NEED FOR DATABASE SECURITY

Database security is the protection of the database against intentional and unintentional threats that may be computer based or non-computer based.

Database security is the business of the entire organization as well as all people who use the data held in the organization's database and any loss or corruption to data would affect the day-to-day operation of the organization and performance of the people.

Therefore, database security encompasses hardware, software, infrastructure, people and data of the organization.

Now there is greater emphasis on database security than in the past as the amount of data stored in corporate database is increasing and people are depending more on the corporate data for decision-making, customer service management, supply chain management and so on.

Any loss or unavailability to the corporate data will cripple today's organization and will seriously affect its performance.

Now the unavailability of the database for even a few minutes could result in serious losses to the organization.

Data Security Risks

We have seen that the database security is the concern of the entire organization. The organization should identify all the risk factors and weak elements from the database security Perspective and find solutions to counter and neutralize each such threat.

A threat is any situation, event or personnel that will adversely affect the database security and the smooth and efficient functioning of the organization.

A threat may be caused by a situation or event involving a person, action or circumstance that is likely to bring harm to the organization.

The harm may be tangible, such as loss of data, damage to hardware, loss of software or intangible such as loss of customer goodwill or credibility and so on.

Data Tampering

Privacy of communications is essential to ensure that data cannot be modified or viewed in transit.

The chances of data tampering are high in case of distributed environments as data moves between sites.

In a data modification attack, an unauthorized party on the network intercepts data in transit and changes that data before retransmitting it.

Data Theft

Data must be stored and transmitted securely, so that information such as credit card numbers cannot be stolen.

Over the Internet and Wide Area Network (WAN) environments, both public carriers and private network owners often route portions of their network through insecure landlines, extremely vulnerable microwave and satellite links, or a number of servers. This situation leaves valuable data opens to view by any interested party. In Local Area Network (LAN) environments within a building or campus, insiders with access to the physical wiring can potentially view data not intended for them.

Falsifying User Identities

In a distributed environment, it becomes more feasible for a user to falsify an identity to gain access to sensitive and important information.

Criminals attempt to steal users' credit card numbers, and then make purchases against the accounts.

Or they steal other personal data, such as bank account numbers and driver's license numbers, and setup bogus credit accounts in someone else's name.

Password-Related Threats

In large systems, users must remember multiple passwords for the different applications and services that they use. Users typically respond to the problem of managing multiple passwords in several ways:

• They may select easy-to-guess password

• They may also choose to standardize passwords so that they are the same on all machines or websites.

All these strategies compromise password secrecy and service availability. Moreover, administration of multiple user accounts and passwords is complex, time-consuming, and expensive.

Unauthorized Access to Data Rows

Certain data rows may contain confidential information that should not be available indiscriminately to users authorized to access the table.

For example, in a shared environment' businesses should have access only to their own data; customers should be able to see only their own orders.

Lack of Accountability

If the system administrator is unable to track users' activities, then users cannot be held responsible for their actions.

There must be some reliable ways to monitor who is performing what operations on the data.

Complex User Management Requirements

System must often support large number of users and therefore they must be scalable.

In such large-scale environments, the burden of managing user accounts and passwords makes your system vulnerable to error and attack.

DATABASE ACCESS CONTROL

Database access control determines:

1. If the user has access to the entire database or just portions of it.

2. What access rights the user has (create, insert, delete, update, read, write)

Access control can support range of administrative policies

1. Centralized Administration: Small number of privileged users may grant and revoke access rights

2. Ownership based Administration: The creator of the table may grant and revoke access rights of the table.

3. Decentralize Administration: The owner of the table my grant and revoke authorization rights to other users, allowing them to grant and revoke access rights to table.

SQL Access controls

Two commands for managing access rights:

1. Grant: Used to grant one or more access rights or can be used to assign a user a role.

Syntax:

GRANT {privilege | role} [ON table] TO {user | role | PUBLIC}

[IDENTIFIED BY password] [WITH GRANT OPTION]

Eg: Grant SELECT ON user TO user1

2. Revoke: Revokes the access rights of a user.

Syntax:

REVOKE {privilege | role} [ON table] FROM {user | role | PUBLIC}

Eg: REVOKE SELECT ON user FROM user1

Typical access rights are: select, insert, update, delete, references.

Role-based Access Control (RBAC) : RBAC decreases administrative burden and improves security.

A database with RBAC needs to provide following capabilities:

1. Create and delete roles

2. Define permissions for a role

3. Assign and cancel assignment of users to roles

Categories of database users:

1. Application owner: An end user who owns database objects as part of an application.

2. End user: An end user who operates on database objects via a particular application but does not own any of the database objects.

