CHAPTER - 6

SECURITY

6.1 Needs of Security

Security refers to the protection from malicious attempts to steal or modify data. The need for database security is given below:

- In the case of shared data, multiple users try to access the data at the same time. In order to maintain the consistency of the data in the database, database security is needed.
- Due to the advancement of internet, data are accessed through World Wide Web, to protect the data against hackers, database security is needed.
- The plastic money (Credit card) is more popular. The money transaction has to be safe. More specialized software both to enter the system illegally to extract data and to analyze the information obtained is available.

Hence, it is necessary to protect the data/money

Security in different level are as follows:

i. **Database System Level**

In database system, authentication and authorization mechanisms can be implemented to allow specific users access only to required data

ii. **Operating System Level**

Operating system super-users can do anything they want to the database so good operating system level security is required.

iii. **Network Level:**

Network must use encryption to prevent eavesdropping (unauthorized reading of messages),Masquerading (pretending to be an authorized user or sending messages supposedly from authorized users)

iv. **Physical level**

Physical access to computers allows destruction of data by intruders; traditional lock-and-key security is needed. Computers must also be protected from floods, fire, etc.

v. **Human level**

Users must be screened to ensure that an authorized users do not give access to intruders. Users should be trained on password selection and secrecy

6.2 Security and Integrity Violations

Security and integrity violations in a database can occur due to various reasons such as human error, software bugs, hardware failure, malicious attacks, or natural disasters. These violations can result in data breaches, data corruption, data loss, or unauthorized access to sensitive data.

Security violations can occur when unauthorized users gain access to sensitive data or when legitimate users perform unauthorized actions on the database. For example, a user might share their login credentials with someone else, or a hacker might exploit a vulnerability in the database software to gain access to sensitive data. To prevent security violations, access to the database should be controlled using strong authentication mechanisms, access control policies, and encryption of sensitive data.

Integrity violations can occur when data is modified or deleted in an unauthorized manner or when data becomes corrupted due to software bugs or hardware failures. For example, a user might accidentally delete an important record from the database, or a software bug might cause data to become corrupted. To prevent integrity violations, access to the database should be controlled using access control policies, and data should be backed up regularly to ensure that it can be recovered in case of data loss or corruption.

To detect security and integrity violations, various techniques such as auditing, monitoring, and logging can be used. These techniques enable organizations to track and analyze

activities performed on the database and detect any suspicious behavior or unusual activity that might indicate a security or integrity violation.

6.3 Access Control

The practice of regulating and limiting access to specific resources, such as data, information, physical areas, or digital systems, is referred to as access control. This is accomplished by combining policies, processes, and technology that manage who is permitted access to particular resources, when they can access them, and what actions they are permitted to execute.

Access control can be implemented in a variety of ways, ranging from simple password-based systems to more complex biometric authentication and multi-factor authentication systems. Some common access control techniques include:

Discretionary Access Control (DAC)

A more flexible approach that allows users to determine who can access their resources. Typically, privilege granting and revocation are used to enforce discretionary access control in database systems. SQL supports discretionary access control through the GRANT and REVOKE commands. Creator of a table or a view automatically gets all privileges on it, along with the right togrant these privileges to other users. The GRANT command gives users privileges to base tables and views. The syntax of this command is as follows:

GRANT privileges ON object TO users [WITH GRANT OPTION]

For our purposes object is either a base table or a view. Several privileges can be specified, including these:

- **SELECT:** The right to access (read) all columns of the table specified as the object, including columns added later through ALTER TABLE commands.
- **INSERT(column-name):** The right to insert rows with (non-null or nondefault) values in the named column of the table named as object. If this right is to be granted with respect to

all columns, including columns that might be added later, we can simply use INSERT. The privileges UPDATE(column-name) and UPDATE are similar.

- **DELETE:** The right to delete rows from the table named as object.
- **REFERENCES (column-name):** The right to define foreign keys (in other tables) that refer to the specified column of the table object. REFERENCES without a column name specified denotes this right with respect to all columns, including any that are added later.

