204320 - Database Management

Chapter 24

Database Security

Adapted for 204320

by Areerat Trongratsameethong

Outline

- 1 Database Security and Authorization
- 2 Discretionary Access Control

1. Introduction to Database Security Issues

- Types of Security
  - Legal (ถูกกฎหมาย) and ethical (จริยธรรม) issues
  - Policy (เชิงนโยบาย) issues
  - System-related issues: เช่น การป้องกันระดับ OS, account เข้า Windows, Folder, …
  - The need to identify multiple security levels: เช่น Top Secret, Secret, Medium, …

1. Introduction to Database Security Issues (2)

- Threats (การคุกคาม) to databases
  - Loss of **integrity:** ความถูกต้องของข้อมูล เสียหายจากคนที่ไม่มีสิทธิ์เข้าถึงข้อมูล
  - Loss of **availability:** ความพร้อมใช้งาน คนที่มีสิทธิเข้าถึงข้อมูลกลับไม่สามารถใช้งานข้อมูลที่เกี่ยวข้องได้
  - Loss of **confidentiality:** ความลับ ข้อมูลที่เป็นความลับถูกคนที่ไม่สิทธิเข้าถึงข้อมูล เข้าถึงข้อมูลได้ เช่น ข้อมูลเงินเดือน ถูกคนที่ไม่มีสิทธิ hack เข้าไปดูได้
- To protect databases against these types of threats four kinds of countermeasures (ตอบโต้) can be implemented:
  - **Access control**
  - **Inference control**
  - **Flow control**
  - **Encryption**

1. Introduction to Database Security Issues (3)

- A DBMS typically includes a database security and authorization subsystem that is responsible for ensuring the security portions of a database against unauthorized access.
- โดยปกติแล้ว DBMS จะมีระบบย่อยเกี่ยวกับการรักษาความปลอดภัยของฐานข้อมูลเพื่อป้องกันไม่ให้คนที่ไม่มีสิทธิเข้าถึงฐานข้อมูลได้
- Two types of database security mechanisms:
  - **Discretionary** security mechanisms: ให้สิทธิ ยกเลิกสิทธิการเข้าถึงข้อมูลในฐานข้อมูล
  - **Mandatory** security mechanisms: Multilevel Control (enforcing multiple levels of security)
    → ในวิชานี้ไม่ได้ลงรายละเอียดในหัวข้อนี้

1. Introduction to Database Security Issues (4)

- The security mechanism of a DBMS must include provisions for restricting access to the database as a whole
  - This function is called **access control** and is handled by creating user accounts and passwords to control login process by the DBMS.
  - การควบคุมการเข้าถึงข้อมูล ผ่านการให้สิทธิ และยกเลิกสิทธิ
  - การควบคุมการเข้าถึงข้อมูล ทำผ่านการสร้าง user account และ password

1. Introduction to Database Security Issues (5)

- The security problem associated with databases is that of controlling the access to a **statistical database**, which is used to provide statistical information or summaries of values based on various criteria.
  - The countermeasures to **statistical database security** problem is called **inference control measures**.

เช่น ระบบฐานข้อมูลประชากรที่นำเสนอข้อมูลสถิติของประชากร การควบคุมเชิงอนุมาน จะกำหนดสิทธิไม่ให้คนที่ไม่ใช่เจ้าของข้อมูลเห็นรายละเอียดข้อมูลของประชากรแต่ละคน

1. Introduction to Database Security Issues (6)

- Another security is that of **flow control**, which prevents information from flowing in such a way that it reaches unauthorized users.
  → ควบคุมการไหลของข้อมูล ป้องกันไม่ให้ข้อมูลไหลไปยังคนที่ไม่มีสิทธิในการเข้าถึงข้อมูล
- Channels that are pathways for information to flow implicitly in ways that violate the security policy of an organization are called **covert channels**.
- **covert channels**: การไหลของข้อมูลที่ละเมิดความปลอดภัยคือ ไหลไปยังคนที่ไม่มีสิทธิในการเข้าถึงข้อมูล

1. Introduction to Database Security Issues (7)

- A final security issue is **data encryption**, which is used to protect sensitive data (such as credit card numbers) that is being transmitted via some type communication network.

