UWE Bristol | University of the West of England

Presented by

Dr. Trupti Padiya

School of Computing and Creative Technologies

Advanced Databases UFCFU3-15-3

Database Security

Slides with modification adapted from Dr. Elias Pimenidis

Introduction to Database Security Issues (1)

Threats to databases

- Loss of integrity
- Loss of availability
- Loss of confidentiality

To protect databases against these types of threats four kinds of countermeasures can be implemented : *access control, inference control, flow control, and encryption*.

Introduction to Database Security Issues (2)

A DBMS typically includes a database security and authorization subsystem that is responsible for ensuring the security portions of a database against unauthorized access.

Two types of database security mechanisms:

- Discretionary security mechanisms
- Mandatory security mechanisms

Introduction to Database Security Issues (3)

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.

Introduction to Database Security Issues (4)

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.

UWE Bristol University of the West of England

Introduction to Database Security Issues (5)

Another security is that of **flow control**, which prevents information from flowing in such a way that it reaches unauthorized users.

Channels that are pathways for information to flow implicitly in ways that violate the security policy of an organization are called **covert channels**.

Introduction to Database Security Issues (6)

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.

The data is **encoded** using some coding 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 data.

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 and classifying users and data in accordance with the policy of the organization.

The DBA has a **DBA account** in the DBMS, sometimes called a **system** or **superuser account**, which provides powerful capabilities.

Database Security and the DBA

1. *Account creation*
2. *Privilege granting*
3. *Privilege revocation*
4. *Security level assignment*

The DBA is responsible for the overall security of the database system.

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

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.

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**.

UWE Bristol University of the West of England

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**.

Types 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.
- The *relation (or table level)*: At this level, the DBA can control the privilege to access each individual relation or view in the database.

Types 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.

Types of Discretionary Privileges(3)

The second level of privileges applies to the **relation level**, whether they are base relations or 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 represent *subjects* (users, accounts, programs) and 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*.

Types of Discretionary Privileges(4)

To control the granting and revoking of relation privileges, each relation R in a database is assigned an **owner account**, which is typically the account that was used when the relation was created in the first place. The owner of a relation is given ***all*** privileges on that relation. In SQL2, the **DBA** can assign an 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.

Types 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.

Types of Discretionary Privileges(6)

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.

Note, that to create a view, the account must have **SELECT** privilege on ***all relations*** involved in the **view definition.**

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.

UWE Bristol University of the West of England

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.

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.

UWE Bristol University of the West of England

Specifying Limits on Propagation of Privileges

Techniques to limit the propagation of privileges have been developed, although they have not yet been implemented in most DBMSs and are not a part of SQL.

Limiting **horizontal propagation** to an integer number *i* means that an account B given the GRANT OPTION can grant the privilege to at most *i* other accounts.

Vertical propagation is more complicated; it limits the depth of the granting of privileges.

UWE Bristol University of the West of England

Mandatory Access Control and Role-Based Access Control for Multilevel Security - 1

The discretionary access control techniques of granting and revoking privileges on relations has traditionally been the main security mechanism for relational database systems.

This is an all-or-nothing method: A user either has or does not have a certain privilege.

In many applications, an ***additional security policy*** is needed that classifies data and users based on security classes. This approach known as **mandatory access control**, would typically be *combined* with the discretionary access control mechanisms.

Mandatory Access Control and Role-Based Access Control for Multilevel Security - 2

Typical **security classes** are top secret (TS), secret (S), confidential (C), and unclassified (U), where TS is the highest level and U the lowest: TS ≥ S ≥ C ≥ U

The commonly used model for multilevel security, known as the Bell-LaPadula model, classifies each **subject** (user, account, program) and **object** (relation, tuple, column, view, operation) into one of the security classifications, T, S, C, or U: **clearance** (classification) of a subject S as **class(S)** and to the **classification** of an object O as **class(O)**.

Mandatory Access Control and Role-Based Access Control for Multilevel Security - 3

Two restrictions are enforced on data access based on the subject/object classifications:

1. A subject S is not allowed read access to an object O unless class(S) ≥ class(O). This is known as the **simple security property**.
2. A subject S is not allowed to write an object O unless class(S) ≤ class(O). This known as the **star property** (or * property).

Mandatory Access Control and Role-Based Access Control for Multilevel Security - 4

To incorporate multilevel security notions into the relational database model, it is common to consider attribute values and tuples as data objects. Hence, each attribute A is associated with a **classification attribute** C in the schema, and each attribute value in a tuple is associated with a corresponding security classification. In addition, in some models, a **tuple classification** attribute TC is added to the relation attributes to provide a classification for each tuple as a whole. Hence, a **multilevel relation** schema R with n attributes would be represented as

$R(A_1,C_1,A_2,C_2, \ldots, A_n,C_n,TC)$

where each C_i represents the classification attribute associated with attribute A_i.

Mandatory Access Control and Role-Based Access Control for Multilevel Security - 5

The value of the TC attribute in each tuple t – which is the *highest* of all attribute classification values within t – provides a general classification for the tuple itself, whereas each C_i provides a finer security classification for each attribute value within the tuple.

