Database Security and Authorization

Chapter 23

Chapter Objectives

- To discuss the techniques used for protecting the database against persons who are not authorized to access either certain parts of the database or the whole database

Chapter Outline

- Introduction
- Access Control Methods
- Discretionary Access Control
- Mandatory Access Control
- Role Based Access Control
- Introduction to Statistical Database Security

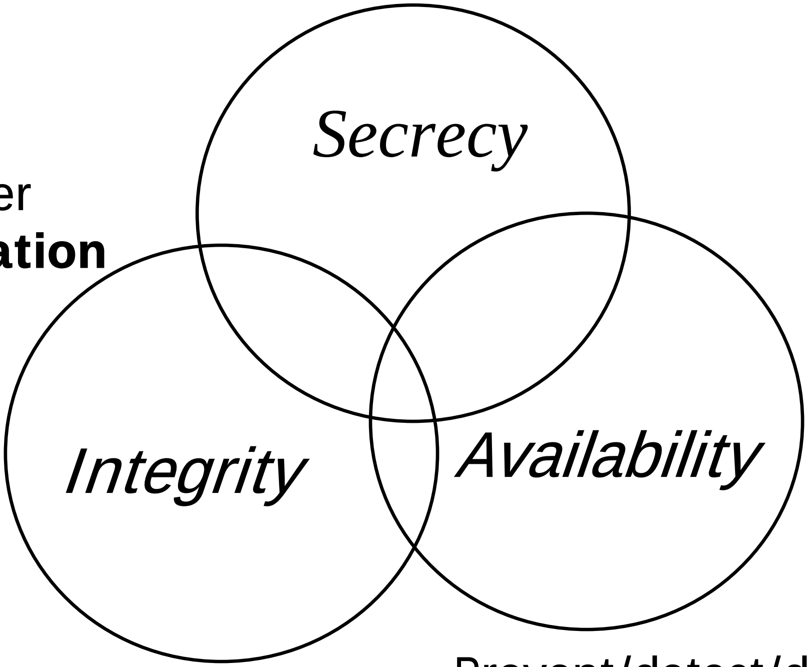
- Introduction

- Security Issues
  - Legal and Ethical
  - Policy
  - System-related
  - Security levels and categories
- Security Threats
  - Loss of integrity
  - Loss of Confidentiality
  - Loss of Availability

-- Security Objectives

Prevent/detect/deter improper **Disclosure** of information

Prevent/detect/deter Improper **modification** of information



Prevent/detect/deter improper **Denial of access** to services

-- Security Control Mechanisms

- Access control
  - creating user accounts and passwords to control login process by the DBMS
- Inference control
  - The countermeasures to **statistical database security** problem
- Flow control
  - Prevents information from flowing in such a way that it reaches unauthorized users
- Encryption
  - protect sensitive data that is being transmitted via some type communication network.

--- Access Control

- Subject: active entity that requests access to an object
  - e.g., user or program
- Object: passive entity accessed by a subject
  - e.g., record, relation, file
- Access right (privileges): how a subject is allowed to access an object
  - e.g., subject s can read object o

-- Database Security and DBA

- The DBA is a person who has a **DBA account** in the DBMS, sometimes called a **system** or **superuser account**, which provides powerful capabilities

- The DBA is responsible for the overall security of the database system.
  1. Account creation
  2. Privilege granting
  3. Privilege revocation
  4. Security level assignment

- Action 1 is access control, whereas 2 and 3 are discretionary and 4 is used to control mandatory authorization

--- Access Protection, User Accounts, and Database Audits ...

- Whenever a person or group of person s need to access a database system, the individual or group must first apply for a user account. The DBA will then create a new **account number** and **password** for the user if there is a legitimate need to access the database
- The user must **log in** to the DBMS by entering account number and password whenever database access is needed
- The database system must also keep track of all operations on the database that are applied by a certain user throughout each **login session**

… --- Access Protection, User Accounts, and Database Audits

- To keep a record of all updates applied to the database and of the particular user who applied each update, we can modify ***system log***, which includes an entry for each operation applied to the database that may be required for recovery from a transaction failure or system crash.

- If any tampering with the database is suspected, a **database audit** is performed, which consists of reviewing the log to examine all accesses and operations applied to the database during a certain time period

- A database log that is used mainly for security purposes is sometimes called an **audit trail**.

- Access Control Methods

- Discretionary Access Control (DAC)
  - grants privileges to users, including the capability to access specific data files, records, or fields in a specific mode (such as read, insert, delete, or update).
- Mandatory Access Control (MAC)
  - classifies users and data into multiple levels of security, and then enforces appropriate rules
- Role-Based Access Control (RBAC)

- DAC ...

- The typical method of enforcing **discretionary access control** in a database system is based on the granting and revoking **privileges**
- Has two levels:
  - **Account level**
    - Create objects (table, view, index, Triggers, Procedures, etc)
    - Alter objects
    - Drop objects
  - **Table level**
    - MODIFY privilege, to insert, delete, or update tuples; and the
    - SELECT privilege
    - REFERENCES privilege

... - DAC ...

