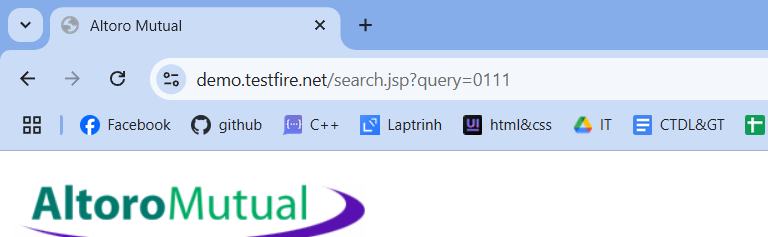
Câu 1:

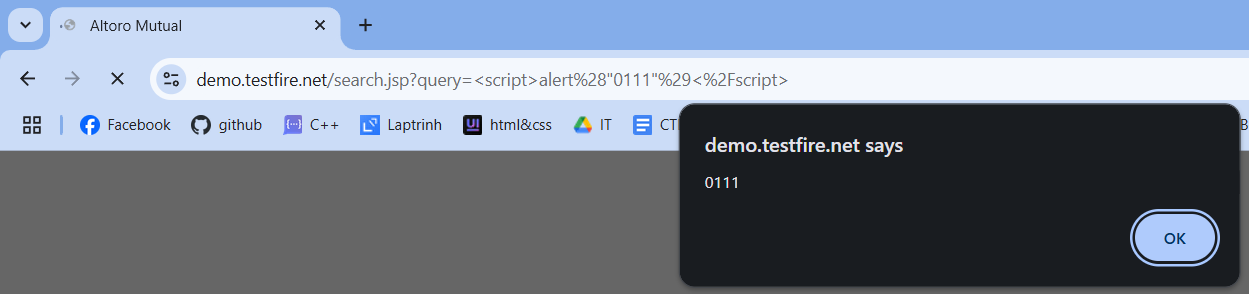


Tham số đầu vào lả1: query

Giá trị đầu vào là: 0111

Các bước kiểm thử và phán đoán:





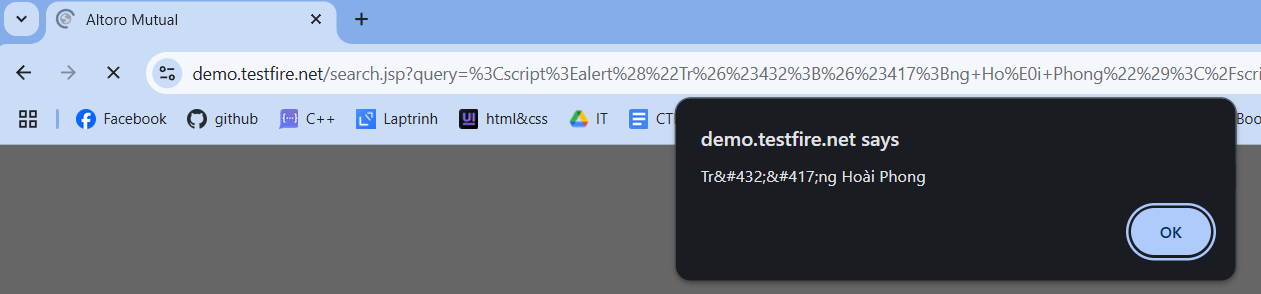
→ **lỗ hổng XSS tồn tại**.

Giá trị kiểm thử cho thấy lỗ hổng và giải thích:

<script>alert("0111")</script>

**Giải thích:**  
Khi giá trị trên được nhập và kết quả hiển thị ra thông báo chứa 0111, điều này chứng minh rằng **đoạn mã JavaScript được thực thi từ phía người dùng**, cho thấy website có lỗ hổng **Reflected XSS**.

Câu 2:

  
Tham số đầu vào lả: query

Giá trị đầu vào là: Dương Vĩnh Toàn

Các bước kiểm thử và phán đoán:

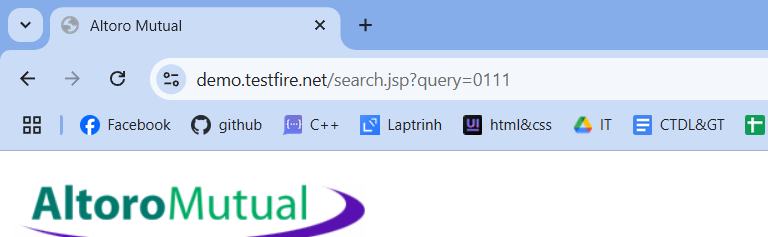
→ **lỗ hổng XSS tồn tại**.

