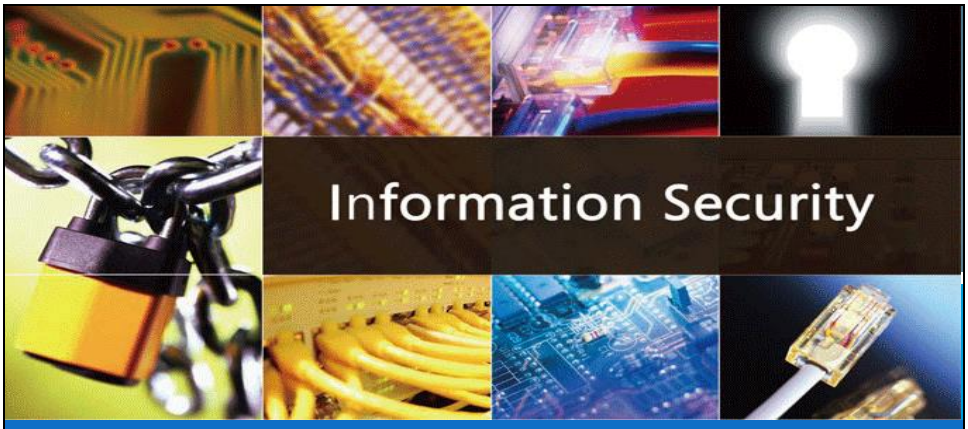



# Database security

Lecturer: Nguyễn Thị Thanh Vân – FIT - HCMUTE

## Objective

- Understand the importance of securing data stored in databases
- Learn how the structured nature of data in databases impacts security mechanisms
- Understand attacks and defenses that specifically target databases

## Importance of Database Security

- Why securing data stored in databases so important and different?
- Databases store massive amounts of sensitive data
- Data has structure that influences how it is accessed
- Accessed via queries or programs written in languages like SQL (Structured Query Language)
- Transactional nature of queries (updates or reads)
- Derived data or database views

## Database Threats Quiz

Choose the best answer.

- Oracle, a major database vendor, sponsored a database security study which identified key security threats. In your view, which of the following is the biggest threat...

☐ External hackers

☐ Insiders and unauthorized users

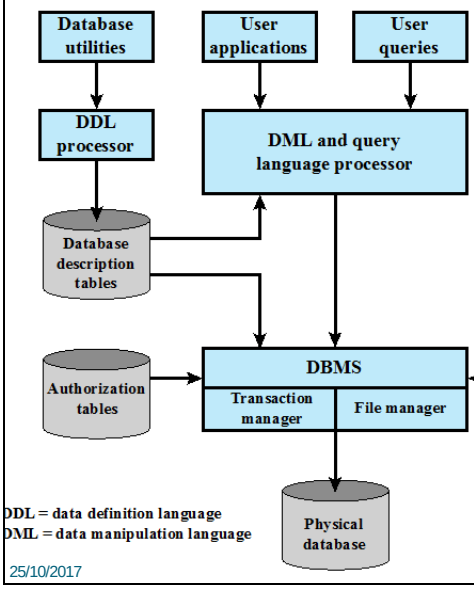
# Database Hacking Quiz

Mark all applicable answers.

Databases are attractive targets for hackers because...

- ☐ They store information such as SS#, DOB etc. that can be easily monetized
- ☐ They store information about lots of users
- ☐ Queries languages used to access data can be abused to gain unauthorized access.

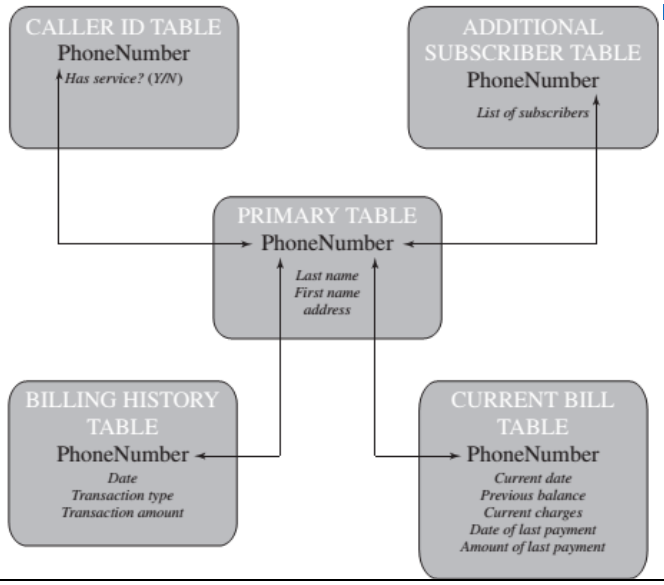
# DBMS Architecture




DDL = data definition language
DML = data manipulation language

- DBS security requirements:
  - OS-based security mechanisms: control r/w
  - security packages
- => security services and mechanisms are designed and integrated with DBSs

