Department of CSE

COURSE NAME: DBMS
COURSE CODE:23AD2102R

Topic: Security Considerations in DBMS, Access Control, Encryption, Intrusion Detection Systems(IDS)

Session – 5

AIM OF THE SESSION

To familiarize students with the basic concepts of Security considerations of DBMS

INSTRUCTIONAL OBJECTIVES

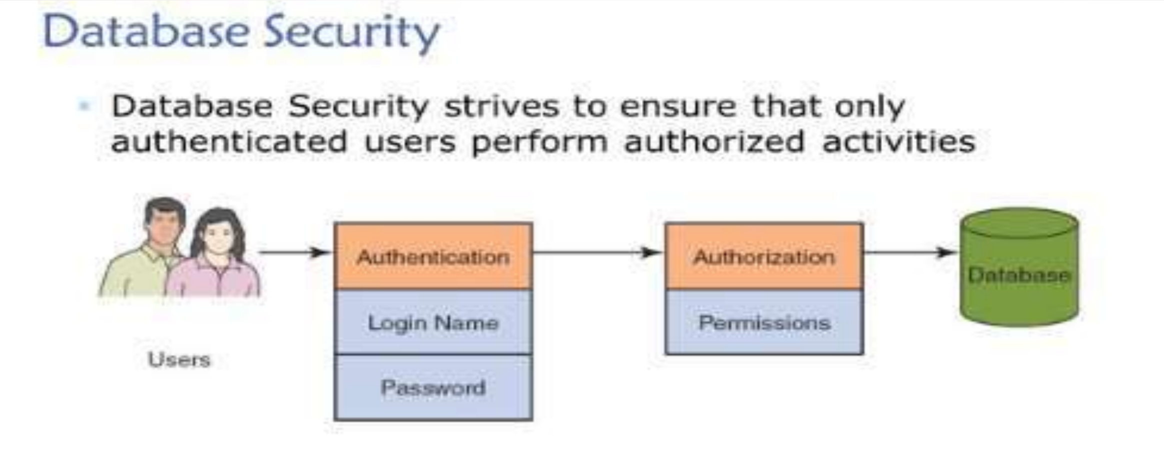
This Session is designed to: discuss and study the concepts of Security considerations of DBMS

LEARNING OUTCOMES

At the end of this session, you should be able to: understand Security considerations of DBMS

SECURITY CONSIDERATIONS IN DBMS

Security considerations in Database Management Systems (DBMS) are critical to protecting data from **unauthorized access** and **malicious attacks**. Here's a detailed look at key security measures, including examples that illustrate their importance.



Database Security

- Database Security strives to ensure that only authenticated users perform authorized activities

1. Access Control

- **Description:** Access control defines who has permission to access the data, including various levels like read, write, or update permissions. Access control typically leverages roles to assign specific privileges, ensuring that users only access data they need.
- **Example:** In a financial institution's DBMS, employees in the "Manager" role may have access to financial reports, while "Customer Service" employees have limited access to basic customer data. This control ensures sensitive data isn't exposed to unauthorized users.

2. Authentication and Authorization

- **Description**: Authentication verifies a user's identity, and authorization ensures the user has permission to access certain data. Techniques include multi-factor authentication (MFA), passwords, and single sign-on (SSO).
- **Example**: An e-commerce platform's DBMS may use MFA for admin logins, requiring both a password and a one-time code sent to the admin's phone. This reduces the risk of unauthorized access if the password is compromised.

3. Encryption

- **Description**: Encryption converts data into a coded form that unauthorized users cannot easily understand. Data can be encrypted both at rest (stored data) and in transit (data being transmitted).

- **Example**: A healthcare database encrypts patient records both when stored on the server and when sent to another location. If an attacker intercepts the data in transit, it would be unreadable without the decryption key, ensuring data privacy.

