People's Democratic Republic of Algeria

Ministery of Higher Education and Scientific Research

Ferhat Abbas University of Setif 1



Faculty of Sciences

Computer Science Department

**THEME**

Detecting SQL injection using Deep Learning

Realized by: Supervised by:

ZITOUNI AHMED FAOUZI DR. BENZINE MEHDI

SEDJAL MOHAMED AYMEN DHIAEDDINE

**2024/2025**

**Table of contenets**

General introduction

Chapter 1 SQL injection

1.1 Introduction

1.2 Web application

1.2.1 Definition…………………………………………………………………………………………………………

1.3 SQL injection………………………………………………………………………………………………………….

1.3.1 Definition6

**Chapter 1**

SQL Injections

**1.1 Introduction**

With increasingly digital living, web applications are at the core of day to day life from managing finances and online purchasing to collaborating and communicating. This ease of the virtual world comes with inherent security challenges. Cyber attackers persistently evolve their methods to exploit weaknesses, thereby endangering unauthorized data access, downtime of services, and irreparable damage to reputation.

        Instead of going with the traditional enumeration, it is pertinent to talk about these five new categories of risks that contemporary web applications have to face:

**Inadequate Input Validation:**

Not validating and sanitizing user-supplied data properly can facilitate a range of attacks, such as injection and cross-site scripting (XSS). Such neglect permits attackers to inject malicious code into the system, thereby violating data integrity and application functionality.

**Insufficient Authentication and Session Management:**

Insufficient methods of validating user identities or handling active sessions can permit unauthorized users to obtain access. Adequate controls such as multi-factor authentication and secure session management are essential for protecting user accounts and sensitive information.

**Lack of data protection:**

Irrespective of whether the data is at rest or in motion, poor encryption or weak key management practices can lead to exposure of sensitive data. Robust cryptographic standards have to be implemented, and sensitive data has to be maintained confidentially from its life cycle for successful protection of data.

**Defective Integration & Dependency Management:**

Contemporary web applications tend to include external services and libraries. When these underlying elements are insecure or outdated, they can inject vulnerabilities into the system overall. Diligent monitoring ofdependencies, along with routine security audits, is required to avert these threats.

**Poor error handling and logging:**

Poorly controlled error messages and a lack of logging can inadvertently disclose system information to prospective attackers or hamper the timely detection of a security incident. Having well defined error handling routines and utilizing solid monitoring systems are essential for the timely detection and containment of malicious activity.

One of the most risky implications of these vulnerabilities is exposure to SQL injection attacks. Where an application does not sanitize input sufficiently, malicious users can inject malicious SQL statements, and these have the potential to result in unauthorized data alteration or even system hijacking. The implication of this type of exploitation can be catastrophic resulting in data loss, system crash, and long-term reputational damage for an organization.

        Through the use of proactive design principles and audits on a regular basis, developers can greatly minimize the chances of exploitation in today's intricate digital world.

**1.2 Web applications**

**1.2.1 Definition**

Web applications are software programs that operate within a web browser, allowing the user to get a set of interactive functionalities such as login systems for user authentication, and real-time chat or messaging features for immediate communication. By doing so, the user is capable of enjoying advanced and interactive functionalities without the installation of other software. Conversely, the entire code that is required is run in the browser of the web, whereas application logic and data are handled on the server side via the implementation of a Database Management System (DBMS), web applications that use DBMS are known as Database-driven Web applications.

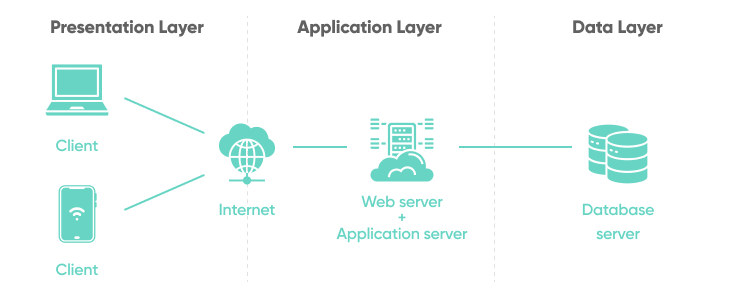
Database-driven Web application are usually composed of a back end database and web pages that include server-side scripts that retrive or update data from the database, depending on user actions like submitting a form, logging in, or searching for products.

Typically, the application relies upon a three-tier structure:

**Presentation Layer:** User interface tier, typically consisting of a web browser or rendering engine using HTML, CSS, and JavaScript.

**Logic Layer:** It handles retrinving and processing the data with the help of server-side scripts.

**Data Layer:** There the data resides, managed by databases including Microsoft SQL Server, MySQL, Oracle, and many more.



**1.3 SQL injection**

**1.3.1 Definition**

A SQL injection attack consists of insertion or “injection” of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system. SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to affect the execution of predefined SQL commands.