Chapter 5

Database and
Data Center Security

Database Security

- Reasons database security has not kept pace with the increased reliance o databases are:
  - There is a dramatic imbalance between the complexity of modern database management systems (DBMS) and the security technique used to protect these critical systems
  - Databases have a sophisticated interaction protocol, Structured Query Language (SQL), is complex
  - Effective database security requires a strategy based on a full understanding of the secu vulnerabilities of SQL
  - The typical organization lacks full-time database security personnel
  - Most enterprise environments consist of a heterogeneous mixture of database platforms enterprise platforms, and OS platforms, creating an additional complexity hurdle for secu personnel
  - The increasing reliance on cloud technology to host part or all of the corporate database

Databases

- Structured collection of data stored for use by one or more applications
- Contains the relationships between data items and groups of data items
- Can sometimes contain sensitive data that needs to be secured

Query language

- Provides a uniform interface to the database for users and applications

- Database management system (DBMS)
  - Suite of programs for constructing and maintaining the database
  - Offers ad hoc query facilities to multiple users and applications

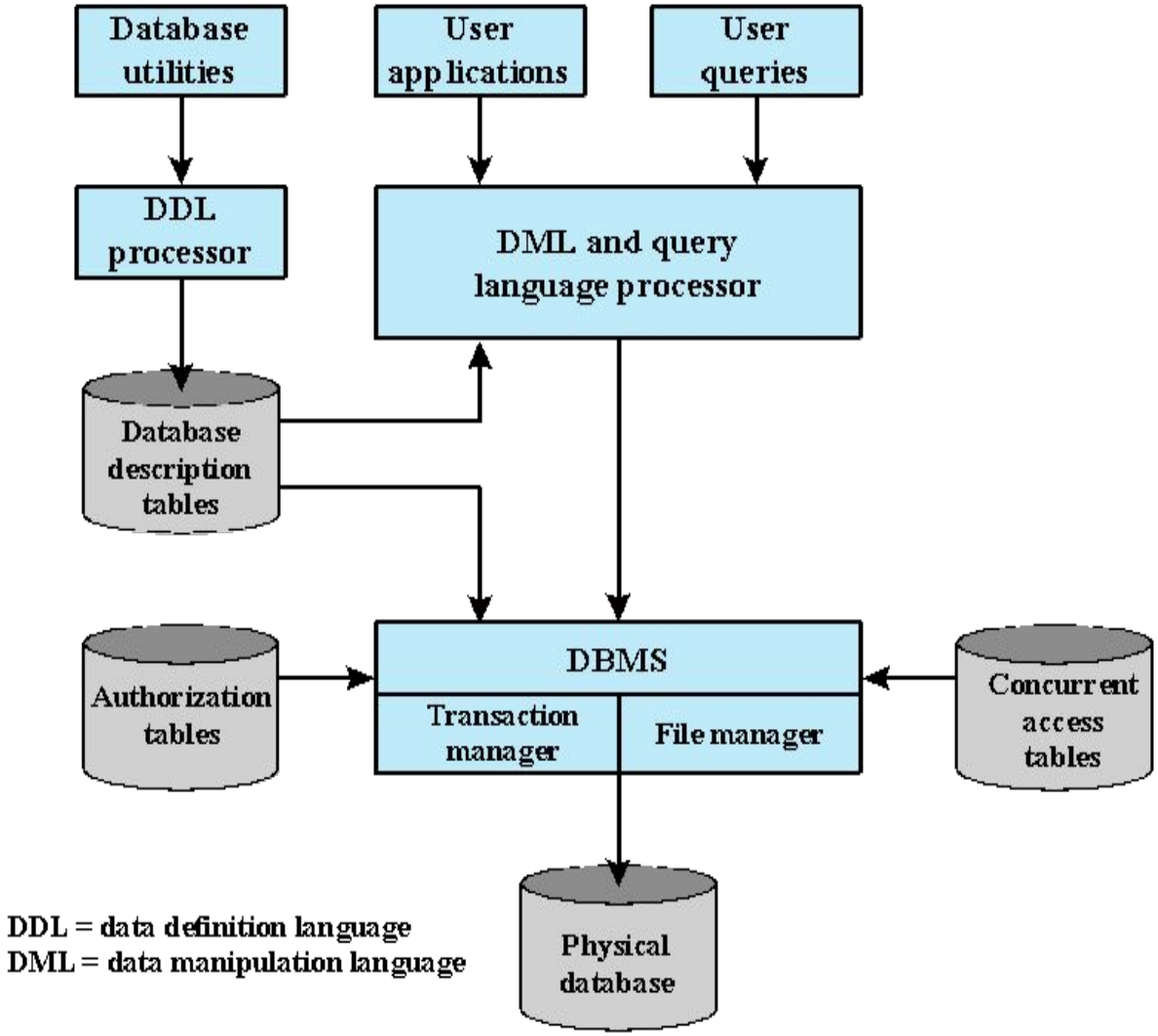


Figure 5.1 DBMS Architecture

Relational Databases

- Table of data consisting of rows and columns
  - Each column holds a particular type of data
  - Each row contains a specific value for each column
  - Ideally has one column where all values are unique, forming an identifier/key for that row
- Enables the creation of multiple tables linked together by a unique identifier that is present in all tables
- Use a relational query language to access the database
  - Allows the user to request data that fit a given set of criteria