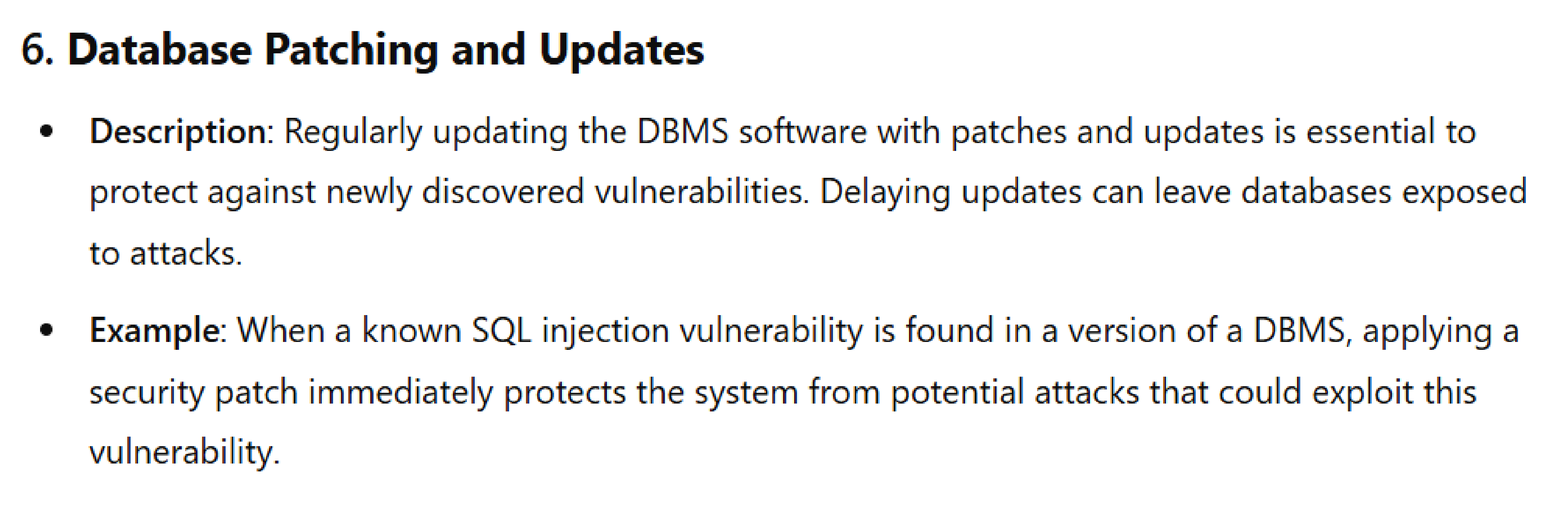
4. Data Masking and Tokenization

- **Description**: Data masking hides sensitive data by replacing it with non-sensitive information, while tokenization replaces data with unique identification symbols, or tokens.

- **Example**: When displaying customer records in a bank's DBMS, credit card numbers may be masked (e.g., showing only the last four digits: "**** **** **** 1234") to protect customer information in case of a data leak.

5. Auditing and Monitoring

- **Description**: Regular auditing and real-time monitoring can track database activities, flagging unusual access patterns or unauthorized activities. This helps in early detection of security breaches.
- **Example**: In a retail company's DBMS, auditing logs can show if someone accessed a high number of customer records in a short period, triggering alerts to investigate if the access was legitimate.



6. Database Patching and Updates

- **Description**: Regularly updating the DBMS software with patches and updates is essential to protect against newly discovered vulnerabilities. Delaying updates can leave databases exposed to attacks.
- **Example**: When a known SQL injection vulnerability is found in a version of a DBMS, applying a security patch immediately protects the system from potential attacks that could exploit this vulnerability.

7. Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS)

