St. Joseph's College of Engineering & Technology Palai

Department of Computer Science & Engineering

S8 CS

RT801 Security in Computing

Module 5

Security in Computing - Module 5

Syllabus

Database Security: - Security issues

SQL security

DAC based on granting & revoking privileges - MAC for multilevel security

Statistical database security.

Contents

Part I. Database Security

Databases are to be secured from various threats such as unauthorised access.

1 Security Issues

Legal and Ethical Issues

There are legal and ethical issues in accessing databases. Some data may be private and unauthorised persons should not be able to access such data.

Policy Issues

There are policy issues at the organisational level such that credit ratings and personal medical records should not be made public.

System related Issues

For example, whether a function is to be handled at physical level, or operating system level or DBMS level comes in this category.

Level Issue

Some organisations such as military or corporate companies need to classify users and data into different security levels. They may be top secret, secret, confidential and unclassified.

Database Administrator (DBA)

Consider Oracle DBMS. A database administrator is there for managing the database system. Normally the user name of the administrator is SYSTEM.

The main functions of the administrator are,

Create user accounts

```
CREATE USER sj08dbcs10 IDENTIFIED BY qwerty;
```

This SQL command creates a user with the user name 'sj08dbcs10' and its password is set as 'qwerty'.

Privilege granting

```
GRANT RESOURCE,CONNECT TO sj08dbcs10;
```

Only when the administrator executes this command newly created user 'sj08dbcs10' is allowed to process different resources and to connect to the database. Resources mean tables, triggers, cursors, procedures, functions etc.. Thus if this user wants to create a table, this command is to be executed by the administrator.

Privilege revocation

```
REVOKE CREATE TRIGGER FROM sj08dbcs10;
```

The ability of the user 'sj08dbcs10' to create a trigger is revoked with this command. After this command, user 'sj08dbcs10' is not allowed to create triggers.

Security level assignment

This operation assigns user accounts to different security levels such as top secret, secret etc..

Auditing

Database administrator is able to view all the actions performed by different users.

2 Discretionary Access Control in a Database System

These are security mechanisms to control users from accessing database. This is done by granting and revoking privileges.

GRANT and REVOKE commands are used to grant different privileges to user accounts, and to revoke different privileges from user accounts.

Consider an example:

Oracle DBMS is installed in a system. This computer system, we call as database server. Database administrator(DBA) has the user name SYSTEM.

Let there are 60 users who want to use the database system. Then DBA creates 60 user accounts as follows:

```
CREATE USER sj08dbcs01 identified by qwerty;
CREATE USER sj08dbcs02 identified by asdf;
_
_
CREATE USER sj08dbcs60 identified by mnbvc;
```

Now all the user accounts are created. But now these users cannot connect to the database or cannot work with tables or triggers.

If the DBA allows all users to connect with the database, it issues the commands,

```
GRANT CONNECT TO sj08dbcs01;
GRANT CONNECT TO sj08dbcs02;
_
_
GRANT CONNECT TO sj08dbcs60;
```

Only when this command is issued, a user is allowed to connect (log on) to the database.

If after some time, DBA wants to disallow user 'sj08dbcs05 to access the database, it issues the command,

```
REVOKE CONNECT FROM sj08dbcs05;
```

Now onwards, user 'sj08dbcs05' cannot connect to the database.

For users to access different resources in the database, such as tables, triggers, procedures, functions, views etc.. DBA must issue commands.

Let user 'sj08dbcs02' is only allowed to create tables, then DBA issues the command,

```
GRANT CREATE TABLE TO sj08dbcs02;
```

Let user 'sj08dbcs02' is only allowed to create tables, then DBA issues the command,

```
GRANT CREATE TABLE TO sj08dbcs02;
```

Now user 'sj08dbcs02' is only allowed to create tables; this user cannot alter tables or drop tables or cannot access other resources such as views, triggers, functions etc..

Also consider a similar command,

```
GRANT CREATE TRIGGER, CREATE PROCEDURE, CREATE VIEW, ALTER TABLE TO sj08dbcs02;
```

If DBA wants to allow user 'sj08dbcs04' to access all the resources in the database such as tables, triggers, procedures, views etc.., it issues the command,

```
GRANT RESOURCE,CREATE VIEW TO sj08dbcs04;
```

Now the user 'sj08dbcs04' can work with all the resources in the database.

After sometime, if DBA wants to disallow user 'sj08dbcs04' from accessing any resource, it issues the command,

```
REVOKE RESOURCE,CREATE VIEW FROM sj08dbcs04;
```

Thus following are the steps normally done by the DBA for a new user account 'sj08dbcs10'

```
CREATE USER sj08dbcs10 IDENTIFIED BY qwerty;
GRANT RESOURCE,CREATE VIEW, CONNECT TO sj08dbcs10;
```

Now a new user 'sj08dbcs10' is created.