If a user has a privilege with the grant option, he or she can pass it to another user (with or without the grant option) by using the GRANT command. A user who creates a view has precisely those privileges on the view that he or she has on every one of the view or base tables used to define the view. The user creating the view must have the SELECT privilege on each underlying table, of course, and so is always granted the SELECT privilege on the view. The creator of the view has the SELECT privilege with the grant option only if he or she has the SELECT privilege with the grant option on every underlying table. In addition, if the view is updatable and the user holds INSERT, DELETE, or UPDATE privileges (with or without the grant option) on the (single) underlying table, the user automatically gets the same privileges on the view.

Only the owner of a schema can execute the data definition statements CREATE, ALTER, and DROP on that schema. The right to execute these statements cannot be granted or revoked.

Suppose that user Joe has created the tables Boats, Reserves, and Sailors. Some examples of the GRANT command that Joe can now execute are listed below:

```
GRANT INSERT, DELETE ON Reserves TO Yuppy WITH GRANT OPTION
GRANT SELECT ON Reserves TO Michael
GRANT SELECT ON Sailors TO Michael WITH GRANT OPTION
GRANT UPDATE (rating) ON Sailors TO Leah
GRANT REFERENCES(bid) ON Boats TO Bill
```


When a privilege is revoked from X, it is also revoked from all users who got it solely from X. The revoke statement is used to revoke authorization.

revoke<privilege list> on <relation name or view name> from <user list>

Example:

revoke select on branch from U1, U2, U3

Mandatory Access Control (MAC)

A more rigid approach that is often used in high-security environments, such as military or government installations that classifies data and users based on security classes. This approach, known as mandatory access control (MAC), would typically be combined with the discretionary access control mechanisms.

Typical security classes are top secret (TS), secret (S), confidential (C), and unclassified(U), where TS is the highest level and U the lowest. Other more complex security classification schemes exist, in which the security classes are organized in a lattice. For simplicity, we will use the system with four security classification levels, where TS ≥ S ≥ C ≥ U, to illustrate our discussion. 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 TS, S, C, or U. We will refer to the clearance (classification) of a subject S as class(S) and to the classification of an object O as class(O).

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

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

If discretionary access controls are also specified, these rules represent additional restrictions. Thus, to read or write a database object, a user must have the necessary privileges (obtained via

6.4 Authorization and Authentication

6.4.1 Authorization

In a database, authorization is the process of approving or revoking access to data depending on the rights and privileges given to a user or position. It is a crucial component of database security since it makes sure that only people with the proper permissions may access the information. Different forms of authorization on parts of the database are:

1. **Read Authorization**

   Read authorization refers to the permission granted to a user or role to read or view data stored in a database. This permission allows the user to access and retrieve data from the database but does not allow them to modify or delete it

2. **Insert Authorization**

   Insert authorization refers to the permission granted to a user or role to insert new data into a database. This permission allows the user to add new records or data entries to the database but not modification of existing data.

3. **Update Authorization**

   Update authorization refers to the permission granted to a user or role to modify existing data in a database. This permission allows the user to change or update the values of existing records or data entries in the database, but not deletion of data.

4. **Delete Authorization**

   Delete authorization refers to the permission granted to a user or role to remove existing data from a database. This permission allows the user to delete records or data entries from the database.

   Users can be given authorization on views, without being given any authorization on the relations used in the view

6.4.2 Authentication

Authentication is the process of verifying the identity of a user or system before allowing access to a resource or service. It is an essential security measure used to protect sensitive information and ensure that only authorized individuals or entities can access it.

Authentication involves the use of credentials, such as usernames and passwords, security tokens, biometric data, or other authentication factors to verify the identity of the user or system. The authentication process can be performed by various means, such as a login form, a security token, or a biometric scanner.