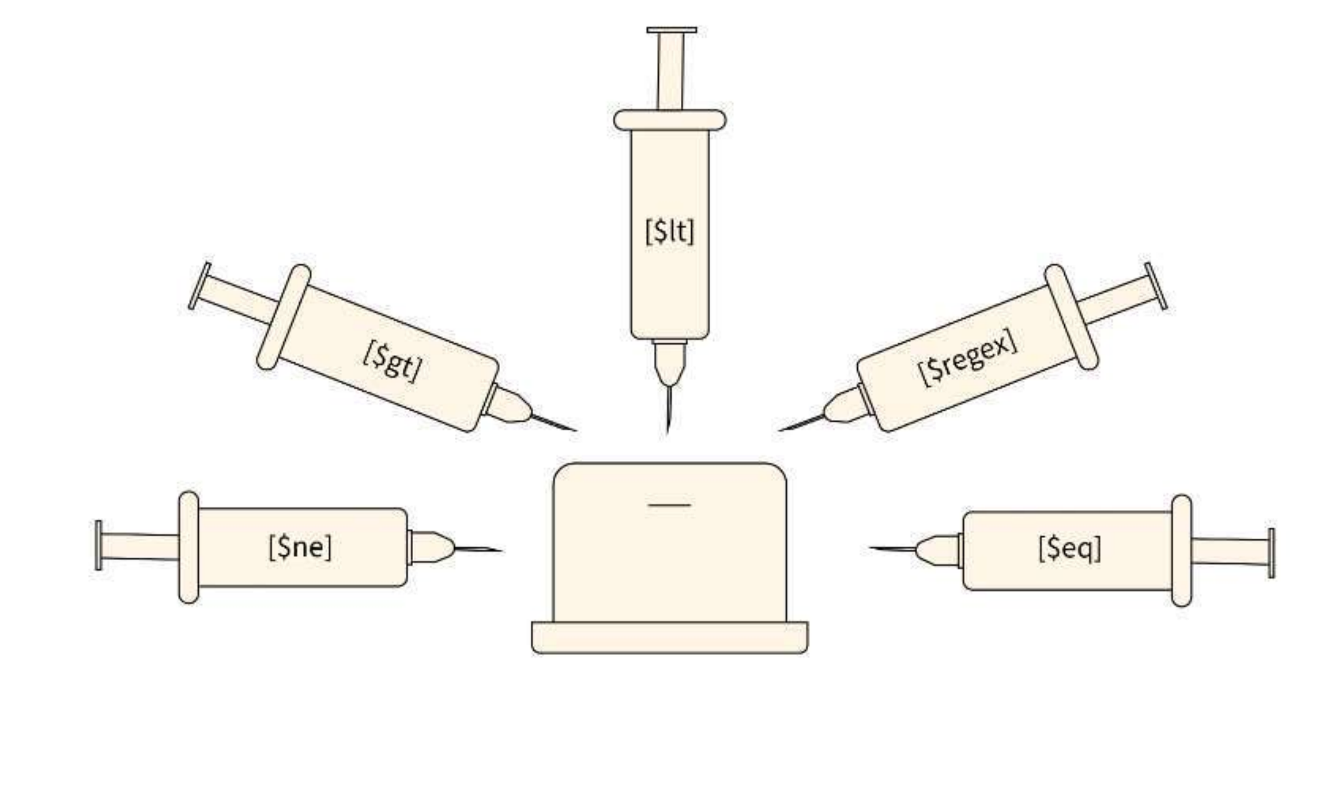
- **Description**: IDS monitors for suspicious activities, while IPS actively blocks potential attacks. Both systems are crucial for protecting the database from external and internal threats.
- **Example**: A social media company's DBMS may use an IDS to monitor unusual data access patterns, such as an admin accessing sensitive user data at odd hours, which could indicate a compromised account.

8. Data Integrity Checks and Consistency Mechanisms

- **Description:** Ensuring data integrity prevents unauthorized modifications and corruption of data. Techniques include checksums, constraints, and transaction logging.
- **Example:** A banking DBMS may use integrity constraints to prevent negative account balances and log all transactions. This prevents unauthorized users from making malicious changes that could compromise data reliability.

9. SQL Injection Prevention

- **Description**: SQL injection is a common attack where malicious SQL code is inserted into a query. This can be prevented by using prepared statements, input validation, and escaping special characters.
- **Example**: In an online booking system, using prepared statements prevents attackers from injecting harmful SQL code, such as "DROP TABLE users," which could delete the users table if input isn't validated.


[$lt]
[$gt]
[$regex]
[$ne]
[$eq]

10. Backup and Disaster Recovery Planning

- **Description:** Regular backups and a disaster recovery plan ensure data availability and integrity in case of system failure, natural disasters, or attacks.
- **Example:** An insurance company's DBMS performs regular backups and stores them at an offsite location. In the event of a ransomware attack, they can restore data from a backup without paying the ransom.