The **apparent key** of a multilevel relation is the set of attributes that would have formed the primary key in a regular(single-level) relation.

Mandatory Access Control and Role-Based Access Control for Multilevel Security - 6

A multilevel relation will appear to contain different data to subjects (users) with different clearance levels. In some cases, it is possible to store a single tuple in the relation at a higher classification level and produce the corresponding tuples at a lower-level classification through a process known as **filtering**.

In other cases, it is necessary to store two or more tuples at different classification levels with the same value for the *apparent key*. This leads to the concept of **polyinstantiation** where several tuples can have the same apparent key value but have different attribute values for users at different classification levels.

Mandatory Access Control and Role-Based Access Control for Multilevel Security - 7

In general, the **entity integrity** rule for multilevel relations states that all attributes that are members of the apparent key must not be null and must have the *same* security classification within each individual tuple.

In addition, all other attribute values in the tuple must have a security classification greater than or equal to that of the apparent key. This constraint ensures that a user can see the key if the user is permitted to see any part of the tuple at all.

Mandatory Access Control and Role-Based Access Control for Multilevel Security - 8

Other integrity rules, called **null integrity** and **interinstance integrity**, informally ensure that if a tuple value at some security level can be filtered (derived) from a higher-classified tuple, then it is sufficient to store the higher-classified tuple in the multilevel relation.

UWE Bristol University of the West of England

Comparing Discretionary Access Control and Mandatory Access Control

- Discretionary Access Control (DAC) policies are characterized by a **high degree of flexibility**, which makes them suitable for a large variety of application domains.
- The main drawback of DAC models is their **vulnerability to malicious attacks**, such as Trojan horses embedded in application programs.

Comparing Discretionary Access Control and Mandatory Access Control - 2

- By contrast, mandatory policies ensure a high degree of protection in a way, they **prevent any illegal flow of information.**
- Mandatory policies have the drawback of being **too rigid** and they are only applicable in limited environments.
- **In many practical situations, discretionary policies are preferred because they offer a better trade-off between security and applicability.**

Role-Based Access Control

Role-based access control (RBAC) emerged rapidly in the 1990s as a proven technology for managing and enforcing security in large-scale enterprise-wide systems. Its basic notion is that permissions are associated with roles, and users are assigned to appropriate roles.

Roles can be created using the CREATE ROLE and DESTROY ROLE commands. The GRANT and REVOKE commands discussed under DAC can then be used to assign and revoke privileges from roles.

UWE Bristol University of the West of England

Introduction to Statistical Database Security

- Statistical databases are used mainly to produce statistics on various populations.
- The database may contain confidential data on individuals, which should be protected from user access.
- Users are permitted to retrieve statistical information on the populations, such as averages, sums, counts, maximums, minimums, and standard deviations.

UWE Bristol University of the West of England

Introduction to Statistical Database Security - 2

- A **population** is a set of tuples of a relation (table) that satisfy some selection condition.
- Statistical queries involve applying statistical functions to a population of tuples.

Introduction to Statistical Database Security - 3

For example, we may want to retrieve the number of individuals in a population or the average income in the population. However, statistical users are not allowed to retrieve individual data, such as the income of a specific person. Statistical database security techniques must prohibit the retrieval of individual data.

This can be achieved by prohibiting queries that retrieve attribute values and by allowing only queries that involve statistical aggregate functions such as COUNT, SUM, MIN, MAX, AVERAGE, and STANDARD DEVIATION. Such queries are sometimes called **statistical queries**.

Introduction to Statistical Database Security - 4

- It is DBMS's responsibility to ensure confidentiality of information about individuals, while still providing useful statistical summaries of data about those individuals to users.
- Provision of **privacy protection** of users in a statistical database is paramount.
- In some cases it is possible to **infer** the values of individual tuples from a sequence statistical queries. This is particularly true when the conditions result in a population consisting of a small number of tuples.

Flow Control

- Flow control regulates the distribution or flow of information among accessible objects. A flow between object X and object Y occurs when a program reads values from X and writes values into Y.
- Flow controls check that information contained in some objects does not flow explicitly or implicitly into less protected objects.
- A flow policy specifies the channels along which information is allowed to move. The simplest flow policy specifies just two classes of information: **confidential (C)** and **nonconfidential (N)**, and allows **all flows except** those **from class C to class N**.

Encryption and Public Key Infrastructures

- Encryption is a means of maintaining secure data in an insecure environment.
- Encryption consists of applying an **encryption algorithm** to data using some prespecified **encryption key**.
  - the resulting data has to be **decrypted** using a **decryption key** to recover the original data.

The Data and Advanced Encryption Standards

- The **Data Encryption Standard (DES)** is a system developed by the U.S. government for use by the general public. It has been widely accepted as a cryptographic standard both in the United States and the rest of the world.
- DES can provide **end-to-end encryption** on the channel between the sender A and receiver B.

Extra Resources

- https://www.w3schools.in/dbms/database-security
- https://www.ibm.com/think/topics/database-security
- https://www.ibm.com/docs/en/db2-for-zos/13?topic=access-use-grant-revoke-privileges-control