6.5 Security and Views

Views in a database can play a significant role in enhancing security. A view is a virtual table that is based on the result of a database query. It allows a user to see a specific subset of data without granting them access to the underlying tables or data.

Views can be used to implement security policies in a database by restricting access to sensitive data. For example, a view can be created that includes only the columns of a table that a user is authorized to see, while hiding the rest of the columns. This can prevent unauthorized users from accessing sensitive data that they are not authorized to view.

Furthermore, views can be used to control the level of data access granted to different users or roles. Access can be restricted by creating views that limit the data that a user or role can see, while preventing them from accessing other data in the underlying tables. This approach can help prevent security breaches, data leakage, or data theft.

GRANT commands) and the security classes of the user and the object must satisfy the preceding restrictions

6.4 Authorization and Authentication

6.4.1 Authorization

In a database, authorization is the process of approving or rejecting access to data depending on the rights and privileges given to a user or position. It is a crucial component of database security since it makes sure that only people with the proper permissions may access the information. Different Forms of authorization on parts of the database are:

1. **Read Authorization**

   Read authorization refers to the permission granted to a user or role to read or view data stored in a database. This permission allows the user to access and retrieve data from the database but does not allow them to modify or delete it

2. **Insert Authorization**

   Insert authorization refers to the permission granted to a user or role to insert new data into a database. This permission allows the user to add new records or data entries to the database but not modification of existing data.

3. **Update Authorization**

   Update authorization refers to the permission granted to a user or role to modify existing data in a database. This permission allows the user to change or update the values of existing records or data entries in the database, but not deletion of data.

4. **Delete Authorization**

   Delete authorization refers to the permission granted to a user or role to remove existing data from a database. This permission allows the user to delete records or data entries from the database.

Users can be given authorization on views, without being given any authorization on the relations used in the view

definition. Ability of views to hide data serves both to simplify usage of the system and to enhance security by allowing users access only to data they need for their job. A combination or relational-level security and view-level security can be used to limit a user's access to precisely the data that user needs.

6.4.2 Authentication

Authentication is the process of verifying the identity of a user or system before allowing access to a resource or service. It is an essential security measure used to protect sensitive information and ensure that only authorized individuals or entities can access it.

Authentication involves the use of credentials, such as usernames and passwords, security tokens, biometric data, or other authentication factors to verify the identity of the user or system. The authentication process can be performed by various means, such as a login form, a security token, or a biometric scanner.

6.5 Security and Views

Views in a database can play a significant role in enhancing security. A view is a virtual table that is based on the result of a database query. It allows a user to see a specific subset of data without granting them access to the underlying tables or data.

Views can be used to implement security policies in a database by restricting access to sensitive data. For example, a view can be created that includes only the columns of a table that a user is authorized to see, while hiding the rest of the columns. This can prevent unauthorized users from accessing sensitive data that they are not authorized to view.

Furthermore, views can be used to control the level of data access granted to different users or roles. Access can be restricted by creating views that limit the data that a user or role can see, while preventing them from accessing other data in the underlying tables. This approach can help prevent security breaches, data leakage, or data theft.

Views can also be used to implement row-level security policies in a database. A view can be created that includes only the rows of a table that a user is authorized to see, while hiding the rest of the rows. This approach is useful for implementing fine-grained access control policies that limit access to specific rows of data.

6.6 Encryption and Decryption

Despite being effective control measures, access and flow methods may not be able to completely safeguard databases from all dangers. Let's say we share data, but someone who shouldn't have access to it does. In this instance, by utilizing encryption, we may mask the message and prevent its disclosure even if the transmission is interrupted. Data is transformed during encryption into a ciphertext format that is difficult for unauthorized parties to understand. It enhances security and privacy when access controls are bypassed, because in cases of data loss or theft, encrypted data cannot be easily understood by unauthorized person.

