Computer Security and Privacy (COM-301)

Discretionary Access Control

Interactive exercise solving

youAllGetASix

In order to make assignment grading easier, the COM-301 TAs have set up a grading portal at `https://youAllGetASix.com`. Students submit their assignments in PDF format via this portal.

Upon receiving a file, the grading script on the server takes the assignment as input. It reads the SCIPER from the first page of the assignment, performs the grading, and stores a report and grade associated to that on the server. This grade report is later reviewed by the TAs.

Describe one attack a student could carry out against this system. Explain the vulnerability that enables this attack. What would you advise the COM-301 TAs to do to prevent the attack

Attack: student could submit a PDF with in another student's/incorrect/non-existent SCIPER, and produce a wrong grade report.

Vulnerability Lack of check that the SCIPER in the PDF corresponds to the student submitting the file.

Fix: The fix is to add a check for to make sure that students can only submit their own SCIPER. The TAs could add a login system that ensures students are who they claim and then check the SCIPER against Moodle's records.

ACL vs Capabilities

Because of COVID-19, EPFL has decided to restrict access to the study rooms on campus: each student needs to book on the EPFL app a seat for the day in a study room to be able to get into the given room. Propose a high-level mechanism for access control of the study rooms. List subjects, objects, and rights.

Does your mechanism use the capability or access-control list model?

Name one advantage and one disadvantage of your proposal.

Subjects: students
Objects: rooms
Possible access operation: enter room or not

Example:
ACL system: For each room, store a list of students who have booked access to this room.
Advantage of an ACL system: handling number of students per room, can easily check how many students per room.
One possible disadvantage of ACL system: difficult to update permissions which is needed often in this setting

Least Privilege and Access control

Access control policies should be implemented in such a way that subjects are never “overprivileged”. In other words, subjects should have the minimal access to an object in order to perform a task.

Imagine a simple permission system where one can have the following permissions:

- r: read the content of an object
- w: write to an object
- x: execute an object

Imagine the system has two directories `submission` and `grading`.

How would you assign permissions from principals to objects implementing least privilege to:

1- Students that need to submit their report to the directory submission

2 - TAs that need to grade reports and write the result on a file `grades` in directory `grading`

3 – Professor that needs to execute a script averaging in folder grading that uses the results in the file `grades` in directory `grading`

Least Privilege and Access control

Your solution should be of the form

Principal	Object	Permission
Student		
TA		
Professor		

Think adversarially to decide on least principles.

Remember there is not only one correct solution, it depends on your threat model.

5

Principals:

- Student (consider N principals of type student)
- TAs
- Professor

Objects:

- Student's own assignment (submission/own.txt)
- Others student's assignment (submission/others.txt)
- Grades file (grades/grades.csv)
- Grading script (grades/script.sh)
- Directories themselves

ACCESS NEEDS:

Students: need to write their own assignment, but not other students' assignments. According to the question, there is no need for the students to read. They only need to submit. [If you would justify that they need to read, e.g., to check the assignment was uploaded correctly, it would be ok. As long as you state the need and provide the correct permission configuration]

TAs: need to be able to read the assignments to grade them, and they need to be able to write the grade in the grades.csv file. [As before, they do not necessarily need to be able to read grades.csv, but it is possible to justify that need]

Prof: needs to be able to read the grades and execute the script. [One could also justify need to write the script (to update) or read the assignments (to check the grades) and update the grades in grade.csv]

PERMISSIONS:

Student – Submission/their_own_assignment.txt : w
TA – Submission/ : r
TA – Grading /grades.csv : w
Prof – Grading/script: e
Prof – Grading/grades.csv: r