Giá trị kiểm thử cho thấy lỗ hổng và giải thích:

<script>alert("Trần Thái Thiên ")</script>

**Giải thích:**  
Khi giá trị trên được nhập và kết quả hiển thị ra thông báo chứa “Trương Hoài Phong”, điều này chứng minh rằng **đoạn mã JavaScript được thực thi từ phía người dùng**, cho thấy website có lỗ hổng **Reflected XSS**.

Câu 3:

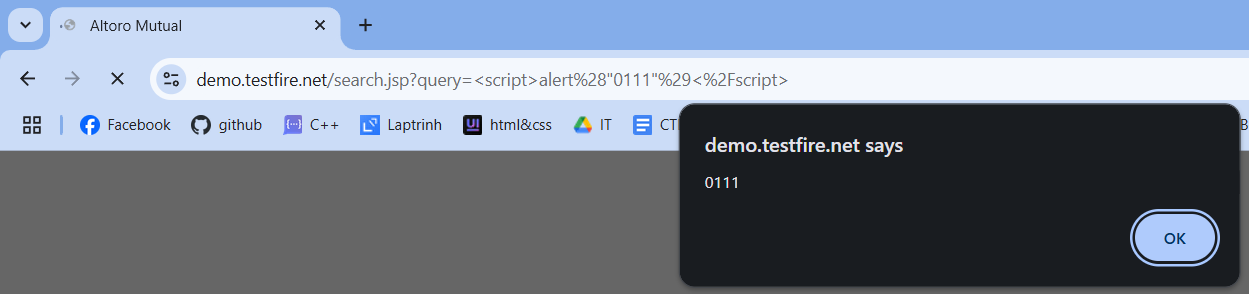


Tham số đầu vào lả1: query

Giá trị đầu vào là: 0111

Các bước kiểm thử và phán đoán:





→ **lỗ hổng XSS tồn tại**.

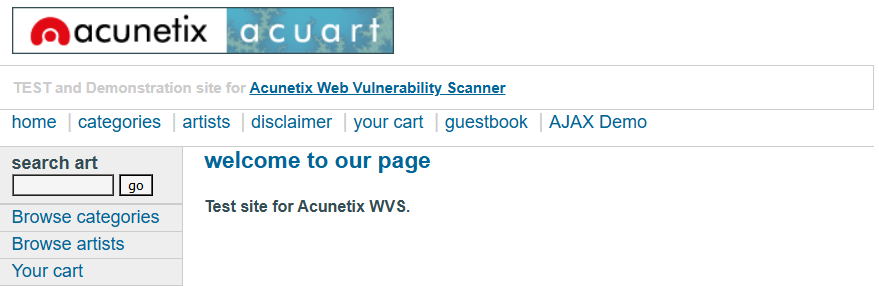
Giá trị kiểm thử cho thấy lỗ hổng và giải thích:

<script>alert("0111")</script>

**Giải thích:**  
Khi giá trị trên được nhập và kết quả hiển thị ra thông báo chứa 0111, điều này chứng minh rằng **đoạn mã JavaScript được thực thi từ phía người dùng**, cho thấy website có lỗ hổng **Reflected XSS**.

Câu 4:

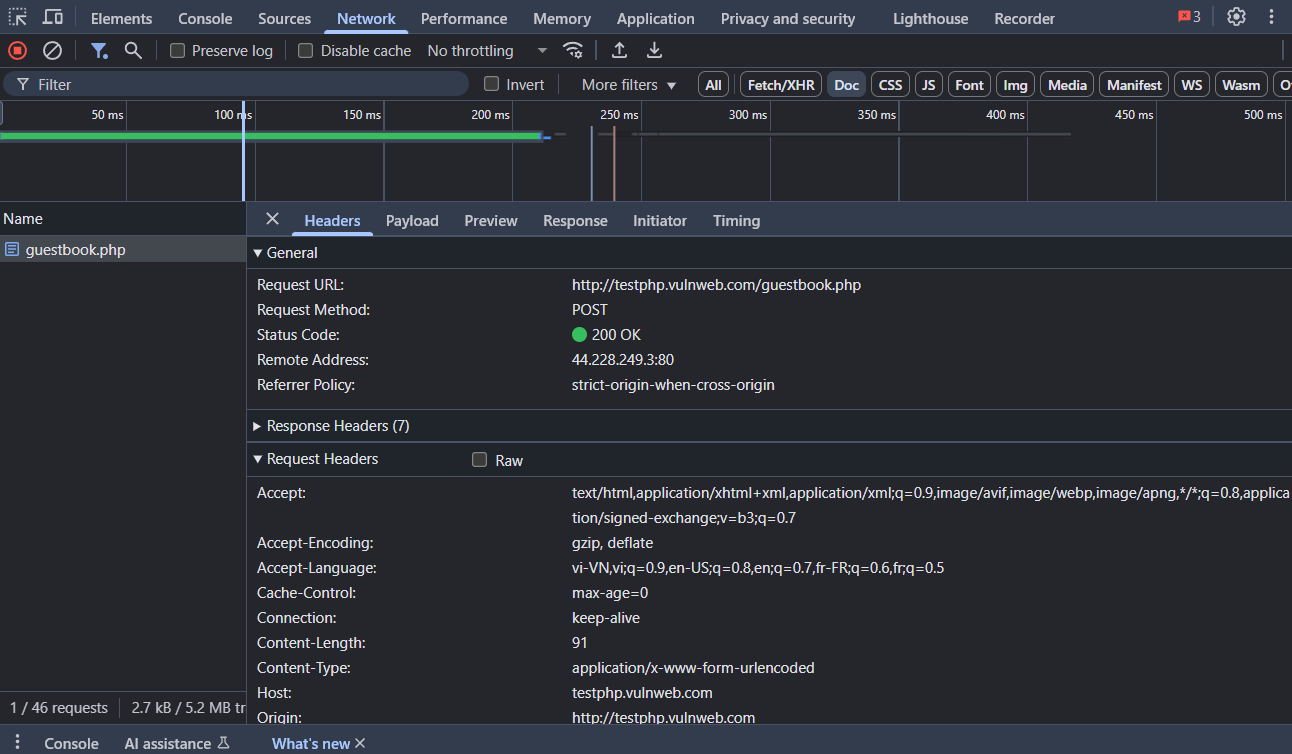
Truy cập vào trang http://testphp.vulnweb.com