  → การเข้ารหัสข้อมูล เพื่อป้องกันข้อมูลความลับ เช่น หมายเลขเครดิตการ์ด ที่ส่งผ่านระบบเครือข่าย

- The data is **encoded** using some **encoding algorithm**.
  - An unauthorized user who access encoded data will have difficulty deciphering it, but authorized users are given decoding or decrypting algorithms (or keys) to decipher (ถอดรหัส) data.
  - ข้อมูลถูกเข้ารหัสด้วย Algorithm ที่ใช้เข้ารหัส โดยที่ข้อมูลถูกแปลงให้อยู่ในรูปแบบที่ทำให้ยากต่อการตีความสำหรับคนที่ไม่มีสิทธิในการเข้าถึงข้อมูล แต่สำหรับคนที่มีสิทธิในการเข้าถึงข้อมูลสามารถถอดรหัสข้อมูลได้ด้วย Algorithm ที่ใช้ถอดรหัส

1.2 Database Security and the DBA

- The database administrator (**DBA**) is the central authority for managing a database system.
  - The DBA's responsibilities include
    - granting privileges to users who need to use the system
    - classifying users and data in accordance with the policy of the organization

  คนที่ทำหน้าที่กำหนดความปลอดภัยให้กับฐานข้อมูลคือ DBA เป็นคนที่ให้สิทธิกับผู้ใช้ และจำแนกผู้ใช้ โดยเป็นคนกำหนดว่า ผู้ใช้ประเภทใด เข้าถึงข้อมูลส่วนใดได้บ้าง
- The DBA is responsible for the overall security of the database system.

1.2 Database Security and the DBA (2)

- The DBA has a DBA account in the DBMS
  - Sometimes these are called a system or superuser account
  - These accounts provide powerful capabilities such as:
    - 1. Account creation
    - 2. Privilege granting
    - 3. Privilege revocation (ยกเลิก)
    - 4. Security level assignment
  - Action 1 is access control, whereas 2 and 3 are discretionary and 4 is used to control mandatory authorization

1.3 Access Protection, User Accounts, and Database Audits

- Whenever a person or group of person 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 id** and **password** for the user if he/she deems (ถือว่า) there is a legitimate (มีสิทธิอันชอบธรรม) need to access the database
- The user must log in to the DBMS by entering account id and password whenever database access is needed.

1.3 Access Protection, User Accounts, and Database Audits(2)

- The database system must also keep **track of all operations** on the database that are applied by a certain user throughout **each login session**.
  - 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.

1.3 Access Protection, User Accounts, and Database Audits(3)

- If any tampering (มีแนวโน้ม) with the database is suspected (น่าสงสัย), a **database audit** is performed
  - A database audit consists of reviewing the log to examine all accesses and operations applied to the database during a certain time period.
- A database log (ลงบันทึก) that is used mainly for security purposes is sometimes called an **audit trail**.

Discretionary Access Control Based on Granting and Revoking Privileges

- The typical method of enforcing **discretionary access control** in a database system is based on the **granting** and **revoking privileges**.
- discretionary access control ในระบบฐานข้อมูลจะดำเนินการในลักษณะ การให้สิทธิ และการยกเลิกสิทธิ

2.1Types of Discretionary Privileges

- The **account level**:
  - At this level, the DBA specifies the particular privileges that each account holds independently of the relations in the database.
  - DBA เป็นคนกำหนดสิทธิในการเข้าถึงฐานข้อมูลในระดับ user account ที่ไม่เกี่ยวข้องกับสิทธิในการเข้าถึงฐานข้อมูลในระดับ relation
- The **relation level** (or **table level**):
  - At this level, the DBA can control the privilege to access each individual relation or view in the database.
  - ในระดับนี้ DBA สามารถควบคุมสิทธิในการเข้าถึงข้อมูลในระดับ relation หรือ view ที่จัดเก็บอยู่ในฐานข้อมูล

2.1Types of Discretionary Privileges(2)

- The privileges at the **account level** apply to the capabilities provided to the account itself and can include
  - the **CREATE SCHEMA** or **CREATE TABLE** privilege, to create a schema or base relation;
  - the **CREATE VIEW** privilege;
  - the **ALTER** privilege, to apply schema changes such adding or removing attributes from relations;
  - the **DROP** privilege, to delete relations or views;
  - the **MODIFY** privilege, to insert, delete, or update tuples;
  - and the **SELECT** privilege, to retrieve information from the database by using a **SELECT** query.

2.1Types of Discretionary Privileges(3)

- The second level of privileges applies to the **relation level**
  - This includes **base relations** and virtual (**view**) relations.
- The granting and revoking of privileges generally follow an authorization model for discretionary privileges known as the access matrix model where
  - The **rows** of a matrix M represents **subjects** (users, accounts, programs)
  - The **columns** represent **objects** (relations, records, columns, views, operations).
  - Each position **M(i,j)** in the matrix represents the types of privileges (read, write, update) that **subject i** holds on **object j**.

