CS 850 DATABASE SECURITY
II SEM M.TECH (CSE-IS)
MIDSEM EXAM- 12 Feb 2024

029

Note: Answer ALL Questions **Max.Marks: 50 Marks**

Q1. By an example, show what problem/problems may arise in Bell LaPadula model if an object hierarchy function does not result in a tree, but results in a graph (such as a Directed Acyclic Graph or Graph with cycles). (8 Marks)

Q2. Discuss the revocation problem with respect to access control lists and capabilities. How might one efficiently implement a command to revoke access to an object by one particular user? (4 Marks)

Q3. The CSE department has decided to develop a new electronic grading system. Here are the key features the department wants the system to handle (do not assume any permissions or restrictions other than those explicitly mentioned): (15)

- Each course has some course instructors (CIs) and teaching assistants (TAs) assigned to it.
- TAs and CIs for a given course can both read and write that course's grades; in addition, a course's CI(s) can submit final grades.
- All faculty members can read the grades for every course.
- Graduate students and faculty are disjoint groups.
- No graduate student can be a TA for more than two courses.
- No graduate student can be a Course instructor for any course.
- Faculty members can be teaching assistants or Course instructors for any course, and there are no limits on how they are assigned to courses. (Yes, they may even serve as both the TA and the CI for the same course; I mention this only to make your task easier.)
- Because of the potential for confusion with final grades—especially if instructors have the same students in multiple courses—the electronic grading system must not permit anyone to perform in more than one Course-instructor role at any given time (e.g., login session).

As part of the system's initial test run, the department wants to configure the system to handle just three of the department's courses: CSE 123, CSE 456, and CSE 789. Thus, the department has identified the following roles and permissions for this system:

R = {Fac Grad, 123TA, 123CI , 456TA, 456CI , 789TA, 789CI }

P = {read 123, write123, submit 123, read 456, write456, submit 456, read 789, write789, submit 789}

For example, read 123, write123, and submit 123 are the permissions to (respectively) read, write, and submit the grades for CS 123.

Provide the following RBAC components to accurately meet and fulfill all of the department's desired features/criteria:

a) A role-hierarchy relation and a permission- assignment relation *PA*
b) A static separation-of-duty relation to capture necessary static constraints
c) A dynamic separation-of-duty relation to capture necessary dynamic constraints

Q3. Suppose you are the database administrator setting up security mechanisms for the Video Stores database (8 marks)

FILM (FNUM, TITLE, OPENED, LENGTH, BUDGET, DIRECTOR, SNUM, RATING, WRITER)
ACTOR (ANUM, FNAME, LNAME, BDATE, GENDER)
ROLES (FNUM, ANUM, PART, SALARY)
STUDIO (SNUM, NAME, ADDRESS, WEBSITE)
ITEM (INUM, FNUM, PRICE, RELEASE, FORMAT, BONUS, INVENTORY)
SHOPPER (SHNUM,FNAME, LNAME, SHIPADDR, CRCARD)
CART (SHNUM, INUM, QUANTITY)

Suppose that the database has the following users:

a) SPIELBERG works in the purchasing department.
b) SHYAMALN is in the shipping department.
c) MARSHALL is the manager of the customer service department.

Provide the SQL commands to grant appropriate access privileges and grant propagation privileges. Explain your choices based on how each user will be expected to use the database. If you make any assumptions about these types of users, be sure to explain them.

Q4. Consider the following PHP script for a login page: (15 Marks)

```
$username = $_GET[user];
$password = $_GET[pwd];
$sql = "SELECT *
        FROM usertable
        WHERE username = '$username'
        AND password = '$password' ";
$result = $db->query($sql);
if ($result->num_rows > 0) { /* Success */ }
else { /* Failure */ }
```

a) Explain why a URL where user is set to " ' or 1 = 1 -- " will result in a successful login.

b) Suppose we change lines 1 and 2 to

```
$username = addslashes($ GET[user])
$password = addslashes($ GET[pwd])
```

Recall that the addslashes function adds a slash before every quote. That is addslashes("a'b") will output the string "a\'b". Explain why this prevents the attack from part (a).

c) Does addslashes completely solve the problem? Consider the GBK Chinese unicode character set. Some characters in GBK are single bytes while others are double bytes. In particular, the following table shows a few GBK characters:

0x 5c = \
0x 27 = '
0x bf 27 = ¿'
0x bf 5c =

That is, the database interprets 0xbf27 as two characters, but interprets 0xbf5c as a single chinese character. Show that using addslashes as in part (b) leads to a SQL injection attack. What value of user will result in a successful login?

---Good Luck---

Department of Computer Science & Engineering

N.I.T.K.-Surathkal

II Sem M.Tech (CSE-IS)

ENDSEM EXAM- APRIL 2024

CS 850 DATABASE SECURITY

Time: 3 Hrs **Max Marks: 100**

Note : Answer ALL

Q1 a). In a medical information system that controls access to patient records and prescriptions, doctors may read and write patient records and prescriptions, nurses may read and write prescriptions only but should learn nothing about the contents of patient records. [10]

i. How can you capture this policy in a lattice model that prevents information flow from patient records to prescriptions?

ii. In your opinion, which security model is most appropriate for this policy? (Why?) Sketch a security model capturing the requirement.

iii. A doctor should not be allowed to make a prescription for herself. How can you augment your model above to prevent this kind of prescription abuse?