Because of the CONNECT option in the GRANT command, user 'sj08dbcs10' can log on to the database.

Because of the RESOURCE and CREATE VIEW option in the GRANT command, user can access all the resources such tables, triggers, procedures, views etc..

At any time, DBA can disallow a user from accessing a resource or all resources, or can disallow logging on to the database.

```
REVOKE RESOURCE,CREATE VIEW, CONNECT FROM sj08dbcs10;
```

Consider a user 'sj08dbcs10'. Let this user has a table named 'student' created as follows:

```
CREATE TABLE student (stdid int,
                      branch char(2),
```

```
                                sem int,
                                rn int,
                                name char(10),
                                marks int);
        INSERT INTO student VALUES (100, 'cs', 8, 1, 'arun', 80);
        INSERT INTO student VALUES (101, 'cs', 8, 2, 'anil', 40);
        ___
        ___
        INSERT INTO student VALUES (160, 'cs', 8, 60, 'kiran', 60);
```

Let the table 'student' owned by 'sj08dbcs10' is as follows:

stdid	branch	sem	rn	name	marks
100	cs	8	1	arun	80
101	cs	8	2	anil	40
-	-	-	-	-	-
-	-	-	-	-	-
160	cs	8	60	kiran	60

Suppose user 'sj08dbcs10' wants to allow another user 'sj08dbcs20' to access this table. For this, user 'sj08dbcs10' issues the command,

```
GRANT SELECT,DELETE ON student TO sj08dbcs20;
```

Now user 'sj08dbcs20' can execute SELECT and DELETE query on student table of user 'sj08dbcs10'.

User 'sj08dbcs20' executes the command as follows:

```
SELECT * FROM sj08dbcs10.student;
```

User 'sj08dbcs20' gets the output as follows:

stdid	branch	sem	rn	name	marks
100	cs	8	1	arun	80
101	cs	8	2	anil	40
-	-	-	-	-	-
-	-	-	-	-	-
160	cs	8	60	kiran	60

Also sj08dbcs20' can execute the commands,

```
CREATE TABLE student20 (stdid int,
                                branch char(2),
                                sem int,
                                rn int,
                                name char(10),
                                marks int);
INSERT INTO student20  SELECT * FROM sj08dbcs10.student;
```

Now a table 'student20' is created and it is filled with the contents of student table owned by 'sj08dbcs10'.

Propogation of Privileges

Let the user 'sj08dbcs10' issues the command,

```
GRANT SELECT ON student TO sj08dbcs20 WITH GRANT OPTION;
```

The term 'WITH GRANT OPTION' means that user 'sj08dbcs20' can now propogate the privilege to other users by using GRANT.

Let the user 'sj08dbcs20' executes the command,

```
GRANT SELECT ON sj08dbcs10.student TO sj08dbcs30;
```

Now user 'sj08dbcs30' can select from student table.

At any time, if user 'sj08dbcs10' revokes the grant on student, then 'sj08dbcs20' cannopt perform select on student; also 'sj08dbcs30' s ability to select from student is automatically revoked.

Specifying Privileges using Views

Consider the stduent table owned by 'sj08dbcs10':

stdid	branch	sem	rn	name	marks
100	cs	8	1	arun	80
101	cs	8	2	anil	40
-	-	-	-	-	-
-	-	-	-	-	-
160	cs	8	60	kiran	60

Now suppose user 'sj08dbcs10' wants to allow user 'sj08dbcs50' to perform select on this table, with the exception that the user 'sj08dbcs50' is not allowed to see the 'marks' column. Then user' sj08dbcs10' creates a view as,

```
CREATE VIEW student100 AS SELECT stdid, branch, sem, rn, name FROM student;
```

Now this view can be shared with the user 'sj08dbcs50' as,

```
GRANT SELECT ON student100 TO sj08dbcs50;
```

Now user 'sj08dbcs50' can execute SELECT on this view as,

```
SELECT * FROM sj08dbcs10.student100;
```

Now the output will be,

stdid	branch	sem	rn	name
100	cs	8	1	arun
101	cs	8	2	anil
-	-	-	-	-
-	-	-	-	-
160	cs	8	60	kiran

A drawback of DAC model is that they are vulnerable to malicious attacks, such as Trojan horses embedded in application programs.

3 Mandatory Access Control (MAC) in a Database System

The above specified DAC mechanism (using GRANT and REVOKE) is the main security mechanism in current RDBMSs such as Oracle. In many environements, databases need more security.

For example, in a military environment data may be classified as top secret, secret, confidential and unclassified.

Thus here, data or users are classified into different levels.

The need for multilevel security exists in military, government and intelligence applications. This approach is known as mandatory access control.