The Data Encryption Standard (DES)

The American government created the DES system for use by the general people. It has received widespread acceptance in the United States as well as abroad as a cryptographic standard. On the communication channel between sender A and receiver B, DES can offer end-to-end encryption. The DES algorithm is a careful and complex combination of two of the fundamental building blocks of encryption: substitution and permutation (transposition).The algorithm's strength comes from repeatedly using these two methods over the course of 16 cycles. The message's original plaintext is encoded in blocks of 64 bits. Despite being 64 bits long, the key can actually be any 56-bit number.

The Advanced Encryption Standard (AES)

AES was introduced by the NIST after DES' suitability was questioned. In contrast to DES, which only supports keys with a size of 56 bits, this method can employ keys with a size of 128 bits, 192 bits, or 256 bits. AES introduces more key combinations than DES, which makes it more difficult to break.

Symmetric Key Algorithms

One key is used for both encryption and decryption; this key is known as a symmetric key. For routine use with sensitive data in the database, quick encryption and decryption is achievable by employing a symmetric key. Only the same secret key will work to decrypt a message that has been encrypted using it. Secret-key algorithms are those used for symmetric key encryption. Secret-key algorithms are sometimes known as content encryption algorithms because they are frequently used to encrypt the content of messages.

The necessity of sharing the secret key is the main drawback of secret-key algorithms. A password-based encryption technique is one approach that might be used to obtain the secret key from a user-supplied password string by applying the same function to the string at both the sender and receiver. The size of the key used affects how strong the symmetric key encryption is. For the same method, having a longer key makes encryption more difficult to crack than using a shorter key.

Public Key Encryption

In 1976, Diffie and Hellman proposed a cryptosystem, which they called public key encryption. Instead of relying on operations on bit patterns, public key algorithms use mathematical functions as their foundation. They address one issue with symmetric key encryption, namely the need for secure key exchange between sender and recipient. Two keys are utilized for encryption and decryption in public key systems. While the secret key is never communicated, the public key might be done so in an insecure manner. Asymmetric key encryption algorithms are those that carry out complementary operations (encryption and decryption) using two linked keys, a public key and a private key. Regarding key distribution, confidentiality, and authentication, the use of two keys can have significant repercussions. The two keys used for public key encryption are referred to as the public key and the private key. The private key is kept secret, but it is referred to as a

private key rather than a secret key (the key used in conventional encryption) to avoid confusion with conventional encryption. The two keys are mathematically related, since one of the keys is used to perform encryption and the other to perform decryption. However, it is very difficult to derive the private key from the public key.

Six components make up a public key infrastructure or scheme:

1. **Plaintext**

   This is the data or readable message that is fed into the algorithm as input.

2. **Encryption Algorithm**

   This algorithm performs various transformations on the plaintext.

3. **and 4. Public and Private Keys**

   These are a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the encryption algorithm depend on the public or private key that is provided as input. For example, if a message is encrypted using the public key, it can only be decrypted using the private key.

5. **Ciphertext**

   This is the scrambled message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different ciphertexts.

6. **Decryption Algorithm**

   This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

As the name suggests, the public key of the pair is made public for others to use, whereas the private key is known only to its owner. A general-purpose public key cryptographic algorithm relies on one key for encryption and a different but related key for decryption. The essential steps are as follows:

SOLUTION TO EXAMS' AND OTHER IMPORTANT QUESTIONS

1. Differentiate between authorization and authentication with brief examples. [2022 Fall, 2017 Fall, 2014 Fall]

ANS: Authentication is the process of verifying the identity of a user or system before allowing access to a resource or service. Authentication involves the use of credentials, such as usernames and passwords, security tokens, biometric data, or other authentication factors to verify the identity of the user or system.

Example:

When we log in to our email account, we are asked to provide our username and password, and the system verifies that the information we provided matches the stored credentials for your account.

Authorization, on the other hand, is the process of granting or denying access to a resource or service based on the authenticated user's permissions or privileges. Authorization determines what a user is allowed to access or perform after they have been authenticated. It involves setting up access controls, permissions, and policies to ensure that only authorized individuals or entities can access the resource or service.

