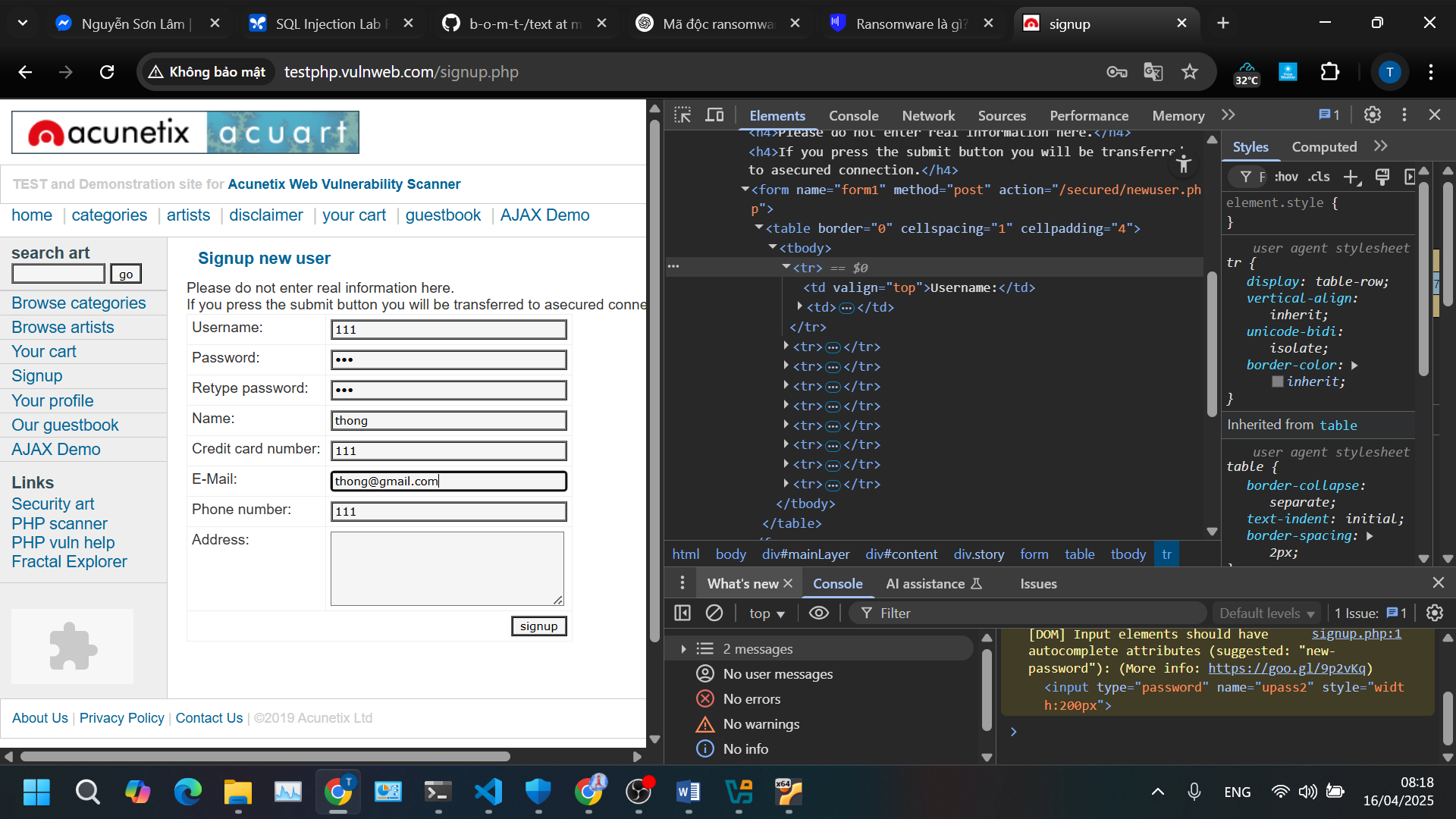