# Relational Database Systems (RDBS)




# RDBS - Components

- Relational model based database systems are widely used in real-world environments
- A relational database consists of relations or tables
- A table is defined by a schema and consists of tuples
- Tuples store attribute values as defined by schema
- Keys used to access data in tuples

| Formal Name | Common Name | Also Known As |
|---|---|---|
| Relation | Table | File |
| Tuple | Row | Record |
| Attribute | Column | Field |

# RDBS – ex

**Department Table**

| Did | Dname | Dacctno |
|---|---|---|
| 4 | human resources | 528221 |
| 8 | education | 202035 |
| 9 | accounts | 709257 |
| 13 | public relations | 755827 |
| 15 | services | 223945 |

Primary key

**Employee Table**

| Ename | Did | Salarycode | Eid | Ephone |
|---|---|---|---|---|
| Robin | 15 | 23 | 2345 | 6127092485 |
| Neil | 13 | 12 | 5088 | 6127092246 |
| Jasmine | 4 | 26 | 7712 | 6127099348 |
| Cody | 15 | 22 | 9664 | 6127093148 |
| Holly | 8 | 23 | 3054 | 6127092729 |
| Robin | 8 | 24 | 2976 | 6127091945 |
| Smith | 9 | 21 | 4490 | 6127099380 |

Foreign key

Primary key

# RDBS - SQL

**Operations on relations:**

- Create, select, insert, update, join and delete
- Example: SELECT * FROM EMPLOYEE WHERE DID = '15'
- It returns tuples for Robin and Cody

**Queries written in a query language (e.g., SQL)** use such basic operations to access data in a database as needed.

## RDBS - Quiz

Choose the best answer:

Two tuples (rows) in a relation can have the same primary key value.

- ☐ Yes
- ☐ No

We can use a database view to enhance data security because...

- ☐ It can exclude sensitive attributes that should not be accessible to certain users
- ☐ A view can only be accessed by a single user

## Database Access Control

- a DACS: provides a specific capability that controls access to portions of the database (DAC or BRAC)
- A DBMS can support a range of administrative policies:
  - Centralized administration: A small number of privileged users may grant and revoke access rights.
  - Ownership-based administration: The owner (creator) of a table may grant and revoke access rights to the table.
  - Decentralized administration: In addition to granting and revoking access rights to a table, the owner of the table may grant and revoke authorization rights to other users, allowing them to grant and revoke access rights to the table.

# Database Access Control

- DACS: distinguishes different access right
- Access rights (create, insert, delete, update, read,and write) to:
  - the entire database,
  - individual tables,
  - selected rows or columns within a table.
  - be determined based on the contents of a table entry.
- SQL provides 2 commands: GRANT and REVOKE

```
GRANT       {privileges | role}
[ON         table]
TO          {user | role | public}
[IDENTIFIED BY  password]
[WITH       GRANT OPTION]
```

# Database Access Control

- Privileges can be for operations such as SELECT, INSERT, UPDATE OR DELETE.

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

- Example: REVOKE SELECT ON ANY TABLE FROM Alice

# Database Access Control Quiz

Choose the best answer.

Alice has SELECT access to a table and she can propagate this access to Bob when...

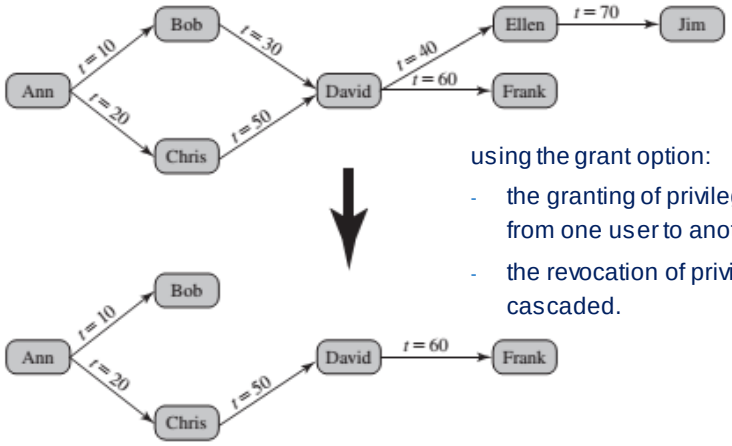
- ☐ Alice was granted this access with GRANT option
- ☐ She can always propagate an access she has

Cascading authorizations occur when an access is propagated multiple times and possibly by several users. Assume that Alice grants access to Bob who grants it further to Charlie. When Alice revokes access to Bob, should Charlie's access be also revoked?