Example:

After we log in to our email account, we may have authorization to read, compose, and send emails, but we may not have authorization to delete emails or change settings.

2. Explain the need of access control, Authorization and Authentication. [2021 Fall,2019 Spring]

ANS: Access control is necessary because not all users need the same level of access to a resource or system. For example, in an organization, employees may have different job roles and responsibilities, and therefore, they require different levels of access to company resources, such as files, folders, databases,

and applications. Without the right access control procedures in place, unauthorized individuals could access confidential data, change or delete data, and seriously harm the firm.

Access control is essential for several reasons:

i. **Protecting Sensitive Information**

Access control ensures that sensitive information is only accessible to authorized users who have a legitimate need to know. This helps to prevent data breaches, theft, and other malicious activities.

ii. **Maintaining Confidentiality**

Access control mechanisms ensure that confidential information is kept confidential by preventing unauthorized users from accessing it.

iii. **Enforcing Compliance**

Access control mechanisms can help organizations comply with regulatory requirements by ensuring that only authorized users have access to sensitive data.

iv. **Preventing Insider Threats**

Access control mechanisms can help prevent insider threats by limiting the access of employees to only the resources they need to do their job.

v. **Enhancing System Availability**

Access control mechanisms can help prevent denial-of-service attacks by limiting access to system resources.

Authentication refers to the process of verifying the identity of a user, device, or application. It involves the use of various authentication methods, such as passwords, biometrics, smart cards, and tokens, to ensure that only authorized users are granted access to sensitive information and resources. Authentication is necessary because it ensures that only legitimate users can access a system, and unauthorized users are denied access.

Authorization refers to the process of determining what actions an authenticated user is allowed to perform. It involves granting or denying access to resources based on the

user's identity, privileges, and permissions. Authorization is necessary because not all users should have the same level of access to a resource or system, and unauthorized access can lead to data breaches, theft, and other malicious activities.

3. List all security issues you are familiar that can breaches security. Encryption and Decryption are very important part in cryptography so justify this statement with suitable example. [2020 Spring]

ANS: Security issues that can lead to a breach of security are:

i. Weak Passwords

Weak passwords or password reuse across multiple accounts can make it easy for attackers to gain unauthorized access.

ii. Social Engineering

Social engineering involves tricking people into disclosing sensitive information or performing an action that can lead to a security breach.

iii. Phishing

Phishing is a type of social engineering where attackers send fraudulent emails or messages to trick users into providing sensitive information.

iv. Malware

Malware, such as viruses, Trojans, and spyware, can infect a system and steal sensitive information or cause damage.

v. Denial-of-Service Attacks

Denial-of-service attacks involve overwhelming a system with traffic or requests to make it unavailable to legitimate users.

vi. Insider Threats

Insider threats involve employees or other trusted individuals who misuse their access to sensitive information or systems.

vii. Physical Security Breaches

Physical security breaches involve unauthorized access to physical assets, such as buildings, servers, or data centers.

viii. Unpatched Software Vulnerabilities

Unpatched software vulnerabilities can be exploited by attackers to gain unauthorized access to systems or steal sensitive information.

ix. Third-Party Risks

Third-party risks involve vulnerabilities in software or services provided by third-party vendors or partners.

x. Internet of Things (IoT) Devices

IoT devices can be vulnerable to attacks that compromise the security of a network or system.

The process of converting plaintext into ciphertext, a form of data that is unreadable and unintelligible to anyone who does not have the key to decrypt it, is known as encryption. Decryption is the reverse process of converting ciphertext back into plaintext.

For modern communication systems to guarantee confidentiality and privacy, encryption and decryption are necessary. Without encryption, unauthorized third parties might intercept and access sensitive information including credit card numbers, login credentials, and medical records.

For example, let's say we want to send a sensitive email message to a friend. We could use encryption to protect the contents of the message from being intercepted and read by anyone who does not have the key to decrypt it.