Chọn guestbook

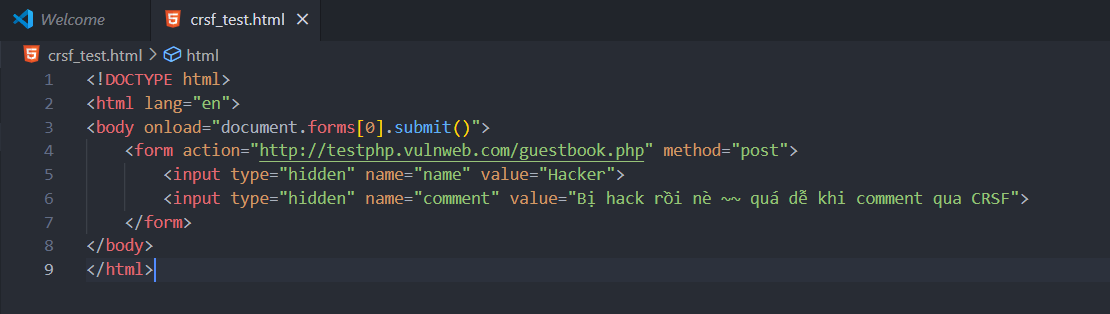
Test thử với 1 comment và submit

F12 lên, chọn netword coi có phần vừa comment không



Như bên dưới đã thấy comment ban nãy của tôi

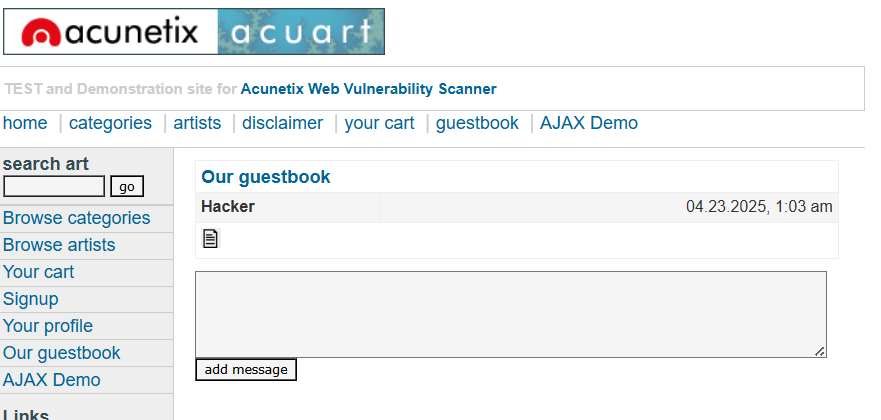
Tạo một file crsf\_test để tấn công crsf thử bằng đoạn mã sau