☐ Yes ☐ No

# Cascading Authorizations

- The **grant option** enables an access right to cascade through a number of users




using the grant option:

- the granting of privileges cascades from one user to another,
- the revocation of privileges also cascaded.

16

# Role-Based Access Control

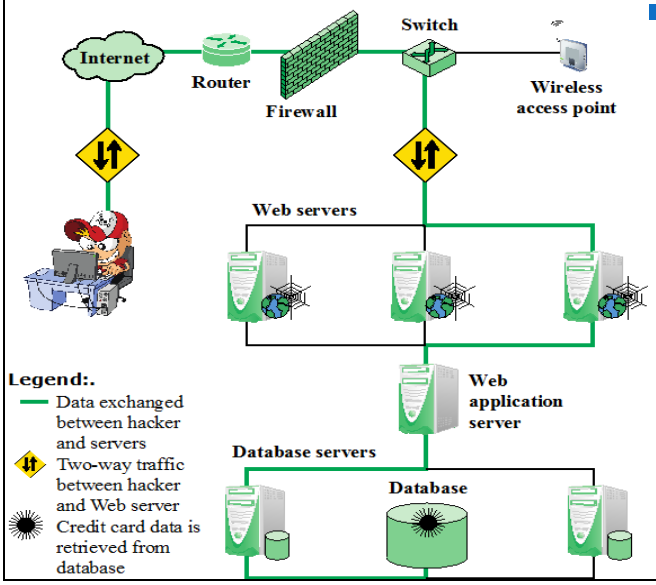
- RBAC:
  - is a natural fit for database access control
  - use of roles in database security
  - provides a means of easing the administrative burden and improving security.
- A database RBAC facility needs to provide the capabilities:
  - Create and delete roles.
  - Define permissions for a role.
  - Assign and cancel assignment of users to roles.
- SQL supports 3 types of roles: server, database, user-defined.
  - The first two types of roles are referred to as fixed roles, are preconfigured for a system with specific access rights.
  - The administrator or user cannot add, delete, or modify fixed roles; it is only possible to add and remove users as members of a fixed role.

25/10/2017 17

# Attacks on Databases: SQL Injections

- Malicious SQL commands are sent to a database
- Can impact both
  - confidentiality (extraction of data) and
  - integrity (corruption of data)
- In a web application environment, typically a script takes user input and builds an SQL query
- Web application vulnerability can be used to craft an SQL injection
- SQL injection attack is one of the most prevalent and dangerous network-based security threats

# A typical SQL Injection




send malicious SQL commands to the database server
=> Results:
- dump database tables
- modify or delete data,
- execute arbitrary OS commands,
- Launch DoS attacks.

# Technique of SQL Injections

The SQLi attack typically works:

- prematurely terminating a text string
- appending a new command.
- terminates the injected string with a comment mark "--".

**Example:**

```
Var Shipcity;
Shipcity = Request.form ("Shipcity");
Var sql = "select * from OrdersTable
where
Shipcity = '" + Shipcity + "'";
```

a user will enter the name of a city. Ex, REDMOND,

- Script generates:

SELECT * FROM OrdersTable Where Shipcity = 'Redmond'.

## SQL Injection Example

- What if user enters:

Redmond' ; DROP table OrdersTable--

- In this case, script is generated:

SELECT * FROM OrdersTable WHERE Shipcity = 'Redmond' ;
DROP OrdersTable

⇒ Server will:

  - select all records in OrdersTable where ShipCity is Redmond.
  - Then, it executes the DROP request

- Malicious user is able to inject code to delete the table
- Many other code injection examples exist

## SQLi Attack Avenues

- User input: In this case, attackers inject SQL commands by providing suitably crafted user input.
- Server variables: variables are logged to a database without sanitization, this could create an SQL injection vulnerability.
- Second-order injection: a malicious user could rely on data already present in the system or database to trigger an SQL injection attack
- Cookies: an attacker could alter cookies when the application server builds an SQL query based on the cookie's content, the structure and function of the query is modified.
- Physical user input: could be scanned using optical character recognition and passed to a database management system.

