Parul® University

NAAC A++ ACCREDITED UNIVERSITY

PARUL INSTITUTE OF ENGINEERING & TECHNOLOGY
FACULTY OF ENGINEERING & TECHNOLOGY
PARUL UNIVERSITY

Database Management System

Unit 8 - Security

Computer Science & Engineering

Dr. Vishwanath (Asst. Prof. PIET-CSE)

Parul® University
NAAC A++ ACCREDITED UNIVERSITY

DIGITAL LEARNING CONTENT

Unit 8 - Security

- Data Security
- Data Integrity
- Authentication and Authorization
- Encryption and Decryption
- Access Control (DAC, RBAC, MAC)
- Intrusion Detection
- SQL Injection

Data Security

Data security in DBMS (Database Management System) refers to the practices and technologies implemented to protect the confidentiality, integrity, and availability of data stored within a database.

Here's a breakdown of these key aspects and how DBMS helps achieve them:

Confidentiality:

•Ensures that only authorized users can access and view sensitive data.

•DBMS enforces access control mechanisms through user authentication (logins and passwords) and authorization (permissions to access specific data).

Integrity:

•Maintains the accuracy and consistency of data within the database.

•DBMS implements data validation rules to ensure data entered conforms to specific criteria (e.g., data type, format, range).

Availability:

•Guarantees that authorized users can access the data they need when needed.

•DBMS provides backup and recovery solutions to restore data in case of hardware failures, software errors, or cyberattacks.

•High availability techniques like clustering and replication can ensure minimal downtime even during maintenance or system outages.

Importance of Data Security in DBMS

Data breaches and unauthorized access can have severe consequences for organizations. Robust data security in DBMS helps to:

- **Protect sensitive information:** Customer data, financial records, and intellectual property need to be safeguarded.
- **Maintain compliance with regulations:** Many industries have data privacy regulations that require specific security measures.
- **Minimize downtime and data loss:** Security measures help prevent disruptions and ensure data recovery in case of incidents.
- **Build trust with stakeholders:** Customers and partners need to have confidence that their data is secure.

Additional Security Measures

- **Auditing and Logging:** Tracks user activity within the database, allowing for monitoring and identification of suspicious access attempts.

- **Data Masking and Anonymization:** Techniques to hide sensitive data elements while still allowing for analysis, useful for protecting privacy.

- **Regular Security Patching:** Keeping DBMS software up-to-date with the latest security patches is crucial to address vulnerabilities exploited by attackers.

Parul® University
NAAC A++ ACCREDITED UNIVERSITY

DIGITAL LEARNING CONTENT

Data integrity

Data integrity in DBMS refers to the overall accuracy, consistency, and completeness of data throughout its lifecycle within the database. It ensures that the data stored in the database is reliable and trustworthy for making informed decisions.

Here's a breakdown of the key aspects of data integrity:

- **Accuracy:** Data values should be correct and free from errors. This involves validating data during entry to ensure it conforms to specified data types, formats, and ranges.

- **Consistency:** Data should be consistent across the entire database. This means avoiding duplicate entries and ensuring relationships between tables are maintained.

- **Completeness:** All the necessary data elements should be present and not missing. Incomplete data can lead to misleading or inaccurate results.

Importance of Data Integrity in DBMS

Maintaining data integrity is crucial for several reasons:

- **Reliable Decision-Making:** Accurate and consistent data is essential for generating trustworthy reports, insights, and making informed decisions.

- **Improved Efficiency:** Reduced errors and inconsistencies minimize the time and resources spent on data correction and troubleshooting.

- **Enhanced Data Sharing:** Confidence in data integrity allows for secure data sharing within the organization and with authorized external parties.

- **Regulatory Compliance:** Many regulations require organizations to maintain the accuracy and completeness of data.

How DBMS helps maintain data integrity?