Bell- LaPadula Model

The commonly used model for multilevel security is Bell-LaPadula model.

In the military world, all data in the database have a security level such as,

unclassified,

confidential,

secret, and

top secret.

User accounts corresponding to generals, lieutenants are assigned these levels depending on which objects they are allowed to see.

A general's user account may be allowed to access all tables, whereas a lieutenant's account may be restricted to only few tables.

The rules in Bell-La Padula model are as follows:

1. The simple security property

A user at security level, k can access only objects(tuples, tables, views, triggers etc..) at its level or lower level.

For example, general's user account can access a lieutenant's data objects, but a lieutenant's user account cannot access a general's data objects.

2. The * property

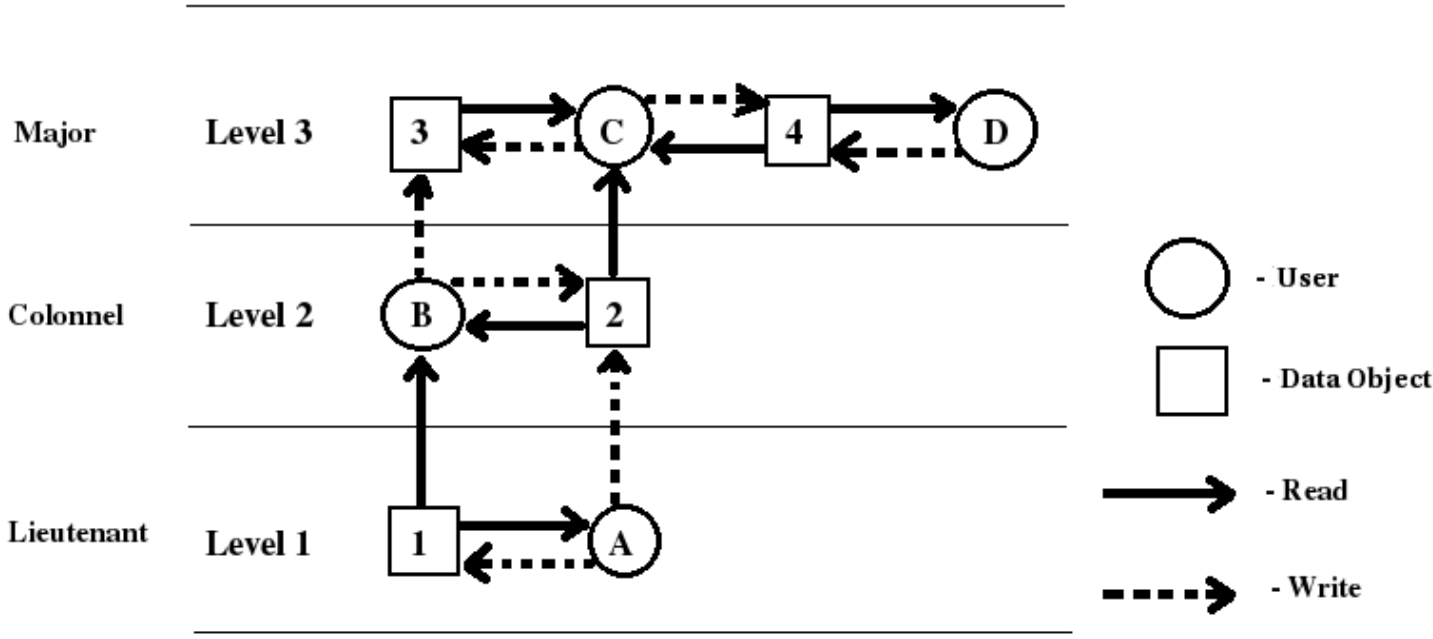
A user account at security level,k can write only data objects at its level or higher level.

For example, a lieutenant's user account can insert a tuple to a general's table, but a general cannot insert a tuple to a lieutenant' table. This is because the general may have seen top secret documents that should not be disclosed to a person in lower rank.

In short, user accounts can read down and write up, but not the reverse.

If the database system uses these measures, no information will leak out from the higher security level.

This is shown below:



In the figure, solid arrow from data object 1 to user account, B indicates that user account, B is reading the data object 1.

The dashed arrow from user account, B to data object 3 indicates that user account, B is writing on data object 3.

From the above, the solid and dashed arrows always go sideways or up.

Consider the table given below. With each data value, a level value is stored.

Employee

Name	Salary	JobPerformance	TC
Smith [1]	40000 [2]	Fair [3]	[3]
Brown [2]	80000 [3]	Good [2]	[3]

A user account with level value 3 (Major) is allowed to see the entire contents of the above table. This is because all tuples have values less than or equal to 3.

