Bùi Nguyễn Thanh Nhân

1050080026-CNTT1

BÀI THỰC HÀNH SỐ 8

KIỂM THỬ LỖ HỔNG SQL INJECTION

<https://github.com/nhan81/lab08>

<https://www.youtube.com/@NhanThanh-vr5ox>

Câu 1(3.5 điểm)

Kiểm thử lỗ hổng SQL Injection trên trang web <https://hackthissite.org>

Cài đặt công cụ

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

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

Ảnh có chứa văn bản, phần mềm, Biểu tượng máy tính, Trang web

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, phần mềm, Biểu tượng máy tính, Phần mềm đa phương tiện

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, phần mềm, Phông chữ, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

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

Tham số đầu vào:

Username (Tên đăng nhập)

Password (Mật khẩu)

Cách thức gửi đến server:

Dựa vào giao diện form HTML và thông báo lỗi nhận được, ta suy ra dữ liệu được gửi thông qua phương thức POST tới máy chủ.

URL: http://65.61.137.117/login.jsp

Dữ liệu gửi lên sẽ có dạng:

http

Sao chép

Chỉnh sửa

POST /login.jsp HTTP/1.1

Host: 65.61.137.117

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

Username=admin&Password=123456

🔹 2. Phân tích và phán đoán kiểu truy vấn (0.25 điểm)

Đây là truy vấn đăng nhập sử dụng thông tin xác thực.

Dữ liệu được người dùng nhập sẽ được đưa vào câu lệnh SQL phía backend, ví dụ:

sql

Sao chép

Chỉnh sửa

SELECT \* FROM users WHERE username = 'admin' AND password = '123456';

🔹 3. 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)

Các giá trị Username và Password được chèn trực tiếp vào phần WHERE của câu truy vấn SQL.

Nếu không có cơ chế kiểm tra hoặc lọc dữ liệu đầu vào, có thể gây ra lỗ hổng SQL Injection.

Vị trí:

sql

Sao chép

Chỉnh sửa

username = 'GIÁ\_TRỊ\_TỪ\_FORM\_USERNAME'

password = 'GIÁ\_TRỊ\_TỪ\_FORM\_PASSWORD'

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

Trường dữ liệu Mức độ rủi ro Giải thích

Username Cao Nếu không được kiểm soát tốt, có thể bị khai thác bằng payload như ' OR '1'='1 để bỏ qua xác thực, dẫn đến truy cập trái phép.

Password Trung bình đến cao Có thể dùng kết hợp với username để khai thác SQLi. Tuy nhiên, thường ít được khai thác một mình.

Tổng thể Nguy hiểm cao Nếu cả hai trường được đưa trực tiếp vào SQL mà không qua prepared statements hoặc escaping, thì có thể dễ dàng bị khai thác qua SQL Injection để:

- Đăng nhập trái phép

- Lấy dữ liệu người dùng

- Thực hiện tấn công tiếp theo (escalation, lateral movement)