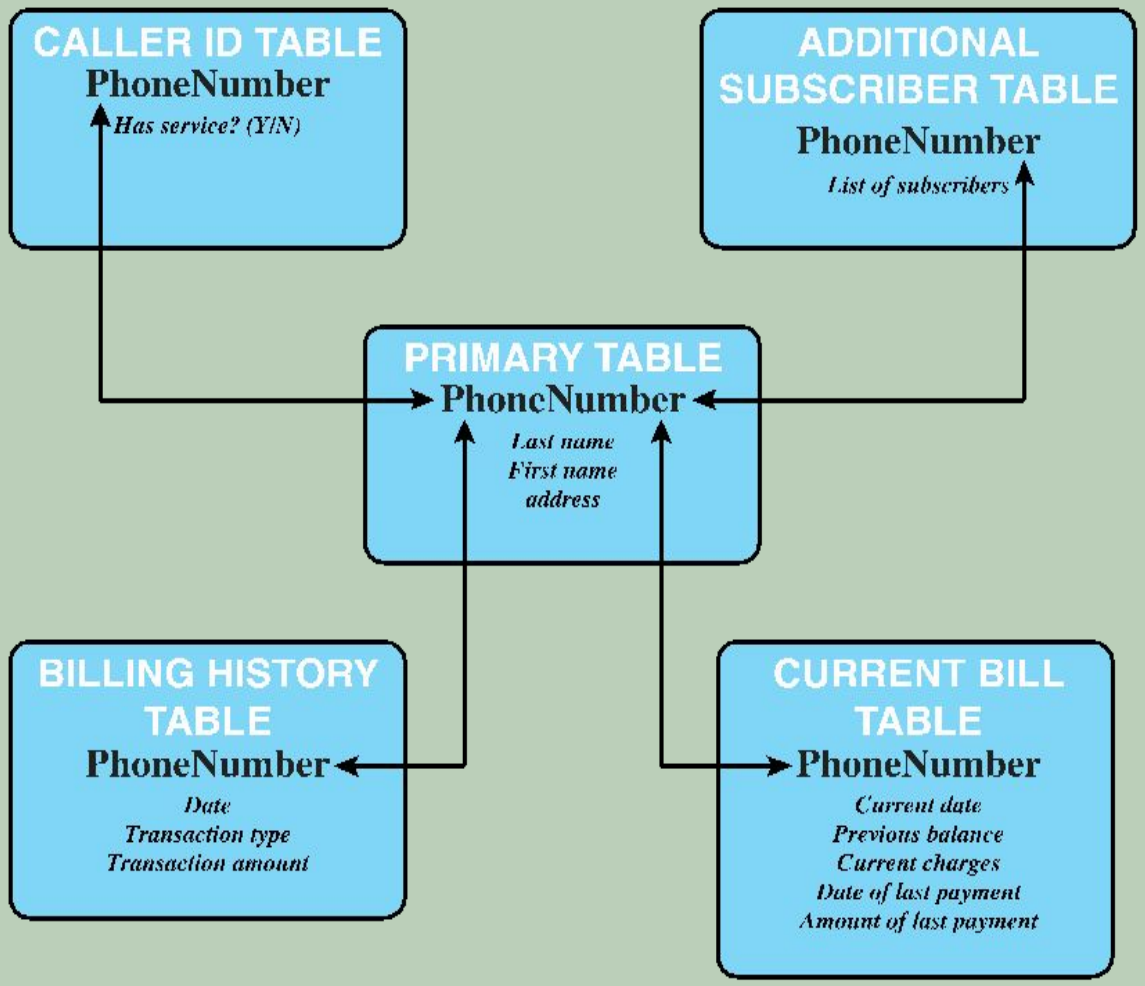


Figure 5.2 Example Relational Database Model. A relational database uses multiple tables related to one another by a designated key; in this case the key is the PhoneNumber field.

Relational Database Elements

- Relation
  - Table/file
- Tuple
  - Row/record
- Attribute
  - Column/field

Primary key

- Uniquely identifies a row
- Consists of one or more column names

Foreign key

- Links one table to attributes in another

View/virtual table

- Result of a query that returns selected rows and columns from one or more tables
- Views are often used for security purposes

Structured Query Language (SQL)

- Standardized language to define schema, manipulate, and query data in a relational database
- Several similar versions of ANSI/ISO standard
- All follow the same basic syntax and semantics

•SQL statements can be used to:
- •Create tables
- •Insert and delete data in tables
- •Create views
- •Retrieve data with query statements

SQL Injection Attacks (SQLi)

- One of the most prevalent and dangerous network-based security threats
- Designed to exploit the nature of Web application pages
- Sends malicious SQL commands to the
- Most common attack goal is bulk extraction of data
- Depending on the environment SQL injection can also be exploited to:
  - Modify or delete data
  - Execute arbitrary operating system commands
  - Launch denial-of-service (DoS) attacks