b) Consider the printserver scenario defined in the Authentication lab. The print server supports the following operations: [15]

print(String filename, String printer); // prints file *filename* on the specified *printer*
queue(); // lists the print queue on the user's display in lines of the form *<job number> <file name>*
topQueue(int job); // moves *job* to the top of the queue
start(); // starts the print server
stop(); // stops the print server
restart(); // stops the print server, clears the print queue and starts the print server again
status(); // prints status of printer on the user's display
readConfig(String parameter); // prints the value of the *parameter* on the user's display
setConfig(String parameter, String value); // sets the *parameter* to *value*

Not everybody working in the company has the same rights to access the print server. Alice is managing the print server, so she has the rights to perform *all*operations. Bob is the janitor who doubles as service technician, he has the rights to *start, stop* and *restart* the print server as well as inspect and modify the service parameters, i.e., invoke the *status, readConfig* and *setConfig* operations. Cecilia is a power user, who is allowed to *print* files and manage the print queue, i.e., use *queue* and *topQueue* as well as *restart* the print server when everything seems to be stuck. Finally, David, Erica, Fred and George are ordinary users who are only allowed to *print* files and display the print *queue*.

i. Prepare the access control list for the print server (i.e. the print server is considered as a single object with the different methods as the possible operations)

ii. Identify roles and define a role hierarchy and permissions for each role, so that the access control policy outlined above can be implemented?

iii. Now consider the situation where Bob leaves the company and George takes over the responsibilities as service technician. At the same time, two new employees are hired: Henry, who should be granted the privileges of an ordinary user, and Ida who is a power user and should be given the same privileges as Cecilia.

c) Assume that a database table named Students (SID, SNAME, DEPTID, LOANAMOUNT). Possible queries are COUNT, SUM and MEAN. The database does not reveal results when a small or large number of people are covered by the query. In our database it has been decided that 4-7 people must be covered by the query. We want to find a sequence of queries that tells us the size of Erika's loan. The obvious query SELECT SUM(Loan) FROM Students WHERE name = 'Erika' will not work because the result, 1, is not in the range 4-7. The query will be ignored. Find a general tracker and a sequence of queries that will give you the information. You are allowed to do 4 queries. [05]

Q2. Assume that you are the DBA for the ABC Toy Company and create a relation called Employees with fields *ename*, *dept*, and *salary*. For authorization reasons, you also define views EmployeeNames (with *ename* as the only attribute) and DeptInfo with fields *dept* and *avgsalary*. The latter lists the average salary for each department. [20]

a. Show the view definition statements for EmployeeNames and DeptInfo.

b. What privileges should be granted to a user who needs to know only average department salaries for the Toy and CS departments?

c. You want to authorize your secretary to fire people (you will probably tell him whom to fire, but you want to be able to delegate this task), to check on who is an employee, and to check on average department salaries. What privileges should you grant?

d. Continuing with the preceding scenario, you do not want your secretary to be able to look at the salaries of individuals. Does your answer to the previous question ensure this? Be specific: Can your secretary possibly find out salaries of *some* individuals (depending on the actual set of tuples), or can your secretary always find out the salary of any individual he wants to?

e. You want to give your secretary the authority to allow other people to read the EmployeeNames view. Show the appropriate command.

f. Your secretary defines two new views using the EmployeeNames view. The first is called AtoRNames and simply selects names that begin with a letter in the range A to R. The second is called HowManyNames and counts the number of names. You are so pleased with this achievement that you decide to give your secretary the right to insert tuples into the EmployeeNames view. Show the appropriate command and describe what privileges your secretary has after this command is executed.

g. Your secretary allows Todd to read the EmployeeNames relation and later quits. You then revoke the secretary's privileges. What happens to Todd's privileges?

h. Give an example of a view update on the preceding schema that cannot be implemented through updates to Employees.

i. You decide to go on an extended vacation, and to make sure that emergencies can be handled, you want to authorize your boss Joe to read and modify the Employees relation and the EmployeeNames relation (and Joe must be able to delegate authority, of course, since he is too far up the management hierarchy to actually do any work). Show the appropriate SQL statements. Can Joe read the DeptInfo view?

j. After returning from your (wonderful) vacation, you see a note from Joe, indicating that he authorized his secretary Mike to read the Employees relation. You want to revoke Mike’s SELECT privilege on Employees, but you do not want to revoke the rights you gave to Joe, even temporarily. Can you do this in SQL?

Q3. Consider the following PHP script for a login page: [15]

```
$username = $_GET[user];
$password = $_GET[pwd];
$sql = "SELECT *
        FROM usertable
        WHERE username = ’$username’
        AND password = ’$password’ ";
$result = $db->query($sql);
if ($result->num_rows > 0) { /* Success */ }
else { /* Failure */ }
```

i. Because addslashes did not eliminate the problem of SQL injection, we decide to employ some form of input filtering. A recommends the use of JavaScript to validate the input before passing it over to the server. Is this enough to ensure that SQL injection will not happen? Why?

ii. As an extension to input filtering, we decide to strip the -- characters in the input so that attackers cannot inject a comment and thereby execute an injection. Can an attacker bypass this restriction? How?

iii. Stored Procedures have a reputation for being able to defend against SQl injection attacks. With this in mind, we decide to employ the following stored procedure:

```
CREATE PROCEDURE VerifyUser
        @username varchar(50),
        @password varchar(50)
AS
BEGIN
        SELECT * FROM UserTable
        WHERE UserName = @username
        AND Password = @password;
END
GO
```

The stored procedure is called from the PHP file using the following code:

```
$sql = “CALL VerifyUser (“+ $_GET[user] + ”,” + $_GET[pwd]+”)”;
```