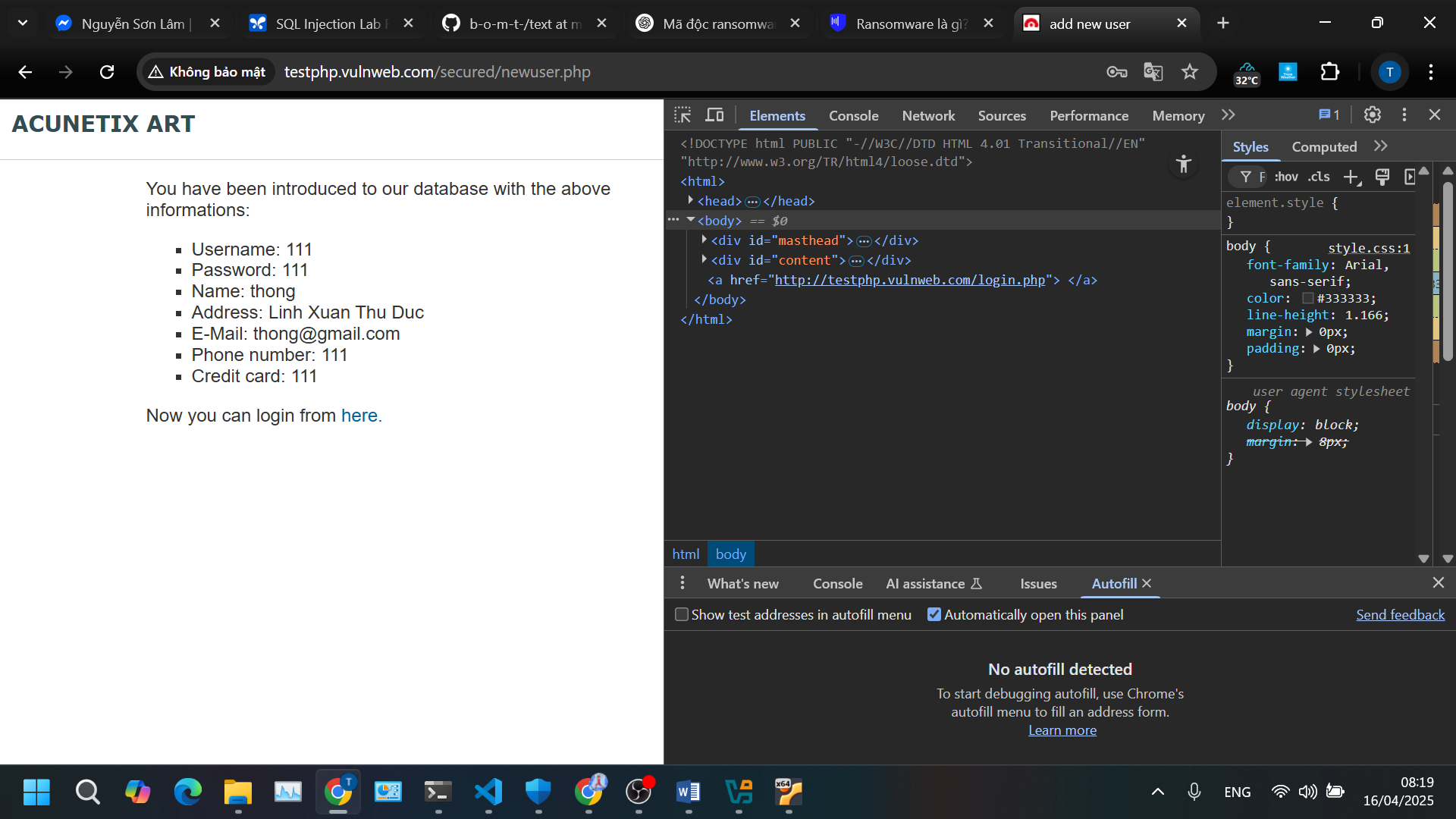
BÁO CÁO BÀI THỰC HÀNH SỐ 8

