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

Realizedby: Supervised by:

ZITOUNI AHMED FAOUZI DR. BENZINE MEHDI

SEDJAL MOHAMED AYMEN DHIAEDDINE

**2024/2025**

**Table of contents**

**General introduction ............................................................................................. 1**

### **Chapter 1: SQL Injection and Modern Detection Approaches…………2**

#### **1.1 Introduction**

#### **1.2 Understanding Web Applications and Databases**

* 1.2.1 Architecture of Web Applications
* 1.2.2 Role of Databases in Web Applications
* 1.2.3 Common Vulnerabilities in Web Applications

#### **1.3 SQL Injection: Definition and Mechanisms**

* 1.3.1 What is SQL Injection?
* 1.3.2 How SQL Injection Works
* 1.3.3 Impact of SQL Injection Attacks

#### **1.4 Techniques of SQL Injection**

* 1.4.1 Tautology-Based SQL Injection
* 1.4.2 Error-Based SQL Injection
* 1.4.3 Blind SQL Injection
  + 1.4.3.1 Content-Based Blind SQL Injection
  + 1.4.3.2 Time-Based Blind SQL Injection
* 1.4.4 Union-Based SQL Injection
* 1.4.5 Other Advanced Techniques

#### **1.5 Traditional Defense Mechanisms Against SQL Injection**

* 1.5.1 Input Validation and Sanitization
* 1.5.2 Parameterized Queries and Prepared Statements
* 1.5.3 Escaping Special Characters
* 1.5.4 Web Application Firewalls (WAF)
* 1.5.5 Limitations of Traditional Methods

#### **1.6 Conclusion**

**Chapter 1**

**1 SQL Injection**

**1.1Introduction**

In today’s increasingly digital world, web applications have become an integral part of daily life, powering everything from online banking and e-commerce to communication and collaboration platforms. While these applications offer unparalleled convenience, they also introduce significant security challenges. Cyber attackers are constantly evolving their methods to exploit vulnerabilities, leading to unauthorized data access, service disruptions, and severe reputational damage for organizations.

Among the myriad of security threats, **SQL Injection (SQLi)** stands out as one of the most prevalent and dangerous attacks. SQL Injection occurs when an attacker exploits vulnerabilities in an application’s input validation mechanisms to inject malicious SQL queries into the database. This can result in unauthorized data access, data manipulation, or even complete system compromise. The consequences of such attacks can be catastrophic, ranging from financial losses and data breaches to long-term damage to an organization’s credibility.

To understand the broader context of SQL Injection, it is essential to examine the common vulnerabilities that plague modern web applications. These vulnerabilities often stem from inadequate security practices and can be categorized into the following five key areas:

1. **Inadequate Input Validation**:  
   Failing to properly validate and sanitize user-supplied data opens the door to a variety of attacks, including SQL Injection and Cross-Site Scripting (XSS). Without robust input validation, attackers can inject malicious code into the system, compromising data integrity and application functionality.
2. **Insufficient Authentication and Session Management**:  
   Weak authentication mechanisms and poor session management can allow unauthorized users to gain access to sensitive systems. Implementing multi-factor authentication (MFA) and secure session handling practices is critical to protecting user accounts and sensitive information.
3. **Lack of Data Protection**:  
   Whether data is at rest or in transit, weak encryption practices or poor key management can lead to data exposure. Adopting robust cryptographic standards and ensuring the confidentiality of sensitive data throughout its lifecycle are essential for effective data protection.
4. **Defective Integration and Dependency Management**:  
   Modern web applications often rely on third-party libraries and external services. If these dependencies are outdated or insecure, they can introduce vulnerabilities into the system. Regular security audits and diligent monitoring of dependencies are necessary to mitigate these risks.
5. **Poor Error Handling and Logging**:  
   Inadequate error handling and insufficient logging can inadvertently expose sensitive system information to attackers or hinder the timely detection of security incidents. Implementing well-defined error-handling routines and robust monitoring systems is crucial for identifying and containing malicious activity.

SQL Injection attacks are a direct consequence of these vulnerabilities, particularly inadequate input validation. When an application fails to sanitize user inputs, attackers can inject malicious SQL statements into queries, potentially gaining unauthorized access to the database. The impact of such attacks can be devastating, leading to data loss, system crashes, and significant reputational damage.

To combat these threats, developers must adopt proactive security measures, including secure coding practices, regular security audits, and the implementation of advanced detection mechanisms. By leveraging modern technologies such as **machine learning (ML)** and **deep learning (DL)**, it is possible to enhance the detection and prevention of SQL Injection attacks. These technologies can analyze patterns in user inputs and database queries to identify and block malicious activity in real-time, offering a more robust defense against evolving cyber threats.

In this project, we aim to explore the application of machine learning and deep learning techniques for SQL Injection detection. By developing intelligent systems capable of identifying and mitigating SQLi attacks, we can contribute to the ongoing effort to secure web applications in an increasingly interconnected world.

#### **1.2 Understanding Web Applications and Databases**