- Whenever the owner A of a relation R grants a privilege on R to another account B, privilege can be given to B *with* or *without* the GRANT OPTION.

- If the GRANT OPTION is given, this means that B can also grant that privilege on R to other accounts. Suppose that B is given the GRANT OPTION by A and that B then grants the privilege on R to a third account C, also with GRANT OPTION. In this way, privileges on R can **propagate** to other accounts without the knowledge of the owner of R.

- If the owner account A now revokes the privilege granted to B, all the privileges that B propagated based on that privilege should automatically be revoked by the system

-- Example ...

- Suppose that the DBA creates four accounts --A1, A2, A3, and A4-- and wants only A1 to be able to create base relations; then the DBA must issue the following GRANT command in SQL:

```
GRANT CREATE TAB TO A1;
```

- In SQL2 the same effect can be accomplished by having the DBA issue a CREATE SCHEMA command as follows:

```
CREATE SCHAMA EXAMPLE AUTHORIZATION A1;
```

... -- Example ...

- User account A1 can create tables under the schema called EXAMPLE.
- Suppose that A1 creates the two base relations EMPLOYEE and DEPARTMENT; A1 is then **owner** of these two relations and hence *all the relation privileges* on each of them
- Suppose that A1 wants to grant A2 the privilege to insert and delete tuples in both of these relations, but A1 does not want A2 to be able to propagate these privileges to additional accounts

```
GRANT INSERT, DELETE ON EMPLOYEE, DEPARTMENT TO A2;
```

... -- Example ...

EMPLOYEE

NAME	SSN	BDATE	ADDRESS	SEX	SALARY	DNO

DEPARTMENT

DNUMBER	DNAME	MGRSSN

… -- Example …

- Suppose that A1 wants to allow A3 to retrieve information from either of the two tables and also to be able to propagate the SELECT privilege to other accounts. A1 can issue the command:

```
GRANT SELECT ON EMPLOYEE, DEPARTMENT TO A3 WITH GRANT OPTION;
```

- A3 can grant the SELECT privilege on the EMPLOYEE relation to A4 by issuing:

```
GRANT SELECT ON EMPLOYEE TO A4;
```

- (Notice that A4 can not propagate the SELECT privilege because GRANT OPTION was not given to A4.)

… -- Example …

- Suppose that A1 decides to revoke the SELECT privilege on the EMPLOYEE relation from A3; A1 can issue:

```
REVOKE SELECT ON EMPLOYEE FROM A3;
```

- (The DBMS must now automatically revoke the SELECT privilege on EMPLOYEE from A4, too, because A3 granted that privilege to A4 and A3 does not have the privilege any more.)

… -- Example …

- Suppose that A1 wants to give back to A3 a limited capability to SELECT from the EMPLOYEE relation and wants to allow A3 to be able to propagate the privilege. The limitation is to retrieve only the NAME, BDATE, and ADDRESS attributes and only for the tuples with DNO=5. A1 then create the view:

```
CREATE VIEW A3.EMPLOYEE AS
SELECT NAME, BDATE, ADDRESS
FROM EMPLOYEE
WHERE DNO = 5;
```

- After the view is created, A1 can grant SELECT on the view A3EMPLOYEE to A3 as follows:

```
GRANT SELECT ON A3 EMPLOYEE TO A3 WITH GRANT OPTION;
```