Maintaining data integrity is crucial for several reasons:

Data Validation Rules: DBMS allows defining rules that restrict the type of data that can be entered into specific fields.
Constraints: Primary keys, foreign keys, and other constraints enforce data consistency and prevent inconsistencies like duplicate entries or orphaned records.
Transaction Management: Transactions ensure data updates are atomic (all or nothing). This prevents partial or incomplete modifications that could compromise data integrity.
Data Cleaning Techniques: Regular data cleansing processes can identify and correct errors or inconsistencies in existing data.

Access Control (DAC, RBAC, MAC)

Access control in DBMS (Database Management System) refers to a set of security mechanisms that regulate how users interact with the database. It ensures that only authorized users can access the database, and even then, it defines what actions they can perform on specific data.

Here's a breakdown of the key components of access control:

•**Authentication:** The process of verifying a user's identity. Users typically provide credentials like username and password during login. The DBMS checks these credentials against a stored database of authorized users.

•**Authorization:** The process of determining what a user can do with the database after successful authentication. It defines the level of access (permissions) a user has for specific data or functionalities within the database.

Authentication and Authorization

In DBMS authentication and authorization are two crucial security mechanisms that work together to control access to data. They act like a layered security approach to ensure only authorized users can access and modify data within the database.

1. RBAC (Role-Based Access Control)

RBAC focuses on user roles within the organization. Permissions are assigned to roles, and users are then assigned to appropriate roles based on their job functions and responsibilities.

Benefits:

- **Simplified Administration:** Managing access control becomes easier by defining permissions at the role level rather than for individual users.
- **Improved Security:** Reduces the risk of unauthorized access by granting permissions based on job requirements, not individual users.
- **Scalability:** Adapts well to changes in personnel and organizational structure as permissions are tied to roles, not specific users.

Example: In a company, the "Sales Manager" role might have permission to view and edit customer data, while the "Customer Service Representative" role might only have permission to view customer information

2. DAC (Discretionary Access Control)

DAC grants individual users the authority to control access to specific data objects (e.g., tables, files) they own or have been granted access to. Users can grant permissions (read, write, delete) to other users at their discretion.

Benefits:

- **Flexibility:** Allows users to share data with specific colleagues as needed.
- **Simplicity:** Relatively easy to implement for small-scale environments.

Drawbacks:

- **Security Concerns:** Increased risk of unauthorized access if users grant permissions carelessly.
- **Management Complexity:** Administering access becomes cumbersome in larger environments with many users and data objects.

Example: A project manager might grant read access to a project report to specific team members for collaboration purposes.

3. MAC (Mandatory Access Control)

MAC enforces a centralized, pre-defined security policy that dictates access permissions. Users have no control over access levels. The system administrator defines access rules based on security classifications (e.g., top secret, confidential) assigned to data and user clearances.

Benefits:

- **High Security:** Provides the strictest access control, ideal for highly sensitive data.

Drawbacks:

- **Limited Flexibility:** Inflexible for dynamic environments where access needs might change frequently.
- **Complex Administration:** Requires significant effort to define and maintain security classifications and user clearances.

Example: This model is often used in government agencies or military systems where data is classified based on security levels.

Access Control models

RBAC, DAC, and MAC are three different access control models that define how users interact with and manage data. Each model offers a distinct approach to authorization and access permissions.

Choosing the Right access control model depends on the specific needs of the organization and the sensitivity of the data:

- **RBAC:** A good choice for most organizations due to its balance of security, manageability, and scalability.
- **DAC:** Suitable for smaller environments with well-defined ownership of data objects, but security concerns increase with larger scales.
- **MAC:** Best suited for environments with highly sensitive data requiring the strictest access control measures.

Benefits of Access Control

Data Security: Prevents unauthorized access to sensitive data, protecting confidential information like customer records, financial data, or intellectual property.

Data Integrity: Reduces the risk of accidental or malicious data modification by restricting unauthorized users from modifying data they shouldn't access.

Compliance: Helps organizations comply with regulations that require data privacy and security.

Improved Efficiency: Streamlines database management by ensuring users only have access to the data they need for their tasks.