3. Administrator: User who has administrative responsibility for part or all of the database.

INFERENCE

Process of performing authorized queries and deducing unauthorized information from the legitimate responses received.

The inference problem arises when the combination of a number of data items is more sensitive than individual items or when a combination of data items can be used to infer data of a higher sensitivity.

The attacker may make user of non sensitive data as well as metadata.

Metadata refers to knowledge about correlations or dependencies among data items that can be used to deduce information not otherwise available to a particular user.

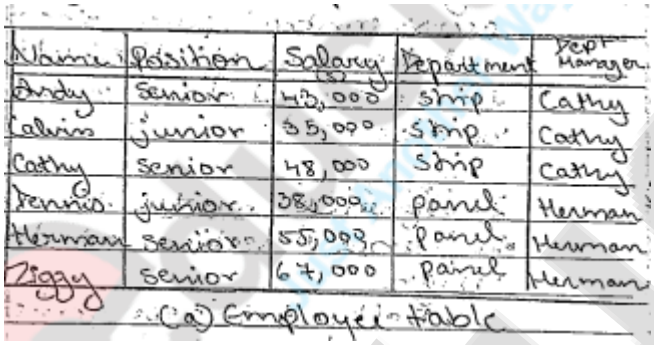
The information transfer path by which unauthorized data is obtained is referred to as an inference channel.

Two inference techniques can be used to derive additional information:

1. Analyzing functional dependencies between attributes within a table or across tables

2. Merging views with same constraints

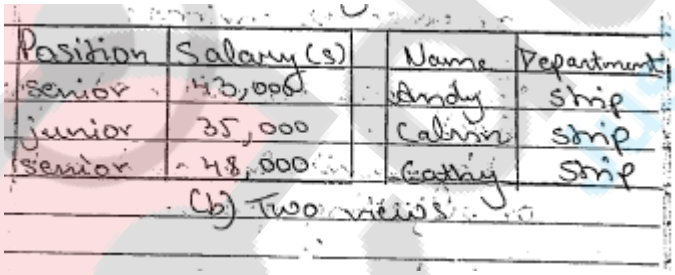
Inference Example



Name	Position	Salary	Department	Dept Manager
Andy	Senior	43,000	Ship	Cathy
Calvin	junior	35,000	Ship	Cathy
Cathy	Senior	48,000	Ship	Cathy
Dennis	junior	38,000	Patrol	Herman
Herman	Senior	55,000	Patrol	Herman
Ziggy	Senior	67,000	Patrol	Herman

(a) Employee table

Employee table

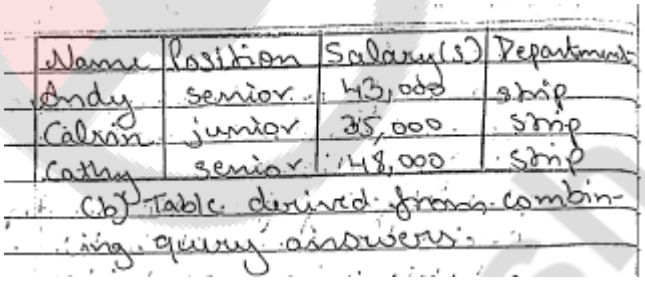


Position	Salary (s)
Senior	43,000
junior	35,000
Senior	48,000

Name	Department
Andy	Ship
Calvin	Ship
Cathy	Ship

(b) Two views

Two views



Name	Position	Salary(s)	Department
Andy	Senior	43,000	Ship
Calvin	junior	35,000	Ship
Cathy	Senior	48,000	Ship

(b) Table derived from combining query answers

Table derived from combining query answes

Figure a shows a table employee with five columns

Figure b shows tow views

Figure c: This violates the access control policy that the relationship of attributes name and salary must not be disclosed.

Inference Counter Measures:

Inference detection at database design:

Alter database structure or access controls

Inference detection at query time:

Counter by monitoring and altering or rejecting queries.

Need some inference detection algorithm

DATABASE ENCRYPTION

Databases typical is a valuable information resource. Protected by multiple layers of security (firewalls, authentication, OS Access control systems, DB access control systems and data encryption)

To encrypt an entire database it is very inflexible and inefficient

To encrypt individual fields it is simple but inflexible

To encrypt records (rows) or columns (attributes) it is best way. It also needs attribute indexes to help data retrieval.

A user at the client can retrieve a record from database with following sequence:

1. The user issues an SQL query for fields from one or more records with a specific value of the primary key.

2. The query processor at the client encrypts the primary key, modifies the SQL query accordingly, and transmits the query to the server.

3. The server processes the query using the encrypted value of the primary key and returns the appropriate record or records.

4. The query processor decrypts the data and returns the results.