KIỂM THỬ LỖ HỔNG SQL INJECTION

Phạm Đức Thông

**Câu 1**





**Giá trị đầu vào** username=111 password=111 retype-password=111 name=thong Credit-card-number=111 e-mail=thong@gmail.com phone-number=111

**Cách thức** được gửi tới Server bằng phương thức POST

**Phân tích và phán đoán kiểu truy vấn** Dựa vào chức năng đăng ký, có thể phán đoán kiểu truy vấn là INSERT để thêm dữ liệu người dùng mới vào bảng users. Dạng truy vấn có thể là:

INSERT INTO users (username, password, name, credit\_card\_number, email, phone\_number) VALUES ('[username]', '[password]', '[name]', '[credit\_card\_number]', '[email]', '[phone\_number]')

**Phân tích và phán đoán vị trí của các giá trị tham số đầu vào được sử dụng trong câu truy vấn** Các giá trị của các tham số username, password, name, credit\_card\_number, email và phone\_number được sử dụng trực tiếp trong mệnh đề VALUES của câu lệnh INSERT. Chúng có thể không được xử lý đúng cách

**Xác định mức độ ảnh hưởng của lỗ hổng trên mỗi giá trị đầu vào**

tập trung vào việc sử dụng giá trị 111 làm cơ sở, và thử các biến thể SQL Injection trên đó.

username:

Giá trị kiểm thử: 111' OR '1'='1

Mục đích: Kiểm tra khả năng chèn SQL Injection thông qua trường username.

Phân tích: Nếu không được xử lý đúng cách, username sẽ trở thành 111' OR '1'='1. Câu truy vấn có thể vẫn thực hiện, nhưng username sẽ bị thay đổi.

Mức độ ảnh hưởng: Trung bình, username bị sai lệch.

password:

Giá trị kiểm thử: 111' OR '1'='1

Mục đích: Tương tự như username, kiểm tra khả năng chèn SQL Injection.

Phân tích: Nếu không được xử lý đúng cách, password sẽ bị thay đổi.

Mức độ ảnh hưởng: Trung bình, password bị sai lệch.

retype-password:

Trường này thường dùng để so sánh với trường password trước khi thực hiện truy vấn. Việc chèn SQL Injection vào trường này ít có khả năng gây ảnh hưởng trực tiếp đến cơ sở dữ liệu, trừ khi logic so sánh password và repassword bị lỗi.

Mức độ ảnh hưởng: Thấp.

name:

Giá trị kiểm thử: 111' OR '1'='1

Mục đích: Tương tự như username và password, kiểm tra khả năng chèn SQL Injection.

Phân tích: Nếu không được xử lý đúng cách, name sẽ bị thay đổi.

Mức độ ảnh hưởng: Trung bình, name bị sai lệch.

Credit-card-number:

Giá trị kiểm thử: 111' OR '1'='1

Mục đích: Kiểm tra khả năng chèn SQL Injection thông qua trường Credit card.

Phân tích: Nếu không được xử lý đúng cách, credit\_card\_number sẽ bị thay đổi.

Mức độ ảnh hưởng: Trung bình, credit\_card\_number bị sai lệch.

Giá trị kiểm thử: 111; DROP TABLE users; --

Mục đích: Thử chèn lệnh phá hoại.

Phân tích: Nếu thành công, có thể xóa bảng users.

Mức độ ảnh hưởng: Rất cao.

e-mail:

Giá trị kiểm thử: thong@gmail.com' OR '1'='1

Mục đích: Tương tự như các trường khác, kiểm tra khả năng chèn SQL Injection.

Phân tích: Nếu không được xử lý đúng cách, e-mail sẽ bị thay đổi.

Mức độ ảnh hưởng: Trung bình, e-mail bị sai lệch.

Giá trị kiểm thử: thong@gmail.com'; DROP TABLE users; --

Mục đích: Thử chèn lệnh phá hoại.

Phân tích: Nếu thành công, có thể xóa bảng users.

Mức độ ảnh hưởng: Rất cao.

phone-number:

Giá trị kiểm thử: 111' OR '1'='1

Mục đích: Tương tự như các trường khác, kiểm tra khả năng chèn SQL Injection.

Phân tích: Nếu không được xử lý đúng cách, phone\_number sẽ bị thay đổi.

Mức độ ảnh hưởng: Trung bình, phone\_number bị sai lệch.