Authentication

Authentication is the process of verifying a user's identity. It confirms whether someone is who they claim to be when attempting to access the database.

- **Process:** Typically, a user enters a username and password (or uses other credentials like biometrics) during login. The DBMS verifies these credentials against a stored database of authorized users and their passwords (or other authentication factors).

- **Benefits:** Authentication prevents unauthorized access to the database by ensuring only legitimate users can log in.

Authorization

Authorization determines what a user can do with the database after successful authentication. It defines the level of access (permissions) a user has for specific data or functionalities.

- **Process:** Once a user is authenticated, the DBMS checks their assigned permissions. These permissions determine whether the user can view, edit, create, or delete data within the database, or perform specific actions like running reports or administrative tasks.

- **Benefits:** Authorization ensures that even authorized users can only access and modify data based on their assigned permissions. This prevents unauthorized modifications, data breaches, and accidental data deletion.

- Authentication and authorization work together to provide a comprehensive security system for your database:

1. **A user attempts to access the database.**
2. **Authentication:** The user provides credentials (username and password).
3. **The DBMS verifies the credentials.**
4. **If valid (successful authentication):**
    1. **Authorization:** The DBMS checks the user's permissions for the requested action.
    2. **If authorized:** The user is granted access to the data or functionality based on their permissions.
    3. **If unauthorized:** The user is denied access.

Encryption and Decryption

In DBMS, encryption and decryption are powerful tools used to safeguard sensitive data at rest (stored on disk) and in transit (being transmitted). They play a vital role in protecting confidential information from unauthorized access even if someone manages to bypass other security measures.

Encryption

Encryption is the process of transforming data into an unreadable format using a cryptographic algorithm and a secret key. This scrambled data, called ciphertext, is unintelligible without the decryption key.

- **Process:** The DBMS employs a chosen encryption algorithm (e.g., AES-256) along with a secret key. The algorithm acts as a recipe for scrambling the data, while the key acts like a password to unlock the encrypted data. The plain text data is transformed into ciphertext using the algorithm and key.

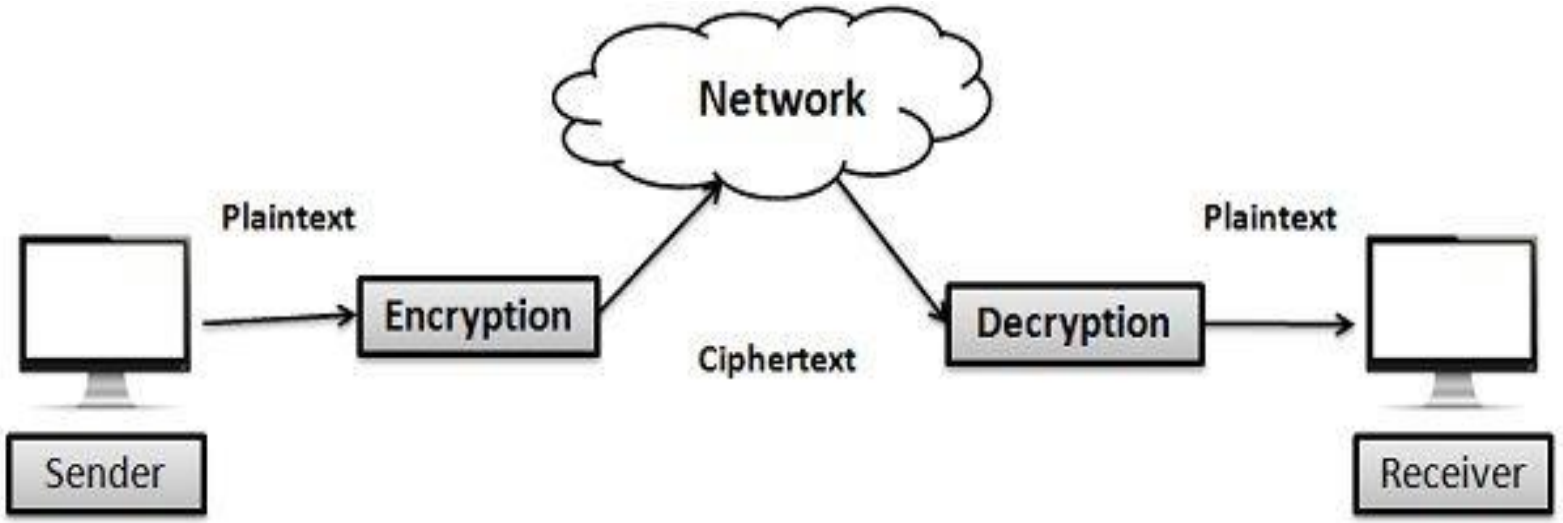
- **Benefits:** Encryption renders sensitive data unreadable even if an attacker gains access to the database. This protects information like customer credit card details, social security numbers, or confidential business data.

