**BÁO CÁO BÀI THỰC HÀNH SỐ 8**

**KIỂM THỬ LỖ HỔNG SQL INJECTION**

***Họ và tên:Giang Thị Sông Hằng***

***Mssv:1050080133***

***Link youtube:*** <https://youtu.be/6jNTiQQWFAs>

***Yêu cầu kết quả thực hành:***

Báo cáo kết quả thực hành có tên theo định dạng HoTenSV\_MSSV\_Lab08.docx

**Câu 1*(3.5 điểm)***

Kiểm thử lỗ hổng SQL Injection trên trang web http://webvul.bkcs.vn/register.php

* Xác định các tham số đầu vào và cách thức chúng được gửi tới server **(0.25 điểm)**

 **Tham số đầu vào**: username, password, email (gửi bằng phương thức POST).

 **Cách thức gửi**: thông qua form HTML đến register.php.

* Phân tích và phán đoán kiểu truy vấn **(0.25 điểm)**
* INSERT INTO users (username, password, email) VALUES ('$username', '$password', '$email')
* Trong đó các biến $username, $password, $email nhận dữ liệu trực tiếp từ người dùng
* Phân tích và phán đoán vị trí của các giá trị tham số đầu vào được sử dụng trong câu truy vấn **(0.5 điểm)**
* Biến $username, $password, $email được chèn trực tiếp vào truy vấn SQL mà không được kiểm soát → dễ bị chèn mã độc (SQLi).
* Xác định mức độ ảnh hưởng của lỗ hổng trên mỗi giá trị đầu vào **(2.5 điểm)**
*  **Test 1 – Nhập admin' -- vào username**  
  → Truy vấn bị cắt tại dấu --, có thể bỏ qua các phần còn lại.  
  → Dễ tạo ra bản ghi giả mạo, có khả năng **bypass hệ thống đăng ký**.
*  **Test 2 – Dùng payload test', 'test'); DROP TABLE users; --**  
  → Nếu không kiểm soát, có thể **xóa bảng users** → mức độ **nguy hiểm nghiêm trọng**.
*  **Test 3 – Thử chuỗi a'); SELECT \* FROM users --**  
  → Nếu hệ thống cho phép, có thể **trích xuất dữ liệu bảng** → mức độ **rất nghiêm trọng**.
* **Câu 2 *(3.5 điểm)***

Kiểm thử lỗ hổng SQL Injection trên trang web http://webvul.bkcs.vn/login.php

* Xác định các tham số đầu vào và cách thức chúng được gửi tới server **(0.25 điểm)**
* Tham số: username, password
* Phương thức gửi: POST
* Phân tích và phán đoán kiểu truy vấn, phân tích và phán đoán vị trí của các tham số đầu vào được sử dụng trong câu truy vấn **(0.25 điểm)**
* SELECT \* FROM users WHERE username = '$username' AND password = '$password'
* Xác định mức độ ảnh hưởng của lỗ hổng trên mỗi giá trị đầu vào **(3 điểm)**
*  **Payload 1**: ' OR '1'='1  
  → Ghi đè điều kiện xác thực → **đăng nhập thành công không cần mật khẩu**.
*  **Payload 2**: ' UNION SELECT NULL, NULL, version() --  
  → Trích xuất dữ liệu hệ thống → **rò rỉ thông tin server/database**.
*  **Payload 3**: ' OR 1=1 LIMIT 1 OFFSET 1 --  
  → Kỹ thuật dò tìm người dùng thứ 2, thứ 3,… → có thể **khai thác dần danh sách tài khoản**.
* **Câu 3 *(3 điểm)***

Kiểm thử lỗ hổng SQL Injection trên trang web http://webvul.bkcs.vn/index.php

* Xác định các tham số đầu vào và cách thức chúng được gửi tới server **(0.25 điểm)**
* Tham số: có thể là id, search, hoặc page được gửi qua URL (GET method).
* Phân tích và phán đoán kiểu truy vấn, vị trí của các tham số đầu vào được sử dụng trong câu truy vấn **(0.25 điểm)**

SELECT \* FROM products WHERE id = $\_GET['id']

* Xác định mức độ ảnh hưởng của lỗ hổng trên mỗi giá trị đầu vào **(2.5 điểm)**
* **Payload 1**: 1 OR 1=1  
  → Hiển thị toàn bộ dữ liệu → mức độ rò rỉ thông tin **nghiêm trọng**.
* **Payload 2**: 1 UNION SELECT NULL, NULL, table\_name FROM information\_schema.tables --  
  → Khai thác cấu trúc CSDL → có thể dùng để **lập kế hoạch tấn công sâu hơn**.
* **Payload 3**: ' OR 'a'='a  
  → Bỏ qua điều kiện truy vấn → truy cập trái phép vào dữ liệu.