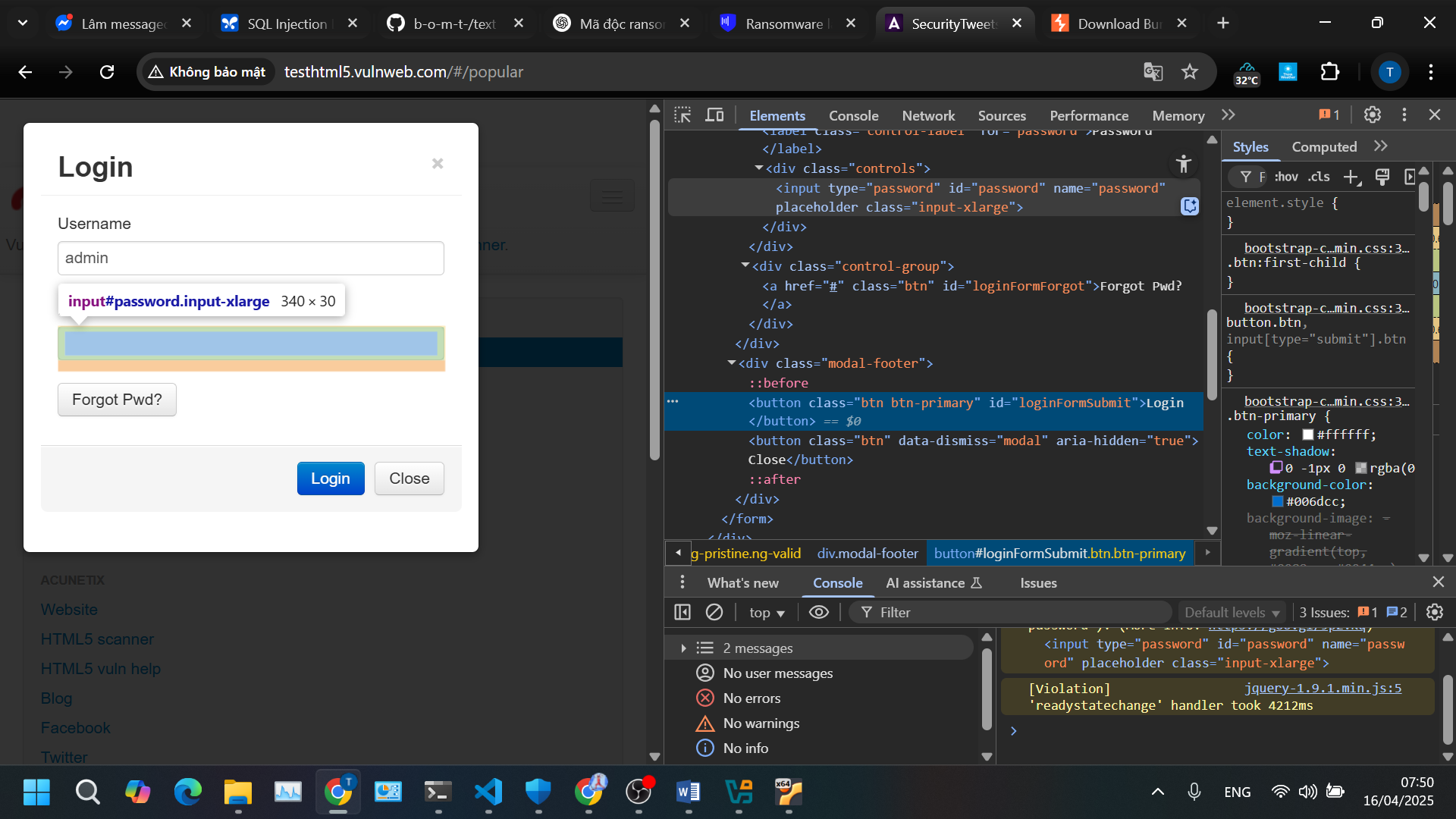
Giá trị kiểm thử: 111; DROP TABLE users; --

Mục đích: Thử chèn lệnh phá hoại.

Phân tích: Nếu thành công, có thể xóa bảng users.

Mức độ ảnh hưởng: Rất cao.

