RL4Eng

Development of Remote and Virtual Laboratories for Teaching and Training Engineering Students in the South Mediterranean and Sub-Saharan Higher Education Institutions

Masoud Hamad

Web Security: Understanding SQL Injection

A Comprehensive Guide to SQL, Web Applications, and Security Risks

Co-funded by the Erasmus+ Programme of the European Union

Introduction to SQL

- SQL (Structured Query Language) is used to interact with databases.
- Common operations include:
- - Querying data: SELECT name FROM employees WHERE age > 30;
- - Adding data: INSERT INTO employees (name, age) VALUES ('John', 35);
- - Updating data: UPDATE employees SET age = 36 WHERE name = 'John';
- - Deleting data: DELETE FROM employees WHERE age < 25;

RL4Eng
Development of Remote and Virtual Laboratories for Teaching and Training Engineering Students in the South Mediterranean and Sub-Saharan Higher Education Institutions
Co-funded by the Erasmus+ Programme of the European Union

Components of Web Applications

- Frontend: Collects user input via forms.
- Backend: Processes input and generates SQL queries.
- Database: Executes the SQL query and sends results back.
- Display: Results are shown in the application.
- Example Flow: User Input → Backend Query → Database Query Execution → Result Display

RL4Eng
Development of Remote and Virtual Laboratories for Teaching and Training Engineering Students in the South Mediterranean and Sub-Saharan Higher Education Institutions
Co-funded by the Erasmus+ Programme of the European Union

What is SQL Injection?

- A vulnerability allowing attackers to manipulate SQL queries through user input.
- It can bypass authentication, retrieve sensitive data, or damage databases.
- Example:
- SELECT * FROM users WHERE username = 'admin' AND password = '';

RL4Eng
Development of Remote and Virtual Laboratories for Teaching and Training Engineering Students in the South Mediterranean and Sub-Saharan Higher Education Institutions
Co-funded by the Erasmus+ Programme of the European Union

SQL Injection Example - Login Bypass

- Scenario: A login form where attackers enter malicious input.
- Input:
- Username: admin' --
- Password: (Leave blank)
- Resulting Query:
- SELECT * FROM users WHERE username = 'admin' --' AND password = '';

RL4Eng
Development of Remote and Virtual Laboratories for Teaching and Training Engineering Students in the South Mediterranean and Sub-Saharan Higher Education Institutions
Co-funded by the Erasmus+ Programme of the European Union

SQL Injection Example - Data Leakage

- Scenario: A search form vulnerable to SQL Injection.
- Input:
- ' UNION SELECT credit_card_number, expiry_date FROM credit_cards --
- Resulting Query:
- SELECT name, email FROM users WHERE name = '' UNION SELECT credit_card_number, expiry_date FROM credit_cards --';

RL4Eng
Development of Remote and Virtual Laboratories for Teaching and Training Engineering Students in the South Mediterranean and Sub-Saharan Higher Education Institutions
Co-funded by the Erasmus+ Programme of the European Union

Types of SQL Injection

- Error-Based SQL Injection: Uses database error messages to extract data.
- Union-Based SQL Injection: Combines results of two or more queries.
- Blind SQL Injection: Relies on observing application behavior.
- Time-Based Blind SQL Injection: Uses time delays to infer information

RL4Eng
Development of Remote and Virtual Laboratories for Teaching and Training Engineering Students in the South Mediterranean and Sub-Saharan Higher Education Institutions
Co-funded by the Erasmus+ Programme of the European Union

Impacts of SQL Injection

- Real-World Consequences:
  - Data theft (e.g., credit cards, personal info).
  - Unauthorized access and privilege escalation.
  - Tampering with or deleting critical data.
  - Reputation damage and financial losses.

RL4Eng
Development of Remote and Virtual Laboratories for Teaching and Training Engineering Students in the South Mediterranean and Sub-Saharan Higher Education Institutions
Co-funded by the Erasmus+ Programme of the European Union

Preventing SQL Injection

Effective Prevention Techniques:

- Parameterized Queries: SELECT * FROM users WHERE username = ? AND password = ?;
- Input Validation: Reject unexpected or dangerous characters.
- Database Access Controls: Limit permissions for queries.
- Using ORM Tools: Like Hibernate or Django ORM.

RL4Eng
Development of Remote and Virtual Laboratories for Teaching and Training Engineering Students in the South Mediterranean and Sub-Saharan Higher Education Institutions
Co-funded by the Erasmus+ Programme of the European Union

Preventing SQL Injection

- Security Testing: Regular scans for vulnerabilities.

Tools for Testing SQL Injection

- Burp Suite: Web vulnerability scanner.
- SQLMap: Automated SQL injection testing.
- OWASP ZAP: Comprehensive security testing.

RL4Eng
Development of Remote and Virtual Laboratories for Teaching and Training Engineering Students in the South Mediterranean and Sub-Saharan Higher Education Institutions
Co-funded by the Erasmus+ Programme of the European Union

Lab

https://github.com/massoudhamad/rl4eng-sql-injection

RL4Eng
Development of Remote and Virtual Laboratories for Teaching and Training Engineering Students in the South Mediterranean and Sub-Saharan Higher Education Institutions

Co-funded by the Erasmus+ Programme of the European Union