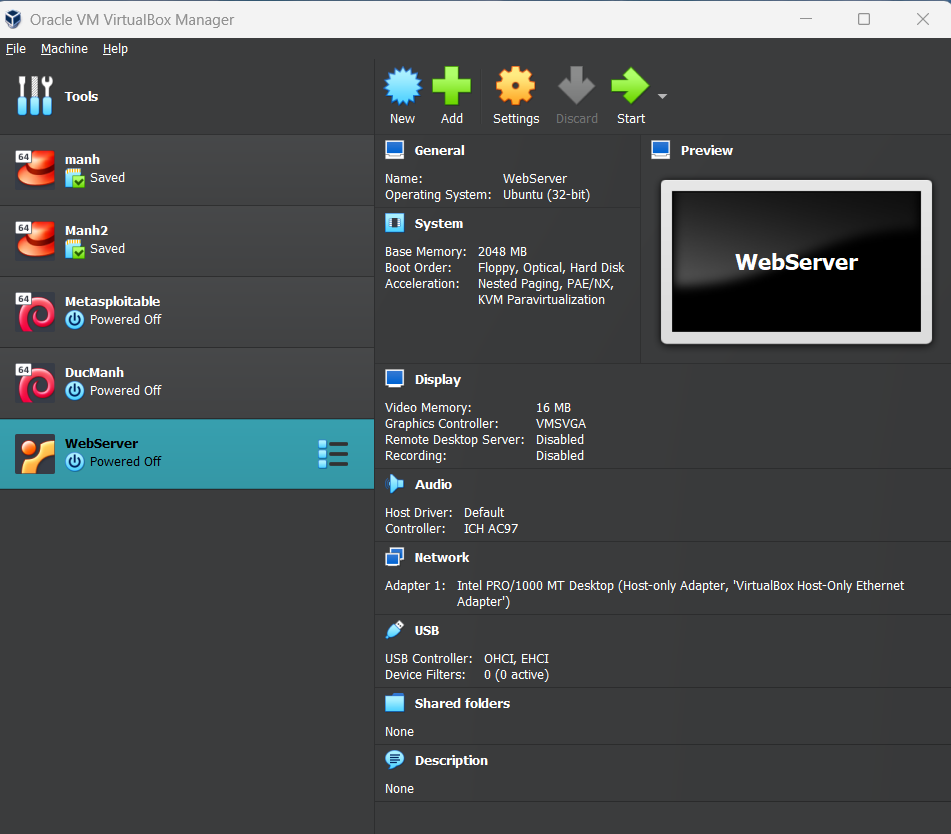
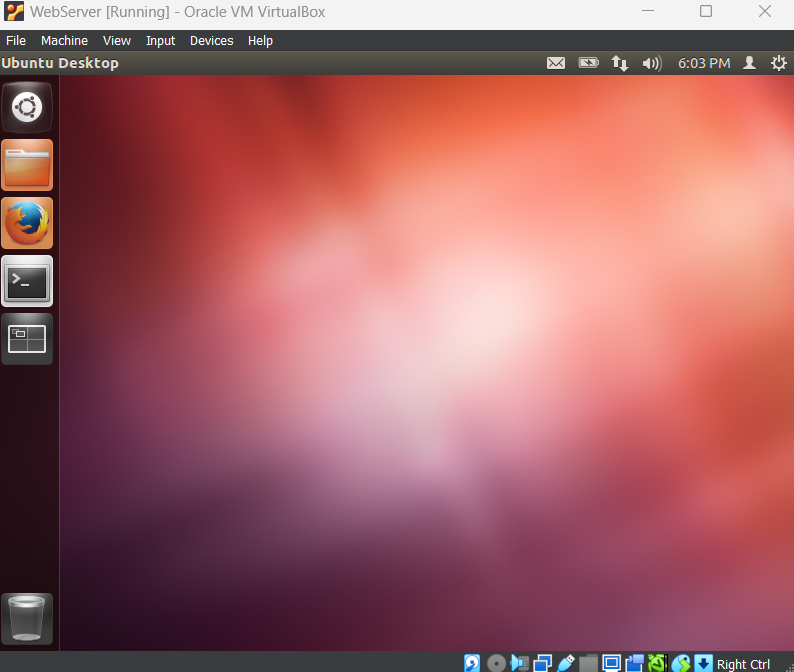
BÁO CÁO BÀI THỰC HÀNH SỐ 8