**Câu 2**



**Tham số đầu vào** là username=admin, password=111

**Cách thức** được gửi tới Server bằng phương thức POST

**Phân tích và phán đoán kiểu truy vấn** Dựa vào chức năng đăng ký, có thể phán đoán kiểu truy vấn là INSERT để thêm dữ liệu người dùng mới vào bảng users. Dạng truy vấn có thể là:

SELECT \* FROM users WHERE username = '[username]' AND password = '[password]'

**Xác định mức độ ảnh hưởng của lỗ hổng trên mỗi giá trị đầu vào**

username:

Mục đích sử dụng giá trị kiểm thử: Kiểm tra khả năng bypass đăng nhập thông qua SQL Injection.

Giá trị kiểm thử: ' OR '1'='1

Kết quả: Nếu website không xử lý input đúng cách, câu truy vấn có thể trở thành:

SELECT \* FROM users WHERE username = '' OR '1'='1' AND password = '[password]'

Mệnh đề '1'='1' luôn đúng, do đó câu truy vấn sẽ trả về tất cả các user trong bảng users. Nếu website chỉ kiểm tra sự tồn tại của kết quả truy vấn thay vì kiểm tra username và password cụ thể, có thể bypass đăng nhập và đăng nhập vào tài khoản đầu tiên trong bảng.

Giá trị kiểm thử: admin' --

Kết quả: Nếu website không xử lý input đúng cách, câu truy vấn có thể trở thành:

SELECT \* FROM users WHERE username = 'admin' --' AND password = '[password]'

Ký tự -- được sử dụng để comment phần còn lại của câu truy vấn, bỏ qua việc kiểm tra password. Nếu có user có username là admin, sẽ bypass đăng nhập.

Giá trị kiểm thử: ' OR 1=1#

Kết quả: Nếu website không xử lý input đúng cách, câu truy vấn có thể trở thành:

SELECT \* FROM users WHERE username = '' OR 1=1#' AND password = '[password]'

Tương tự như --, kí tự # dùng để comment, bypass kiểm tra password, và đăng nhập vào tài khoản đầu tiên.

Mức độ ảnh hưởng: Rất cao, có thể bypass đăng nhập, truy cập trái phép vào tài khoản của người khác hoặc tài khoản quản trị.

password:

Mục đích sử dụng giá trị kiểm thử: Tương tự như username, kiểm tra khả năng bypass đăng nhập thông qua SQL Injection.

Giá trị kiểm thử: ' OR '1'='1

Kết quả: Nếu website không xử lý input đúng cách, câu truy vấn có thể trở thành:

SELECT \* FROM users WHERE username = '[username]' AND password = '' OR '1'='1'

Tương tự như username, mệnh đề '1'='1' luôn đúng, do đó câu truy vấn sẽ trả về tất cả các user, có thể bypass đăng nhập.

