**BÁO CÁO LAB 1**

*Môn: ATMMTNC*

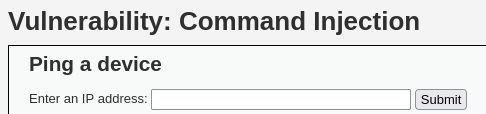
*GVTH: Đỗ Thị Phương Uyên*

|  |  |
| --- | --- |
| Sinh viên thực hiện | **Sinh viên 1**  MSSV: 21521520  Họ tên: Huỳnh Minh Tân Tiến  **Sinh viên 2**  MSSV: 21521817  Họ tên: Bùi Hoàng Trúc Anh  **Sinh viên 3**  MSSV: 21521253  Họ tên: Lê Hoàng Oanh  **Sinh viên 4**  MSSV: 21520353  Họ tên: Nguyễn Ngọc Trà My |
| Lớp | **NT534.O21.ATCL** |
| Tổng thời gian thực hiện Lab trung bình |  |
| Phân chia công việc  *(nếu là nhóm)* | **[Sinh viên 1]**:  **[Sinh viên 2]**: |
| Link Video thực hiện *(nếu có yêu cầu)* |  |
| Ý kiến *(nếu có)*  *+ Khó khăn gặp phải + Đề xuất, góp ý…* |  |
| Điểm tự đánh giá *(bắt buộc)* | 10/**10** |

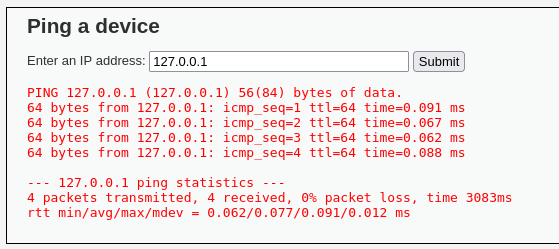
*[Nội dung báo cáo chi tiết – Trình bày tùy sinh viên, Xuất file .PDF khi nộp]*

1. **DVWA**
2. **Command Injection**

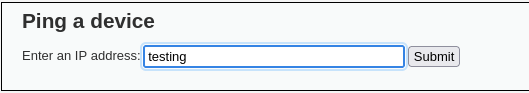
Khi ta mở thử thách “Command Injection”, ta sẽ thấy một dụng cụ dung để ping tới địa chỉ IP.

****

Ta thử nhập địa chỉ IP hợp lệ là 127.0.0.1



Và ta thử nhập địa chỉ IP không hợp lệ là testing thì không có hiện ra kết quả.



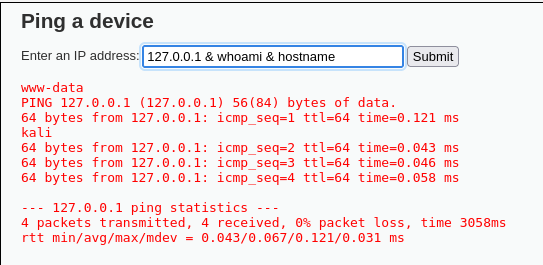
Vì vậy, có vẻ như code lấy input, chuyển nó làm tham số cho lệnh ping và thực thi chuỗi "ping [input]". Nếu nó hoạt động, ta sẽ thấy đầu ra của lệnh chưa được định dạng. Nếu không, chúng ta sẽ không thấy gì cả.

Để tiến hành tấn công, ta cần:

* Lệnh whoami để tìm hiểu người dùng.
* Lệnh tên máy chủ để tìm ra tên máy chủ.
* Ký tự “&” để kết hợp những điều trên.

Kết hợp những thứ đó lại với nhau, chúng ta có được chuỗi tấn công của mình:

127.0.0.1 & whoami & hostname

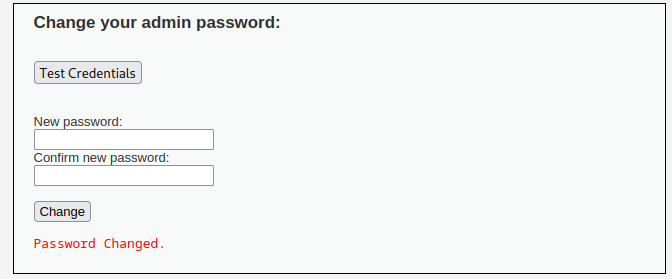


Ta thấy dòng lệnh đã được thực hiện thành công với dòng 1 là output của whoami và dòng 4 là output của hostname.