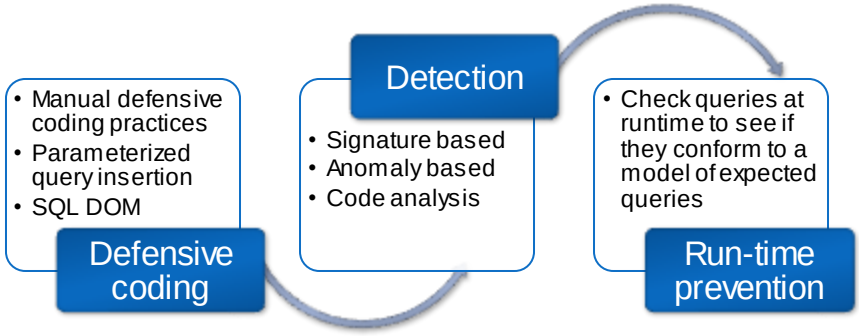
# SQLi attacks

- SQL Injections can do more harm than just by passing the login algorithms. Some of the attacks include
  - Deleting data
  - Updating data
  - Inserting data
  - Executing commands on the server that can download and install malicious programs such as Trojans
  - Exporting valuable data such as credit card details, email, and passwords to the attacker's remote server
  - Getting user login details etc



# SQL Injection Defenses

- An integrated set of techniques is necessary:



# SQL Login Quiz

Mark all applicable answers.

A web application script uses the following code to generate a query:

**Query = "SELECT accounts FROM users WHERE login = ' " + login + " ' AND pass = ' " + password + " ' AND pin = " + pin; The various arguments are read from a form to generate Query.**

This query is executed to get a user's account information when the following is provided correctly...

- ☐ Login name
- ☐ Password
- ☐ PIN

# SQL Login Quiz #2

Choose the best answer.

**Query = "SELECT accounts FROM users WHERE login = ' " + login + " ' AND pass = ' " + password + " ' AND pin = " + pin; The various arguments are read from a form to generate Query.**

If a user types "'or 1 = 1 --" for login in the above query...

- ☐ Query will fail because the provided login is not a correct user
- ☐ An injection attack will result in all users' account data being returned

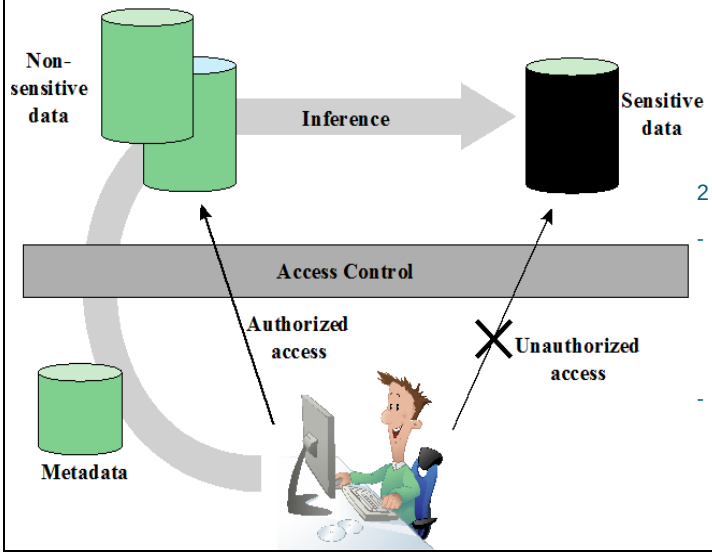
# Inference Attacks on Databases

- Inference attacks:
  - relates to database security
  - is the process of performing authorized queries and deducing unauthorized information from the legitimate responses received.
- Problem:
  - the combination of a number of data items is more sensitive than the individual items,
  - the combination of data items can be used to infer data of a higher sensitivity

# Inference Attacks on Databases, ex

- Average score on an exam is a query that any student should be able to run.
- **Attacker wants to find exact score of some student.**
- **Inference attack when target takes the exam late**
  - Average score before target takes the exam
  - Average score after target takes the exam
  - **Target score can be easily found**

## Indirect Information Access via Inference Channel




2 inference techniques:

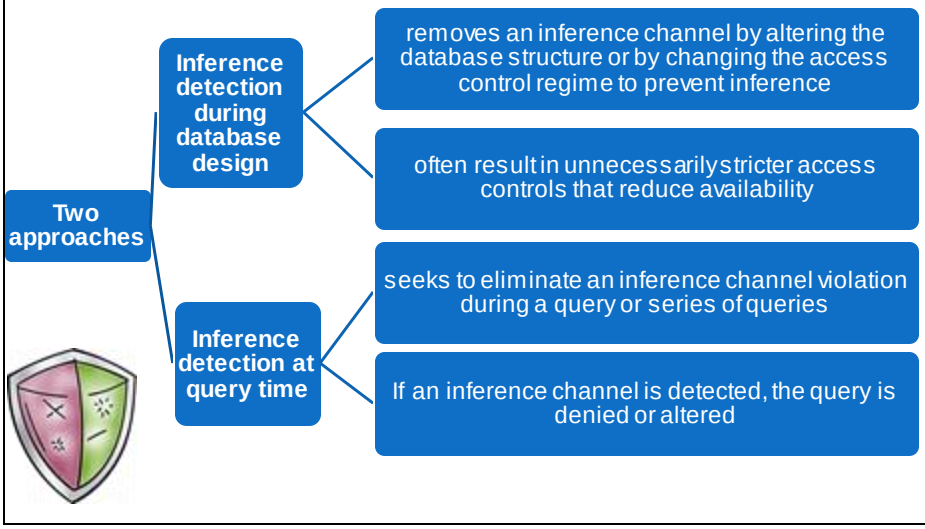
- analyzing functional dependencies between attributes within a table or across tables,
- merging views with the same constraints.

## Inference Attacks on Databases

**Another example:** only one student has junior standing in a senior class

- Get average score of students who have junior standing
- **This query discloses score of a single student**

## Defenses Against Inference Attacks




## SQL Inference Attack Quiz

Choose the best answer.

The database that stores student exam scores allows queries that return average score for students coming from various states. Can this lead to an inference attack in this system?

- ☐ Yes, depending on how many students come from each state
- ☐ No, it is not possible

## SQL Inference Attack Quiz #2

Choose the best answer.

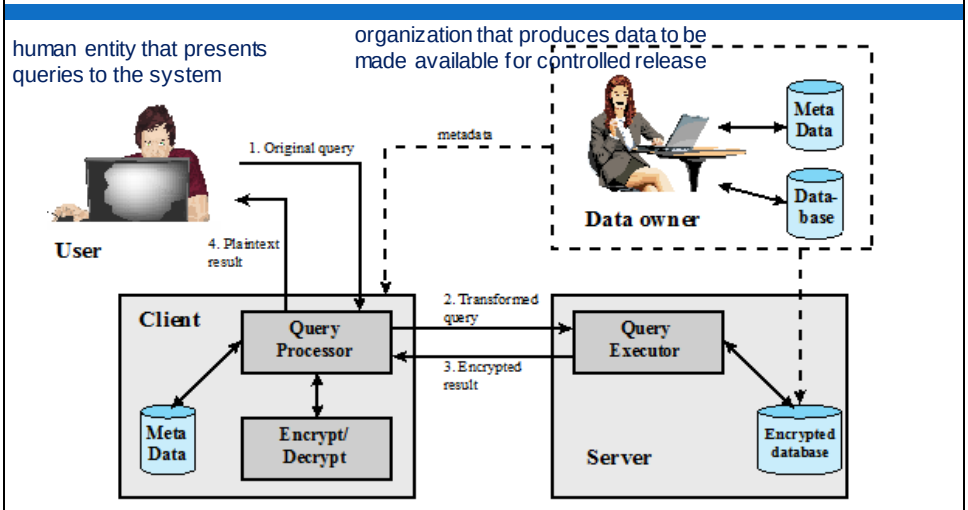
Assume in (1), the data in the database is de-identified by removing student id (and other information such as names). Furthermore, the field that has the state of the student is generalized by replacing it with the US region (e.g., Midwest). The generalization ensures that there are at least two students from each region. Are inference attacks still possible?

☐ Yes ☐ No

## Database encryption

- The database is protected by multiple layers of security:
  - Firewalls
  - Authentication mechanisms
  - General access control systems
  - Database access control systems.
- Database encryption is warranted and often implemented for particularly sensitive data
- There are two disadvantages to database encryption:
  - **Key management:** Authorized users must have access to the decryption key for the data. Providing secure keys to selected parts of the database to authorized users and applications is a complex task.
  - **Inflexibility:** When part or all of the database is encrypted, it becomes more difficult to perform record searching.

# A Database Encryption Scheme




# Summary

- Used to **store lots of sensitive data** that can be accessed via programs (queries)
- Access control must be **based on operations allowed by databases**
- New attacks on databases arise due to their unique characteristics
- Defenses **must address such attacks**

# Q & A