Best Practices for Database Security

1. **Follow the Principle of Least Privilege (PoLP)**: Grant users the minimum level of access necessary.
2. **Implement Multi-Factor Authentication (MFA)**: Especially for users with elevated privileges.
3. **Use Strong Encryption Protocols**: Implement robust encryption standards like AES-256.
4. **Conduct Regular Security Audits**: Identify and address vulnerabilities proactively.
5. **Monitor Database Activity Continuously**: Set up real-time alerts for unusual access patterns.

1. Data Security

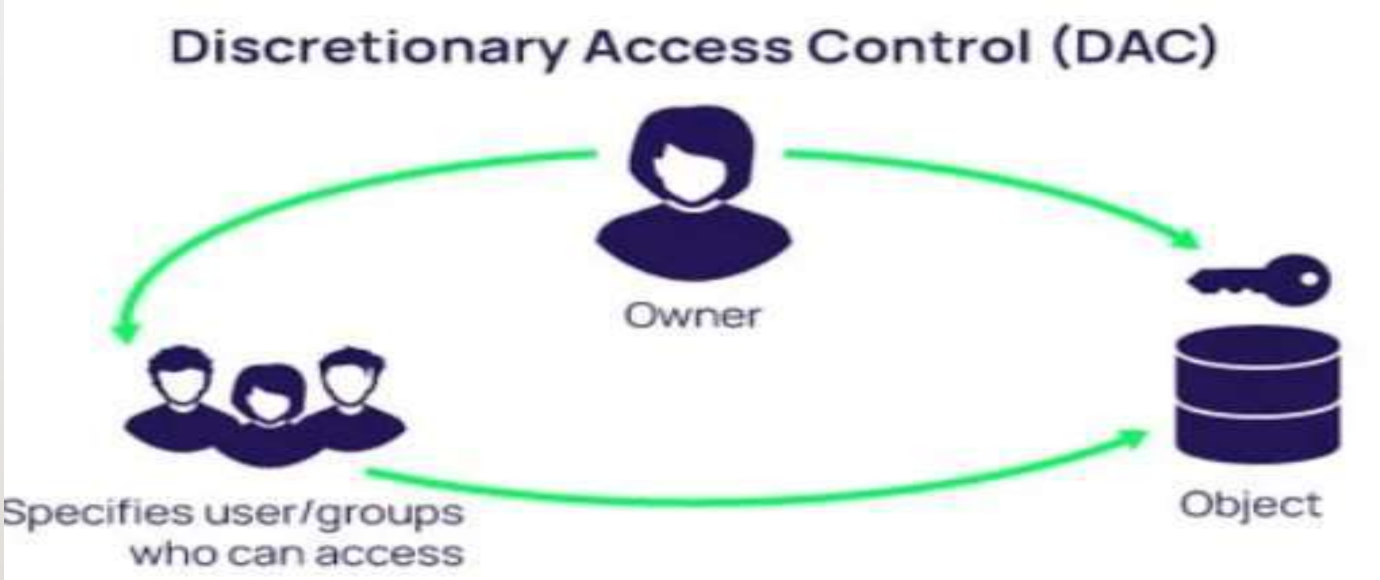
Data security refers to measures and practices designed to protect data from unauthorized access, corruption, or theft throughout its lifecycle. This can include encryption, access controls, and physical security measures. The goal is to ensure data confidentiality, integrity, and availability.

Example: In a banking application, data security ensures that customer information, such as account numbers, balance, and personal details, is encrypted and accessible only by authorized users. For instance, account data might be encrypted at rest and in transit, and access to it might require multifactor authentication.

2. Discretionary Access Control (DAC)

Discretionary Access Control allows data owners to set policies on who can access their resources. In DAC, access rights are granted based on the identity and the discretion of the data owner. Users can pass on their access rights to others, which makes it flexible but potentially less secure.