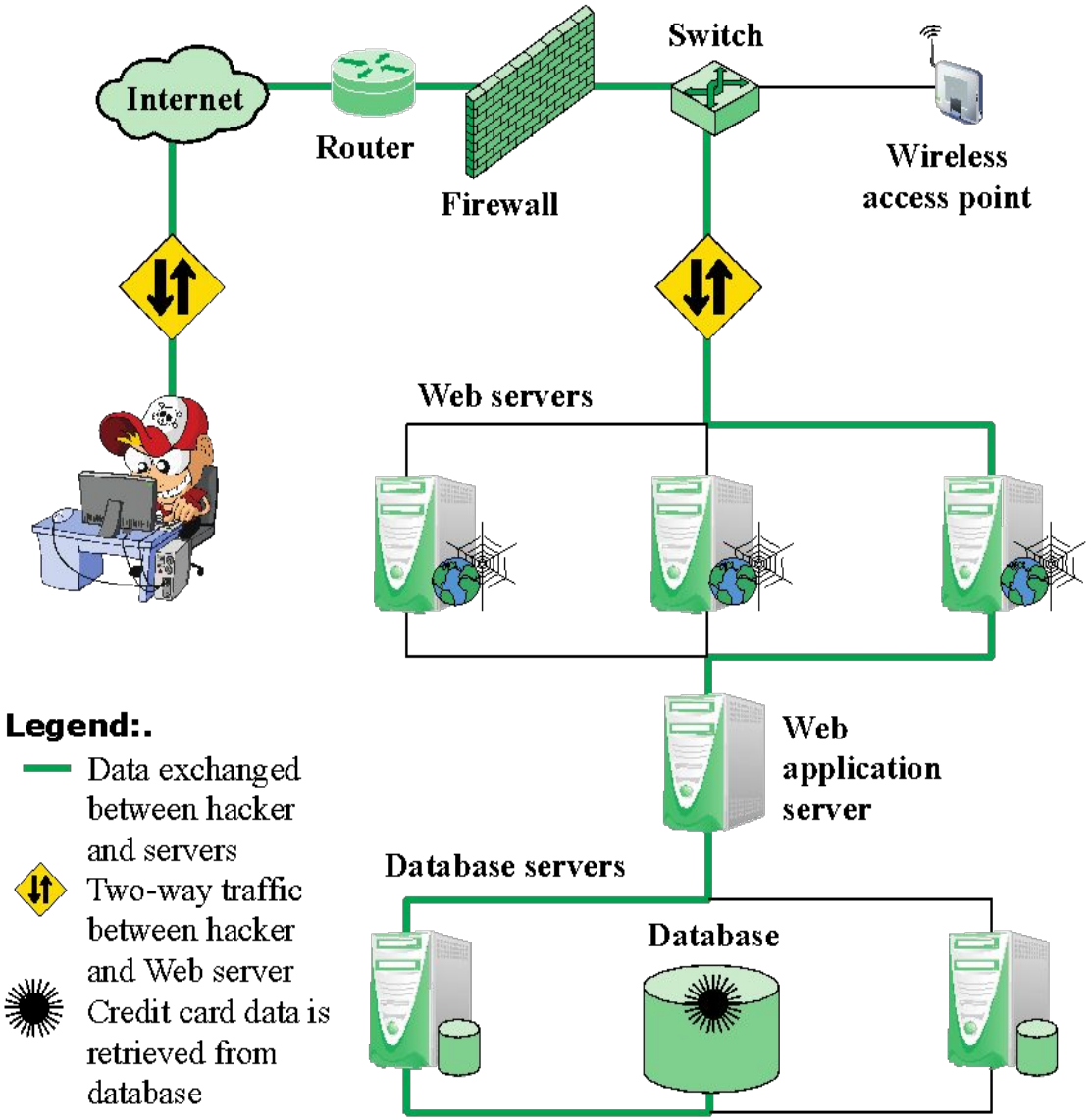


Figure 5.5 Typical SQL Injection Attack

Injection Technique

SQLi Attack Avenues

User input

- Attackers inject SQL commands by providing suitable crafted user input

Server variables

- Attackers can forge the values that are placed in HTTP and network headers and exploit this vulnerability by placing

Second-order injection

- A malicious user could rely on data already present in the system or database to trigger an SQL injection attack, so when the attack

Cookies

- An attacker could alter cookies such that when the application server does not perform sufficient validation but uses the cookie's

Physical user input

- Applying user input that constructs an attack outside the realm of web requests

Inband Attacks

- Uses the same communication channel for injecting SQL code and retrieving results
- The retrieved data are presented directly in application Web page
- Include:

  - Tautology
    - **This form of attack injects code in one or more conditional statements so that they always evaluate to true**
  - End-of-line comment
    - **After injecting code into a particular field, legitimate code that follows are nullified through usage of end of line comments**
  - Piggybacked queries
    - **The attacker adds additional queries beyond the intended query, piggy-backing the attack on top of a legitimate request**

Inferential Attack

- There is no actual transfer of data, but the attacker is able to reconstruct the information by sending particular requests and observing the resulting behavior of the Website/database server
- Include:
  - Illegal/logically incorrect queries
    - This attack lets an attacker gather important information about the type and structure of the backend database of a Web application
    - The attack is considered a preliminary, information-gathering step for other attacks
  - Blind SQL injection
    - Allows attackers to infer the data present in a database system even when the system is sufficiently secure to not display any erroneous information back to the attacker