Here's how the process might work:

i. We use an encryption program to encrypt the plaintext message using a key.

ii. The encryption program transforms the plaintext message into ciphertext, which is unreadable and unintelligible without the key.

iii. We send the ciphertext message to our friend via email or another communication channel.

iv. Our friend receives the ciphertext message and uses a decryption program to decrypt the message using the key.

v. The decryption program transforms the ciphertext message back into plaintext, which our friend can now read.

Without encryption, the plaintext message would be vulnerable to interception and reading by unauthorized third parties, which could compromise the confidentiality of the message and lead to various security issues. However, by using encryption and decryption, the message remains secure and private, even if it is intercepted by an attacker.

4. Why ACL technique is considered safe way for database security? How is any user allowed or prevented from accessing a certain resources? [2017 Fall]

ANS: Due to its capacity to offer smooth control over resource access, the ACL (Access Control List) technique is regarded as a secure method for database security. Administrators can set and maintain access permissions for specific people or groups using ACLs, ensuring that only authorized parties can access a database's resources. Here is a more thorough description of how ACLs are used to permit or restrict user access to particular resources:

i. Resource Access Control

Every database resource, including tables, views, stored procedures, and columns, is accompanied by an access control list (ACL) that contains a list of access control entries (ACEs). The privileges or permissions assigned to specific users or groups for that particular resource are described in ACEs.

ii. User Authentication

Users must identify themselves by supplying legitimate credentials, such as usernames and passwords, before they may access the database. By confirming their identity, user authentication makes sure that only authorized users can continue.

iii. ACL Evaluation

The corresponding ACL is consulted whenever a user tries to access a particular resource. If there is a match, the system checks the user's credentials against the ACEs in the ACL.

iv. Permission Verification

The system checks to see if the user has the required permissions to carry out the requested operation on the resource if the user's credentials match an ACE in the ACL. Read, write, update, delete, execute, and other special operations could be included in the permissions.

v. Access Granting

Access to the resource is granted if the user's credentials and the intended action comply with the permissions outlined in the ACE. The requested operation—such as retrieving data, editing records, or running queries—can be carried out by the user.

vi. Access Denial

Access to the resource is prohibited if the user's credentials do not match any ACE in the ACL or if the intended activity exceeds the permitted scope. The user is unable to use the resource or carry out the unlawful behavior.

vii. Access Control Administration

The management and upkeep of the ACLs is the duty of the database administrators or other authorized individuals. To make sure that access permissions appropriately represent the security policies and requirements of the organization, they create, alter, or remove ACEs as necessary.

Database systems achieve a strong security foundation by using ACLs. Administrators can accurately define which individuals or groups have access to particular resources

and what actions they can take thanks to ACLs, which offer a granular level of control. The possibility of unauthorized access, data breaches, or harmful actions within the database is reduced thanks to this fine-grained access control. It lets companies to implement the principle of least privilege, limiting user access to sensitive information or crucial processes while providing users only the rights necessary to carry out their responsibilities.

5. What are the needs of cryptography? [2014 Spring]

ANS: Using a format that is incomprehensible to unauthorized parties, cryptography is the process of safeguarding communication and information. Data is encrypted (scrambled) and then decrypted (unscrambled) using mathematical techniques and algorithms. Data confidentiality, integrity, authentication, and non-repudiation are all goals of cryptography.

Cryptography is needed for various reasons, including:

i. **Confidentiality:** By encrypting data, it makes sure that private information stays that way and is inaccessible to unauthorized people.

ii. **Integrity:** Cryptography ensures the integrity and reliability of data by confirming that it was not changed or tampered with during transmission or storage.

iii. **Authentication**: It aids in establishing the legitimacy of people, systems, or equipment in order to avoid impersonation and unwanted access.

iv. **Non-Repudiation:** Through the use of cryptography, parties can demonstrate their participation in a transaction or the validity of their digital signatures, preventing action denial.

v. **Key Management:** Secure cryptographic key management is necessary for encryption, decryption, and other processes.