners, head of electoral commission Senior security officer are all employees. Eligible voters are citizens.

Permissions: Cast ballot, access election data, access election results,

Constraints: Eligible voter may vote and view election results. System administrator may access the election data on behalf of the electoral commissioner. Electoral commissioner may access election results and announce the results. Security office protects the entrance to the sever room.

Part Three

Credentials-Based Authorization

The final step is to identify one component of your authorization policy that could be implemented with credentials-based authorization. You will design credentials and guards for your system then specify what state the guards will store and what credentials a principal must present when they make a request. Feel free to revise some of your designs from Part One and Part Two, if appropriate.

One component that could be implemented with credentials-based authorization:

System administrators and electoral commissioners gaining access to the secured server room to to gain access to the election database.

The credentials that will be need are:

- Fingerprints
- (Physical tokens) Plastic ID card with a chip, together with a PIN.

The guard will be a programmable door lock that will store the credentials.

To gain access to the secure room, a principal will need to present a Plastic ID card

together with a PIN. The principal will need to insert the plastic ID into the guard and enter a PIN. The guard will verify whether or not the PIN is valid. If the PIN is valid, the guard will then request the fingerprint of the principal. Access to the server room will be granted if all credentials are valid.

In this case, note that there will be a security officer to prevent spoofing attacks.

REFERENCE

The lecture materials

To submit this assignment, please refer to the instructions in the course.