Out-of-Band Attack

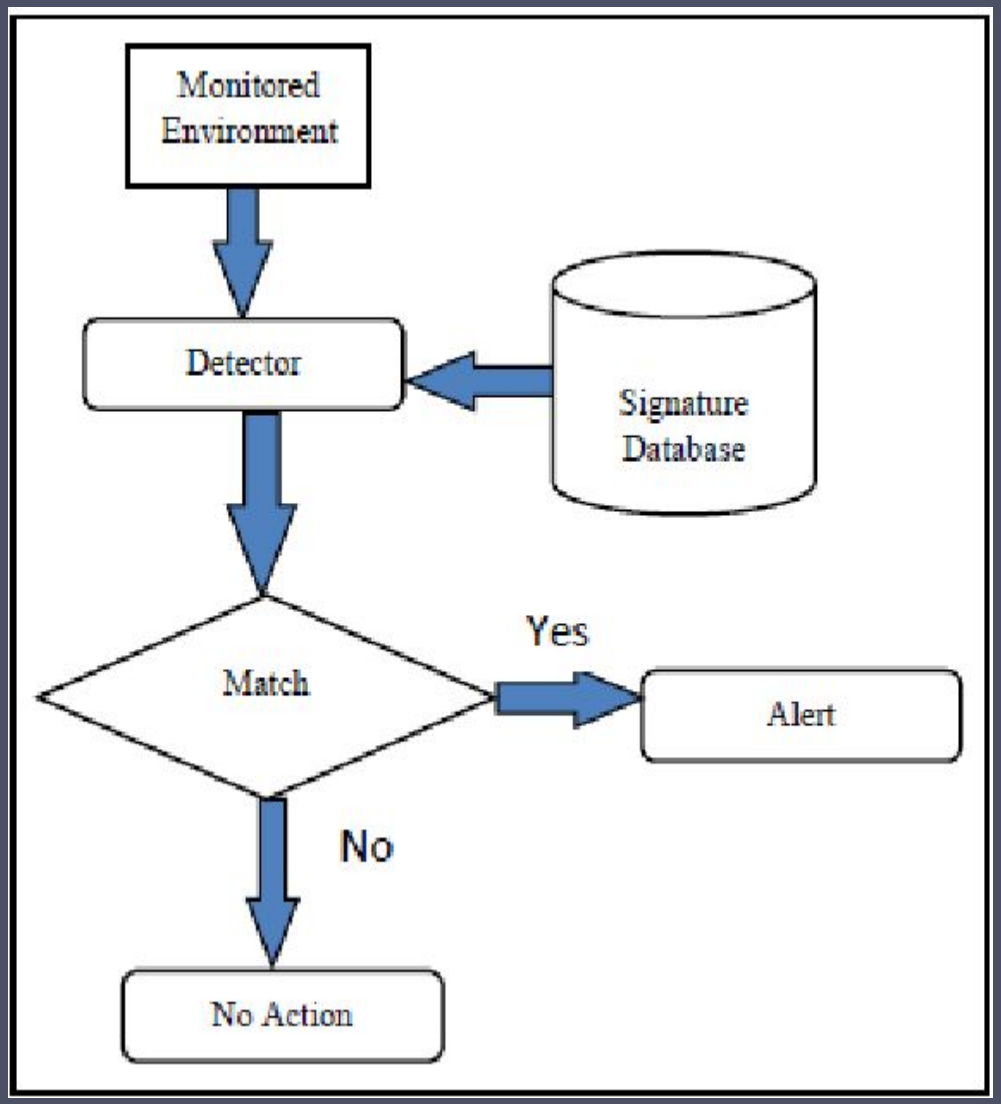
- Data are retrieved using a different channel
- This can be used when there are limitations on information retrieval, but outbound connectivity from the database server is lax

SQLi Countermeasures

- Three types:
  - Defensive coding
    - Manual defensive coding practices
    - Parameterized query insertion
    - SQL DOM
  - Detection
    - Signature based
    - Anomaly based
    - Code analysis
  - Run-time prevention
    - Check queries at runtime to see if they conform to a model of expected queries