Giá trị kiểm thử: ' OR 1=1#

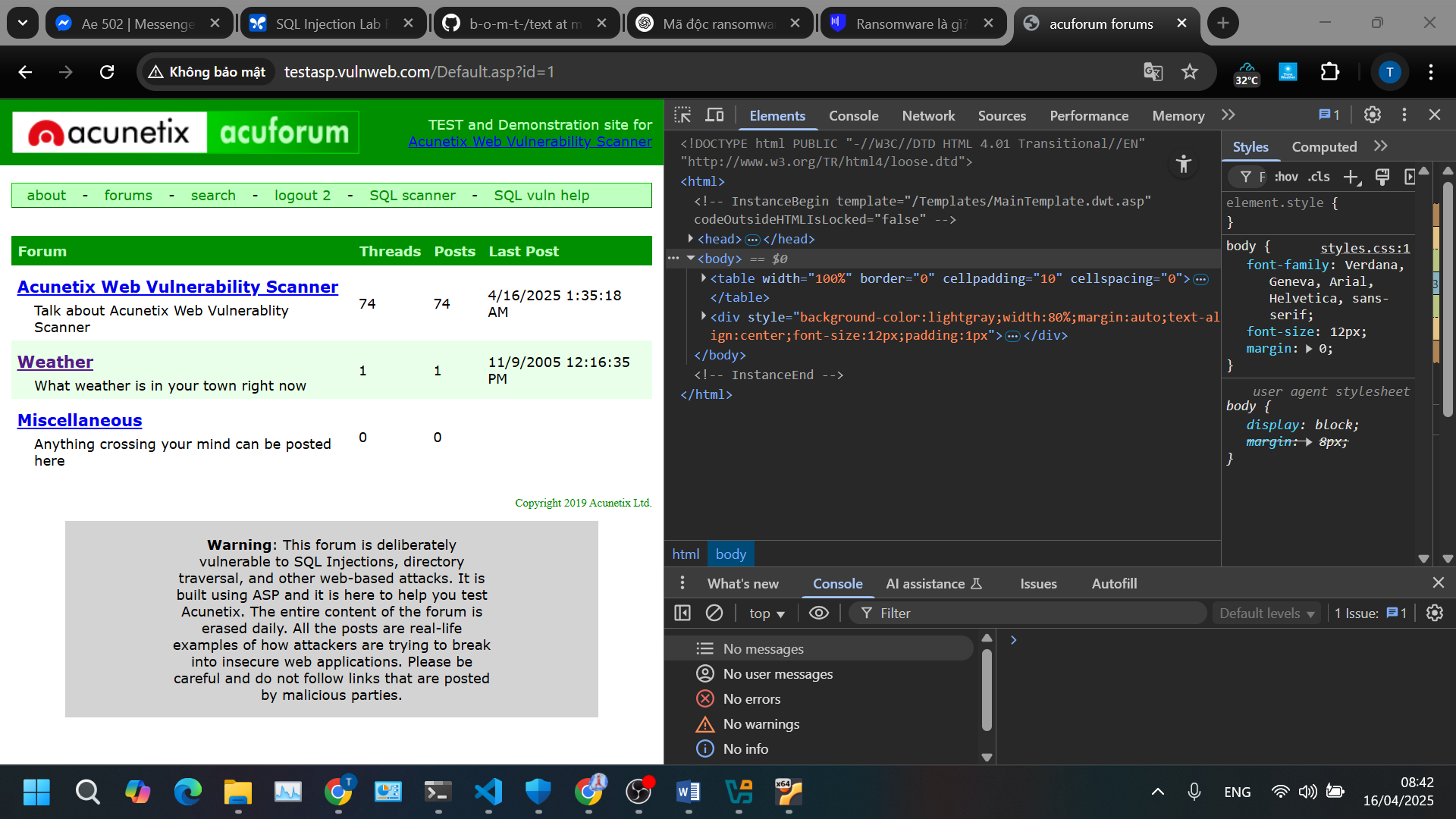
Kết quả: Nếu website không xử lý input đúng cách, câu truy vấn có thể trở thành:

SELECT \* FROM users WHERE username = '[username]' AND password = '' OR 1=1#'

Tương tự, bypass kiểm tra password và có thể đăng nhập thành công.

Mức độ ảnh hưởng: Rất cao, tương tự như username.

**Câu 3**



**Tham số đầu vào** <http://testasp.vulnweb.com/Default.asp?id=1>

Dựa vào URL, rất có thể có một tham số được truyền qua URL. Ví dụ: http://testasp.vulnweb.com/Default.asp?id=1. Tên tham số (trong ví dụ này là id) cần được xác định bằng cách quan sát phản hồi của trang web khi thay đổi giá trị tham số.

**Cách thức gửi** Phương thức GET, dữ liệu được gửi qua URL.

**Phân tích và phán đoán kiểu truy vấn, vị trí của các tham số đầu vào được sử dụng trong câu truy vấn**

Vì không có thông tin về chức năng cụ thể của trang Default.asp, chúng ta cần giả định dựa trên các trang web động thông thường. Một giả định hợp lý là trang này hiển thị một mục cụ thể (ví dụ: một sản phẩm, một bài viết) dựa trên một ID. Do đó, truy vấn có thể là:

SELECT \* FROM items WHERE id = [id]

Giá trị của tham số id được sử dụng trong mệnh đề WHERE của câu lệnh SELECT.

**Xác định mức độ ảnh hưởng của lỗ hổng trên mỗi giá trị đầu vào**