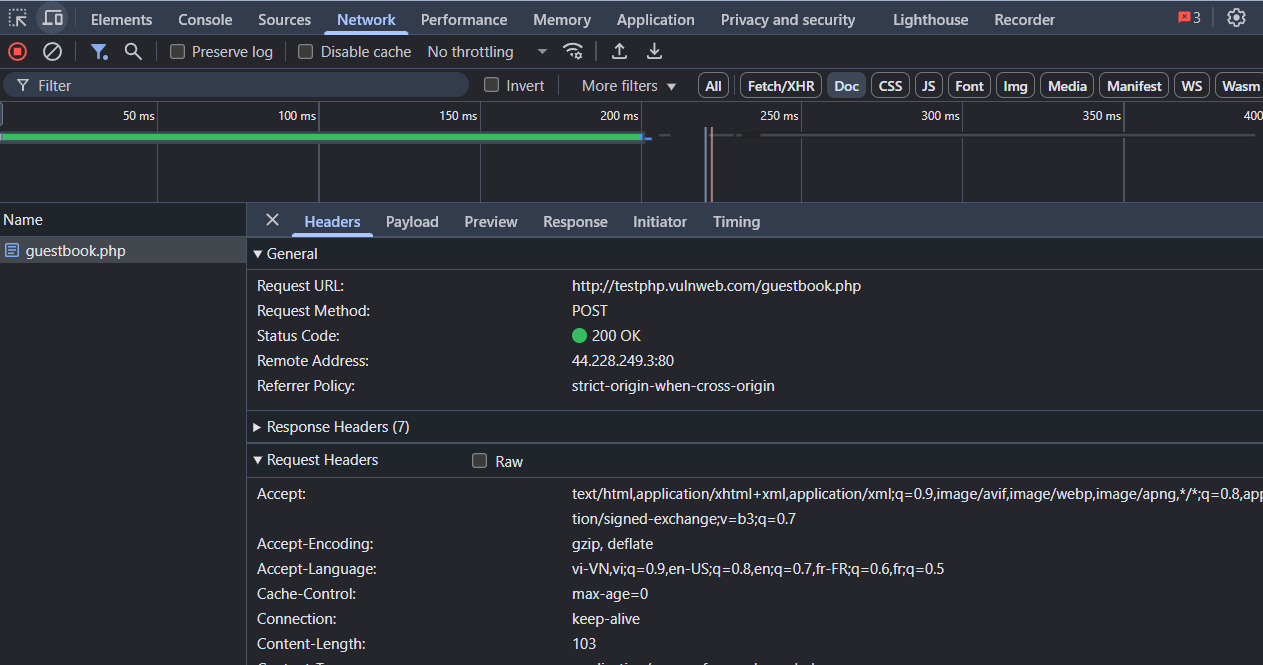
Lưu lại, và chạy file bên dưới bằng cách double click



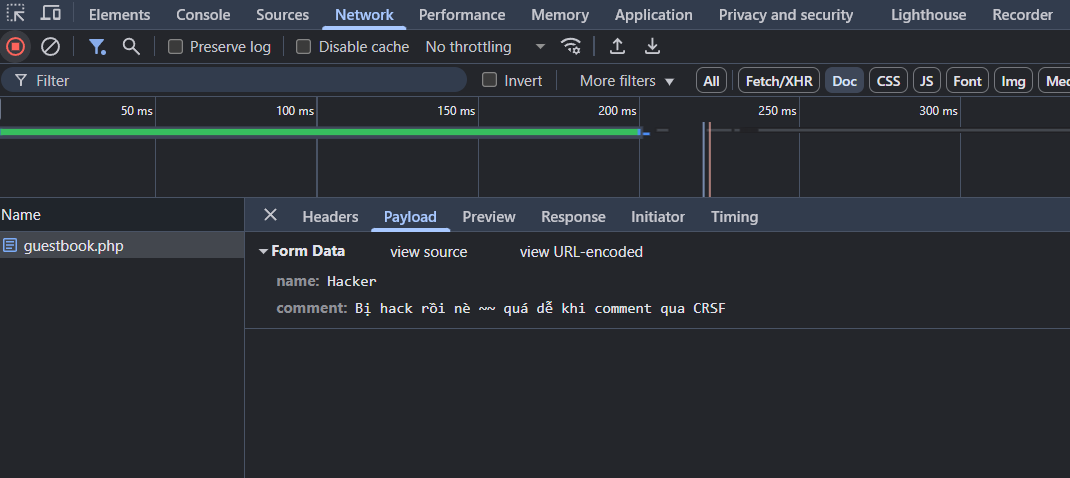
Như đã thấy bên dưới có file Hacker



Tiếp tục F12 để kiểm tra như phần test thử bên trên



Kết quả đã tấn công được



Kết quả kiểm thử

Website dễ bị tấn công CSRF:

* Khi người dùng (đã từng truy cập trang web) mở file HTML giả mạo (csrf\_test.html), form được gửi tự động mà không cần tương tác từ người dùng.
* Một bình luận giả mạo với tên là "Hacker" và nội dung "Bị hack rồi nè ~~ quá dễ khi comment qua CRSF" đã được thêm vào sổ lưu bút mà không cần sự đồng ý của người dùng.

Lý do lỗ hổng tồn tại:

* Trang guestbook.php không có cơ chế xác thực CSRF token.
* Không kiểm tra nguồn gửi yêu cầu (Referer, Origin).
* Không có bất kỳ biện pháp bảo vệ người dùng khỏi việc gửi yêu cầu từ bên thứ ba.