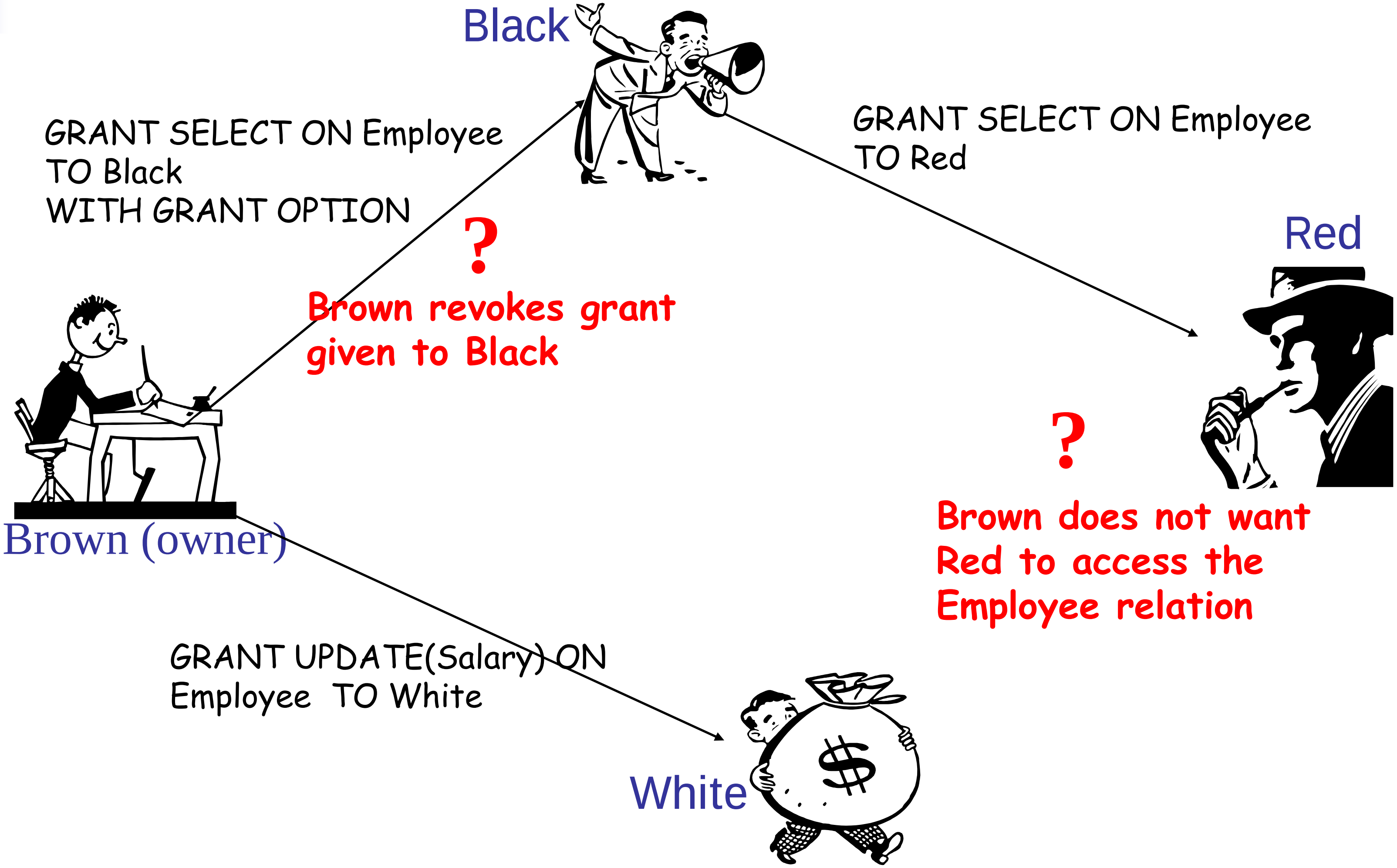
… -- Example

- Finally, suppose that A1 wants to allow A4 to update only the SALARY attribute of EMPLOYEE; A1 can issue:

```
GRANT UPDATE ON EMPLOYEE (SALARY) TO A4;
```

- (The UPDATE or INSERT privilege can specify particular attributes that may be updated or inserted in a relation. Other privileges (SELECT, DELETE) are not attribute specific.)

-- Problems with DAC



Black

GRANT SELECT ON Employee
TO Black
WITH GRANT OPTION

?

Brown revokes grant
given to Black

GRANT SELECT ON Employee
TO Red

Red

?

Brown does not want
Red to access the
Employee relation

Brown (owner)

GRANT UPDATE(Salary) ON
Employee TO White

White

-- Techniques to limit the propagation of privileges

- Limiting **horizontal propagation** to an integer number *k:* means that an account B given the GRANT OPTION can grant the privilege to at most *k* other accounts.
- **Vertical propagation** is more complicated; it limits the depth of the granting of privileges
- They have not yet been implemented in most DBMSs and are not a part of SQL.

- Mandatory Access Control (MAC) ...

- **Security label**
  - Top-Secret, Secret, Public
- **Objects:** security classification
  - File 1 is Secret, File 2 is Public
- **Subjects:** security clearances
  - Ali is cleared to Secret, Mustafa is cleared to Public
- **Dominance (≥)**
  - Top-Secret ≥ Secret ≥ Public

… - MAC

- **Access rights**: defined by comparing the security classification of the requested objects with the security clearance of the subject
  - If *access control rules* are satisfied, access is permitted Otherwise access is rejected

- Two restrictions are enforced on data access based on subject/object classification
  1. Simple property: A subject S is not allowed read access an object O unless class(S) >= Class(O)
  2. Star property: A subject S is not allowed to write an object O unless class(S) <= class(0)

- Role-Based Access Control ...

- Mandatory access control is rigid because the security class should be assigned to each subject and data object.
- In the real world, access privileges are associated with the role of the person in the organization. (example: bank teller)
- Each role is created and is granted/revoked privileges.
- Each user is granted/revoked roles.

- Inference Control

- Must prohibit the retrieval of individual data through statistical (aggregate) operations on the database.

Example:

```
SELECT  MAX(Salary)
FROM     EMPLOYEE
WHERE  Dept = 'CSE'
AND Address  LIKE  '%Bahrain%' ;
```

Note: What if only one or few employees are from Bahrain?

-- Solutions for Inference Control

- No statistical queries are permitted whenever the number of tuples in the selected population is smaller than a certain number.
- Prohibit a sequence of queries that refer to the same population of tuples repeatedly.
- Partition the database into groups larger than certain size, and queries can refer to any complete group or set of groups, but never to a subset of a group.

END