2.1Types of Discretionary Privileges(4)

- To control the granting and revoking of relation privileges, each relation R in a database is assigned and **owner account**, which is typically the account that was used when the relation was created in the first place.
  - The owner of a relation (the account that created) is given all privileges on that relation.
  - In SQL2, the DBA can assign and owner to a whole schema by creating the schema and associating the appropriate authorization identifier with that schema, using the **CREATE SCHEMA** command.
  - The owner account holder can **pass privileges** on any of the owned relation to other users by **granting** privileges to their accounts.

2.1Types of Discretionary Privileges(5)

- In SQL the following types of privileges can be granted on each individual relation R:
  - **SELECT** (retrieval or read) privilege on R:
    - Gives the account retrieval privilege.
    - In SQL this gives the account the privilege to use the **SELECT** statement to retrieve tuples from R.
  - **MODIFY** privileges on R:
    - This gives the account the capability to modify tuples of R.
    - In SQL this privilege is further divided into **UPDATE**, **DELETE**, and **INSERT** privileges to apply the corresponding SQL command to R.
    - In addition, both the **INSERT** and **UPDATE** privileges can specify that only certain attributes can be updated by the account.

2.1Types of Discretionary Privileges(6)

- In SQL the following types of privileges can be granted on each individual relation R (contd.):
  - **REFERENCES** privilege on R:
    - This gives the account the capability to **reference** relation R when specifying integrity constraints.
    - The privilege can also be **restricted** to specific attributes of R.
- Notice that to create a **view**, the account must have **SELECT** privilege on all relations involved in the view definition.

2.2 Specifying Privileges Using Views

- The mechanism of **views** is an important discretionary authorization mechanism in its own right. For example,
  - If the owner A of a relation R wants another account B to be able to retrieve only some fields of R, then A can create a view V of R that includes only those attributes and then grant SELECT on V to B.
  - The same applies to limiting B to retrieving only certain tuples of R; a view V' can be created by defining the view by means of a query that selects only those tuples from R that A wants to allow B to access.

2.3 Revoking Privileges

- In some cases it is desirable to grant a privilege to a user temporarily. For example,
  - The owner of a relation may want to grant the **SELECT** privilege to a user for a specific task and then revoke that privilege once the task is completed.
  - Hence, a mechanism for **revoking** privileges is needed. In SQL, a **REVOKE** command is included for the purpose of **canceling privileges**.

2.4 Propagation of Privileges using the GRANT OPTION

- 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.

2.5 An Example

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

```
GRANT CREATETAB TO A1;
```

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

```
CREATE SCHEMA EXAMPLE AUTHORIZATION A1;
```


2.5 An Example (2)

- 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;
```


2.5 An Example (3)

EMPLOYEE

Name	Ssn	Bdate	Address	Sex	Salary	Dno

DEPARTMENT

Dnumber	Dname	Mgr_ssn

Figure 24.1
Schemas for the two relations EMPLOYEE and DEPARTMENT.

2.5 An Example (4)

- 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't propagate the SELECT privilege because GRANT OPTION was not given to A4

2.5 An Example (5)

- 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.

2.5 An Example (6)

- 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 A3EMPLOYEE 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 A3EMPLOYEE TO A3
   WITH GRANT OPTION;
```


2.5 An Example (7)

- 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.

Granting Roles to User in Oracle

- It's easier to GRANT or REVOKE privileges to the users through a role rather than assigning a privilege directly to every user.
- If a role is identified by a password, then, when you GRANT or REVOKE privileges to the role, you definitely have to identify it with the password.
- We can GRANT or REVOKE privilege to a role as below.
- For example: To grant CREATE TABLE privilege to a user by creating a testing role:
- First, create a testing Role

  CREATE ROLE testing
- Second, grant a CREATE TABLE privilege to the ROLE testing. You can add more privileges to the ROLE.

  GRANT CREATE TABLE TO testing;

Reference: http://beginner-sql-tutorial.com/sql-grant-revoke-privileges-roles.htm

Granting Roles to User in Oracle (2)

GRANT or REVOKE privileges to the users through a role (Cont.)

- Third, grant the role to a user.

  GRANT testing TO user1;

REVOKE privileges

- To revoke a CREATE TABLE privilege from testing ROLE, you can write:

  REVOKE CREATE TABLE FROM testing;

- The Syntax to drop a role from the database is as below:

  DROP ROLE role_name;

- For example: To drop a role called developer, you can write:

  DROP ROLE testing;

Reference: http://beginner-sql-tutorial.com/sql-grant-revoke-privileges-roles.htm

Summary

- 1 Database Security and Authorization
- 2 Discretionary Access Control