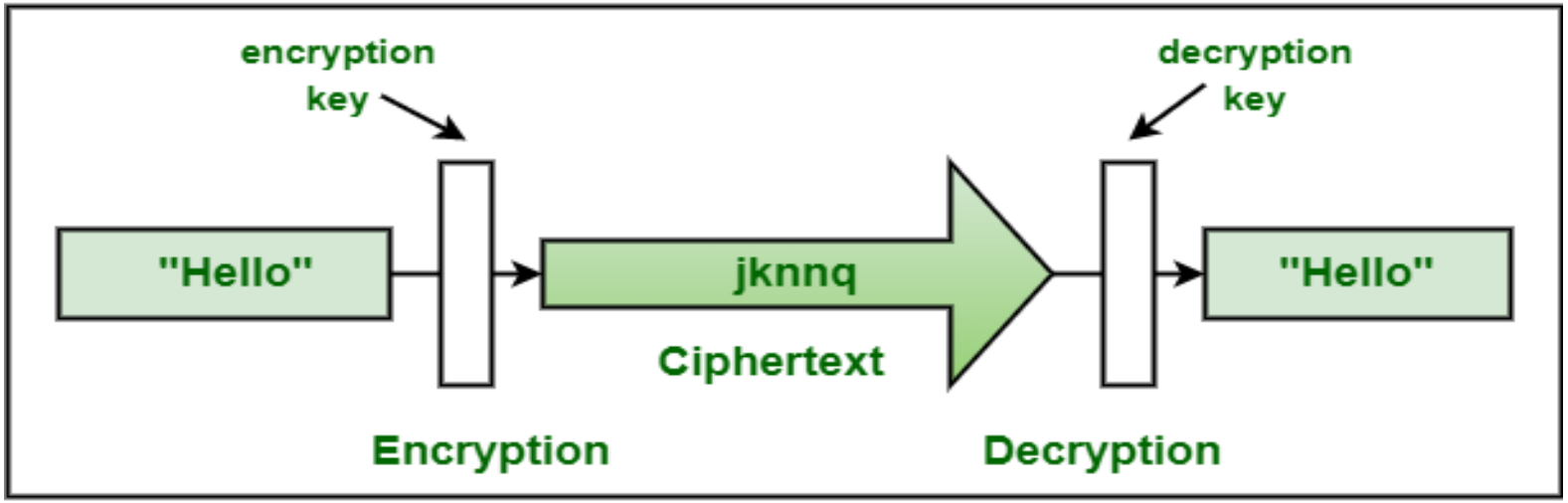
Decryption

Decryption is the reverse process of encryption. It involves transforming ciphertext back into its original, readable plain text format.

- **Process:** The authorized user or application requesting access to the encrypted data provides the decryption key. The DBMS utilizes the same encryption algorithm and the provided key to reverse the transformation, making the data readable again.

- **Benefits:** Decryption allows authorized users to access and work with the confidential data while keeping it secure when stored or transmitted.