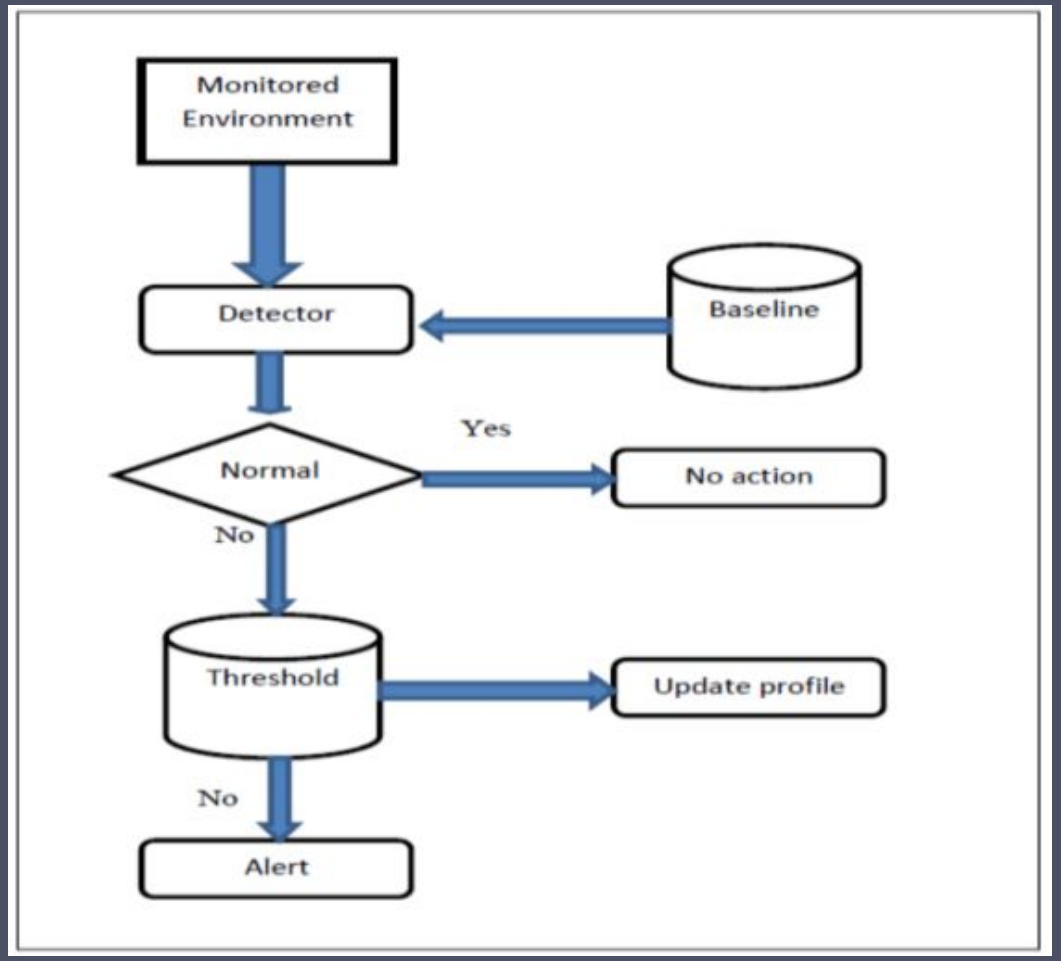
Signature Based Detection

signature-based systems, which can only detect attacks for which a signature has previously been created



Anomaly based Detection

for detecting both network and computer intrusions and misuse by monitoring system activity and classifying it as either *normal* or *anomalous*. The classification is based on heuristics or rules



Database Access Control

Database access control system determines:

- If the user has access to the entire database or just portions of it
- What access rights the user has (create, insert, delete, update, read, write)

Can support a range of administrative policies

Centralized administration

- Small number of privileged users may grant and revoke access rights

Ownership-based administration

- The creator of a table may grant and revoke access rights to the table

Decentralized administration

- 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