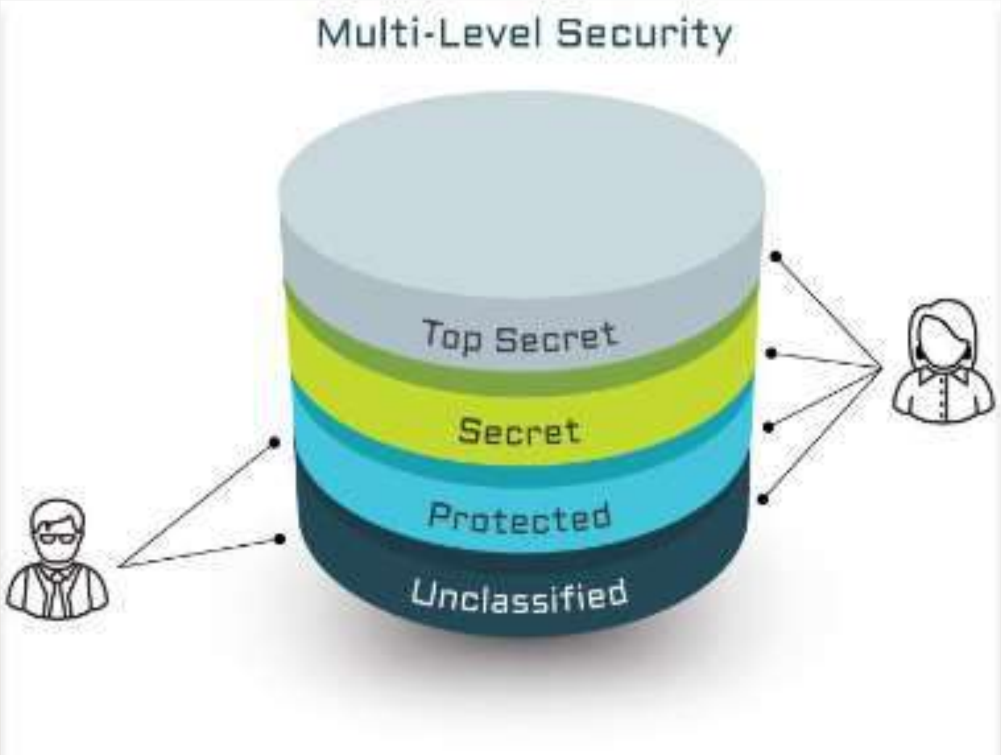
Example: In a shared folder within a company's file server, the folder owner (e.g., a team leader) can set permissions for other team members to access the folder. The team leader might allow specific users to view or edit files in that folder. However, if a user with access decides to share the folder further, they can do so, which could expose sensitive data unintentionally.


Discretionary Access Control (DAC)
Owner
Object
Specifies user/groups who can access

3. Multilevel Access Control (MAC)

Multilevel Access Control (or Mandatory Access Control, MAC) restricts access based on security levels, classifications, or labels. Users and resources are assigned security labels, and access is allowed only when a user’s security level matches or exceeds the resource level. This is commonly used in environments with strict security requirements, like government or military institutions.

Example: In a military database, files might be labeled as "Top Secret," "Secret," or "Confidential." A user with a “Top Secret” clearance can access all files, while a user with “Secret” clearance cannot access “Top Secret” files. Access is strictly managed based on predefined rules and cannot be altered by users, ensuring sensitive data is tightly controlled.


Multi-Level Security
Top Secret
Secret
Protected
Unclassified

4. Distributed Access Control

Distributed Access Control applies when data is stored and managed across multiple, often geographically distributed systems. It allows access to resources based on policies that are enforced across all locations, enabling secure and efficient management in distributed systems.

Example: Consider a global corporation with offices in multiple countries, each with its own data center. An employee may have access to files in their country's data center but will require additional permissions to access resources in data centers in other countries. Access requests are managed centrally but applied across all distributed data centers, ensuring consistent security policies regardless of location.

Encryption

- **Definition:** Encryption is the process of converting plaintext into ciphertext using a cryptographic algorithm and an encryption key. This process ensures that only those with the decryption key can access the original information.
- **Purpose:** To protect sensitive information during transmission or storage, making it unreadable to unauthorized parties.

Decryption

- **Definition:** Decryption is the process of converting ciphertext back into plaintext using a decryption key. It’s the reverse of encryption, making the information accessible in its original, readable form.
- **Purpose:** To allow authorized parties to read the encrypted data by reversing the encryption process.