Network
Plaintext
Encryption
Ciphertext
Decryption
Plaintext
Sender
Receiver


encryption
key
decryption
key
"Hello"
jknnq
Ciphertext
"Hello"
Encryption
Decryption

Intrusion Detection

It refers to the practice of monitoring and analyzing database activity to identify suspicious or malicious attempts to access, modify, or steal data. It acts as a vital security layer to safeguard your database from unauthorized activities and potential breaches.

Here's a breakdown of the concept:

•**Goal:** To detect and prevent unauthorized access, data manipulation, or denial-of-service attacks targeting the database.

•**Techniques:** Intrusion Detection Systems (IDS) are employed to monitor database activity. These systems can analyze various aspects, including:

- User login attempts (successful and failed)
- Database queries and access patterns
- Data changes and modifications
- Network traffic to and from the database server.

Types of Intrusion Detection

- **Signature-based:** Compares database activity to known attack patterns or signatures stored in a database. This approach is effective for detecting well-known attacks but might miss zero-day attacks (previously unknown vulnerabilities).

- **Anomaly-based:** Analyzes user behavior and database activity patterns to identify deviations from normal usage. This can detect novel attacks but might generate false positives due to legitimate but unusual activity.

Benefits of Intrusion Detection

- **Enhanced Security:** Proactive detection of suspicious activity allows for timely intervention and mitigation of potential threats.
- **Improved Forensics:** Alerts and logs generated by the IDS can aid in forensic analysis after a security incident, helping to identify the source of the attack and assess the damage.
- **Compliance:** Many regulations require organizations to implement intrusion detection measures to protect sensitive data.
- By implementing intrusion detection in DBMS, organizations can significantly enhance their database security and protect valuable data from unauthorized access and malicious attacks.

SQL Injection

SQL injection (SQLi) is a critical security vulnerability that targets applications that interact with databases. It involves attackers injecting malicious SQL code into seemingly harmless user inputs, such as login forms, search bars, or any field that accepts user data. This injected code can then be executed by the database server, potentially leading to a variety of security breaches.

How SQL Injection Works?

Attacker Input: The attacker crafts malicious code, typically a modified SQL statement, and injects it into a vulnerable application field.

Unaware Application: The application processes the user input without proper validation or sanitization, treating it as legitimate data.

Database Execution: The application unknowingly sends the entire user input, including the malicious code, to the database server.

Compromised Database: The database server interprets and executes the injected SQL code, potentially leading to unauthorized actions.

Potential Consequences of SQL Injection

Data Theft: Attackers can steal sensitive data like customer records, financial information, or intellectual property stored in the database.

Data Manipulation: Malicious code can modify or delete data within the database, leading to data corruption or disruption of operations.

Unauthorized Access: Attackers might gain unauthorized access to the database server itself, potentially escalating privileges and compromising the entire system.

Denial-of-Service (DoS): Injected code can overwhelm the database server with requests, causing it to become unavailable to legitimate users.

How to Prevent SQL Injection in DBMS?

Input Validation and Sanitization: Always validate and sanitize user input before using it in database queries. This involves checking for unexpected characters or malicious code and filtering out any harmful elements.

Parameterized Queries: Utilize parameterized queries where placeholders are used for user input instead of directly embedding user data into the SQL statement. This approach separates the data from the query, preventing malicious code execution.

Stored Procedures: Consider using stored procedures, pre-compiled SQL code blocks stored on the database server. These procedures accept parameters and limit the potential for manipulation through user input.

How to Prevent SQL Injection in DBMS?

Regular Security Updates: Maintain the DBMS software and web applications up-to-date with the latest security patches to address known vulnerabilities.

Security Awareness Training: Educate developers and users on the dangers of SQL injection and best practices for secure coding and data handling.

Parul® University

NAAC A++ ACCREDITED UNIVERSITY

DIGITAL LEARNING CONTENT

Thank You!!!

DIGITAL LEARNING CONTENT

Parul® University

www.paruluniversi