KIỂM THỬ LỖ HỔNG SQL INJECTION

**Họ và tên:Nguyễn Huỳnh Đức Mạnh**

****

****

**Câu 1(3.5 điểm)**

**1. Tham số đầu vào & cách gửi lên server (0.25đ)**

Trang register.php có form đăng ký với các trường:

- username

- password

- repassword

- email

Dữ liệu được gửi qua phương thức POST như sau:

POST /register.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded username=test&password=123456&repassword=123456&email=test@example.com

**2. Kiểu truy vấn SQL sử dụng (0.25đ)**

Trang đăng ký có thể dùng truy vấn sau:

INSERT INTO users (username, password, email)

VALUES ('$username', '$password', '$email');

Đây là truy vấn INSERT, dễ bị tấn công nếu không kiểm tra đầu vào.

**3. Vị trí sử dụng các giá trị đầu vào trong truy vấn (0.5đ)**

Các biến $username, $password, $email được chèn trực tiếp vào câu lệnh SQL.

Ví dụ payload:

testuser'); DROP TABLE users; --

+ Truy vấn trở thành:

INSERT INTO users (username, password, email)

VALUES ('testuser'); DROP TABLE users; --', '123456', 'abc@example.com');

+ Làm hỏng truy vấn hoặc gây hại nghiêm trọng.

1. **Mức độ ảnh hưởng của từng trường (2.5đ)**

|  |  |  |
| --- | --- | --- |
| Trường | Mức độ ảnh hưởng | Ghi chú ngắn gọn |
| username | Cao | Có thể chèn mã SQL nguy hiểm |
| password | Trung bình | Ít bị khai thác vì thường được mã hóa |
| email | Cao | Dễ bị chèn mã nếu không lọc |
| repassword | Thấp | Chỉ dùng để so sánh, không ảnh hưởng SQL |

**Câu 2 (3.5 điểm)**

**1. Tham số đầu vào & cách gửi tới server (0.25đ)**

Trang login.php có form đăng nhập với các trường:

- username

- password

Dữ liệu được gửi qua phương thức POST như sau:

POST /login.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

username=admin&password=123456

**2. Kiểu truy vấn SQL và vị trí tham số đầu vào (0.25đ)**

Trang login thường sử dụng truy vấn SELECT như sau:

SELECT \* FROM users WHERE username='$username' AND password='$password';

+ Tham số `$username` và `$password` được chèn trực tiếp vào điều kiện WHERE.

Nếu không kiểm tra kỹ, kẻ tấn công có thể thay đổi logic truy vấn để vượt qua xác thực.

**3. Mức độ ảnh hưởng của lỗ hổng trên mỗi giá trị đầu vào (3.0đ)**

Các giá trị đầu vào có thể bị khai thác SQLi là:

|  |  |  |  |
| --- | --- | --- | --- |
| Trường | Payload kiểm thử | Mục đích kiểm thử | Kết quả & Phân tích |
| username | ' OR '1'='1 | Đăng nhập không cần mật khẩu | Truy vấn trở thành: SELECT \* FROM users WHERE username='' OR '1'='1' AND password='...'; → Luôn đúng → Đăng nhập thành công. |
| password | ' OR '1'='1 | Vượt qua xác thực mật khẩu | Nếu username đúng, chỉ cần bypass password → Đăng nhập thành công. |
| username | '; DROP TABLE users; -- | Phá hủy dữ liệu | Truy vấn bị tách làm 2 câu: xóa bảng users. → Mức độ ảnh hưởng rất nghiêm trọng. |

Tất cả giá trị trên đều cho thấy hệ thống dễ bị SQL Injection nếu không có biện pháp lọc và kiểm tra đầu vào như dùng prepared statement hoặc escape ký tự.

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)

- 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)

- Xác định mức độ ảnh hưởng của lỗ hổng trên mỗi giá trị đầu vào (2.5 điểm)