A user account with level value 2 (Colonnel) is not allowed to see the 'salary' value of Brown and 'jobperformance' value of Smith. This is because these data items have level values greater than the level value, 2. For a user account with level value, 2, table will be filtered as shown below:

Employee

Name	Salary	JobPerformance	TC
Smith [1]	40000 [2]	Null [2]	[2]
Brown [2]	Null [2]	Good [2]	[2]

Note that 'Null' is stored corresponding to the data items which had a level value 3.

A user account with level value 1 (Lietenant) is not allowed to access data items with level values 2 and 3. . This is because these data items have level values greater than the level value, 1. For a user account with level value, 1, table will be filtered as shown below:

Employee

Name	Salary	JobPerformance	TC
Smith [1]	Null [1]	Null [1]	[1]

Note that 'Null' is stored corresponding to the data items which had level values 3 and 2.

MAC models ensure high degree of protection becuase they prevent any illegal flow of information.

4 Statistical Database Security

Statistical databases are used to produce statistics about various populations. For example, Population database of a nation may contain all the details of its citizens such as social security number, name, address, date of birth, job, salary, income, etc..

These individual details must be protected from user access. But a user may be allowed to access various statistics such as average, sum, count, maximum, minimum etc..

For example, a user should not be able to access salary details of a particular person. But a user may be allowed to access the average salary of people in a particular district.

Thus statistical database security techniques must prohibit the retrieval of individual data.

This can be done by

prohibiting queries that retrieve individual data, and

by allowing only those queries that contain aggregate functions such as SUM, COUNT, AVERAGE, MIN, MAX, STANDARD DEVIATION etc.. Such queries are called statistical queries.

An unauthorised person can overcome this restriction also. Consider the following table, Person.

Person

Name	Social_ Sec _No	Income	Address	Town	State	Zip	Gender	Last_Degree

In some cases, it is possible for a user to access indicidual details by using aggregate functions. For example, If a database user wants to find the income of 'Mary Forbs', and he knows that she holds a PhD degree and she lives in Bharananganam, then he can issue a query,

```
SELECT COUNT (*) FROM Person
  WHERE Last_Degree='PhD' AND Gender='F' AND Town='Bharananganam' AND State='Kerala';
```

Let the user gets the answer as 1. This means there exists only one person who holds 'PhD' in City 'Bharananganam'. Next, if the user types the following query, he can find out the income of that person, ie 'Mary Forbs'.

```
SELECT AVG (Income) FROM PERSON
  WHERE Last_Degree='PhD' AND Gender='F' AND Town='Bharananganam' AND State='Kerala';
```

This possibility can be reduced if no statistical queries are permitted whenever the number of tuples in the result for the specified condition falls below a certain value.

Questions

MGU/May2012

1. Describe SQL security (4marks).

2. Write the need for database security (4marks).

3a. Discuss the security issues for databases and how database security is provided.

OR

b. i)Write briefly on statistical database security.

ii)Explain how MAC provides provides multilevel security for databases (12marks).

MGU/May2010

1. Define the term SQL security (4marks).

2a. Explain the necessity of database security and describe the statistical database security.

OR

b. Explain MAC for multilevel security (12marks).

MGU/Nov2009

1. What are the security issues in database security (4marks)?

2. Explain the functions of MAC (4marks).

3a. Explain SQL security DAC based on granting.

OR

b. Explain security issues in databases and how the issues are achieved (12marks).

MGU/June 2009

1. Explain security requirement in database (4marks).

2a. Explain MAC for multilevel security.

OR

b. Explain the necessity of database security and describe the statistical database security (12marks).

MGU/May2008

1. Discuss the security issues in a database (4marks).

2a. What are the security requirements in a database? Explain how security is achieved in a database.

OR

b. Explain the different methods used in statistical database security (12marks).

MGU/Nov2008

1. Discuss the security requirements for a database (4marks).

2a. Discuss the security issues in a database. How is security achieved in a database?

OR

b. Write briefly on statistical database security (12marks).

MGU/July2007

1. Define functions of MAC (4marks).

2. What are security requirements of database (4marks)?

3a. Explain how does the database and SQL security is achieved.

OR

b. Explain the various methods used in statistical database security (12marks).

MGU/Jan2007

1. What are the security issues in databases (4marks)?

2. Define multilevel security (4marks).

3a. How to make a statistical database secure.

OR

b. Explain MAC for multilevel security (12marks).

MGU/July2006

1. Define SQL security (4marks).

2. Describe discretionary security mechanism (4marks).

3a. What are security requirements of databases.

OR

b. Discuss the aspect of security versus precision in system (12marks).

References

.

Stallings, W (2006). Cryptography and Network Security. Pearson Education.

Ramachandran, J (2002). Designing Security Architecture Solutions. Wiley Dreamtech.

website: http://sites.google.com/site/sjcetcssz