1. **CSRF**

CSRF (Cross Site Request Forgery) là kỹ thuật tấn công bằng cách sử dụng quyền chứng thực của người dùng đối với một website. CSRF là kỹ thuật tấn công vào người dùng, dựa vào đó hacker có thể thực thi những thao tác phải yêu cầu sự chứng thực.



Khi test thử form bằng cách đổi password thành ‘password1’, ta nhận thấy rằng không có bất kì yêu cầu chứng thực nào, ta có thể dễ dàng đổi password mà không cần biết password cũ.

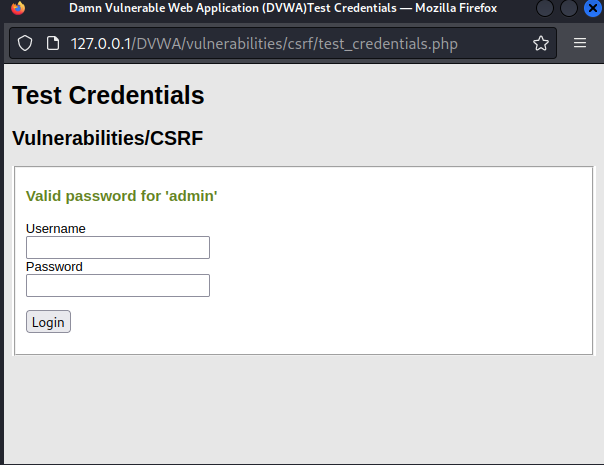
Sau đó, ta nhận thấy trên URL: <http://127.0.0.1/DVWA/vulnerabilities/csrf/password_new=password1&password_conf=password1&Change=Change>#

Ta tấn công bằng cách thay đổi [your\_new\_password] = password ta muốn:

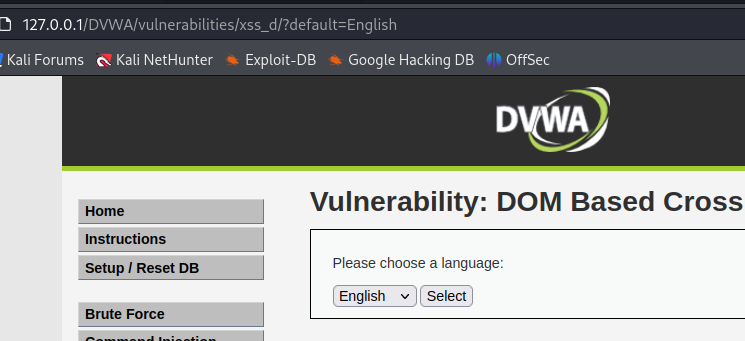
http://127.0.0.1/DVWA/vulnerabilities/csrf/password\_new= [your\_new\_password] &password\_conf= [your\_new\_password] &Change=Change#

Ta nhận thấy có thể thay đổi password mà không cần nhập vào form đổi password

Sau đó login lại bằng password đã đổi:

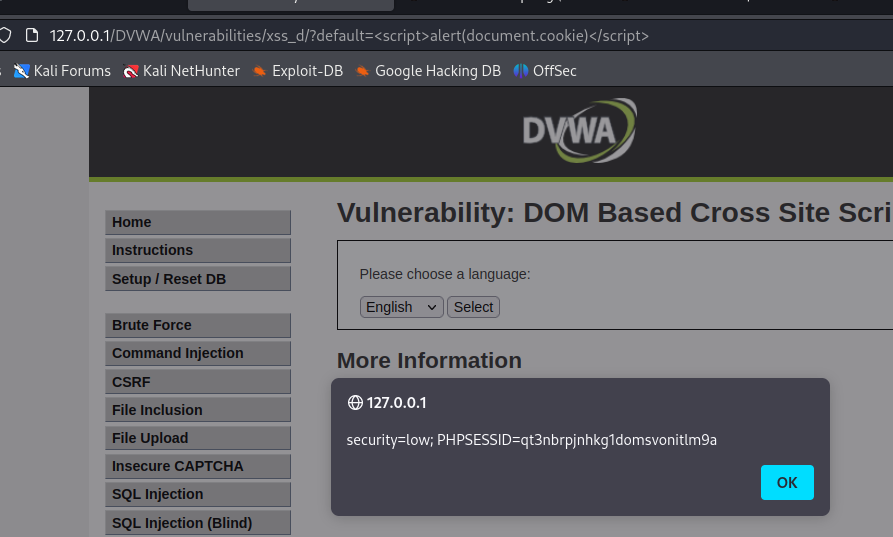


1. **XSS (DOM)**



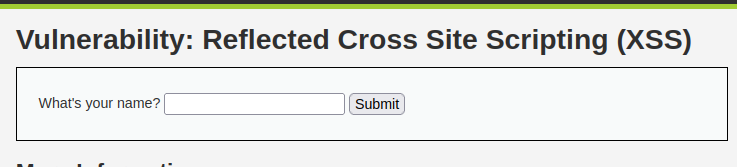
XSS type 0

Bằng cách chèn các payload độc vào url để tấn công vào dom của website, ta có thể lấy được các thông tin mong muốn bằng các đoạn code javascript, đoạn script dùng cho tấn công xss để lấy được cookie: <script>alert(document.cokkie)</script>

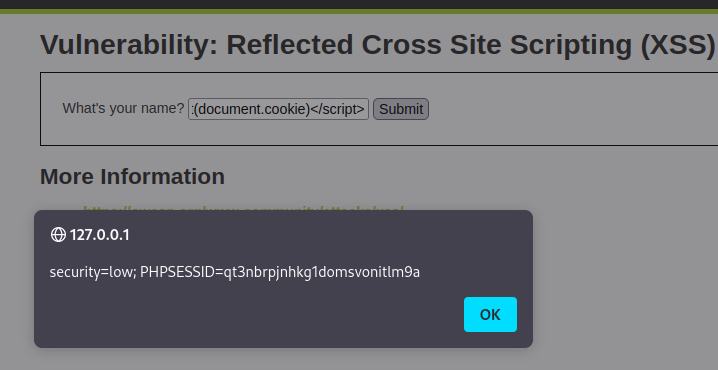


1. **XSS (Reflected)**

XSS type 1

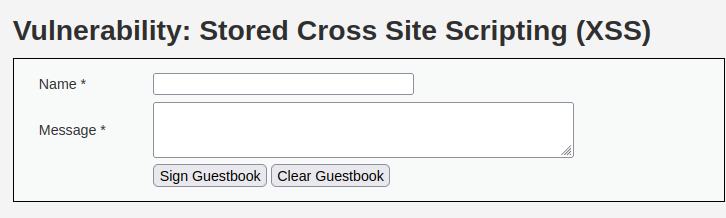


Tấn công XSS bằng cách sẽ chèn các đoạn script vào form submit để lấy thông tin cookie.



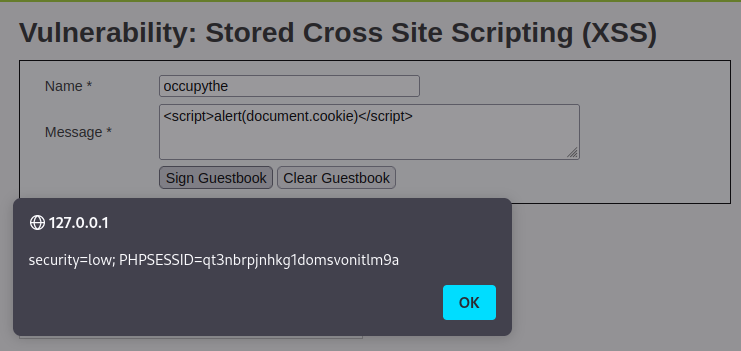
1. **XSS (Stored)**

XSS type 2



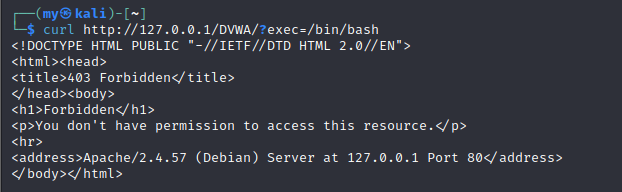
Tấn công vào các dạng thông tin được lưu trữ trên server, cũng bằng việc chèn các đoạn script để lấy các thông tin

Do trường name đã bị giới hạn về độ dài nên script sẽ được chèn ở trường message.

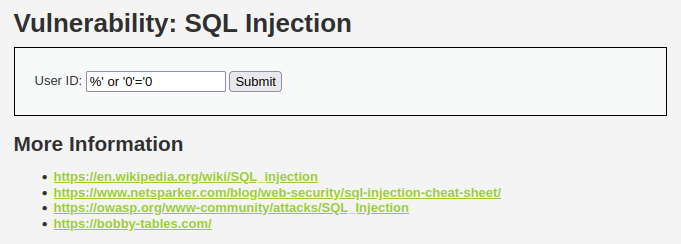


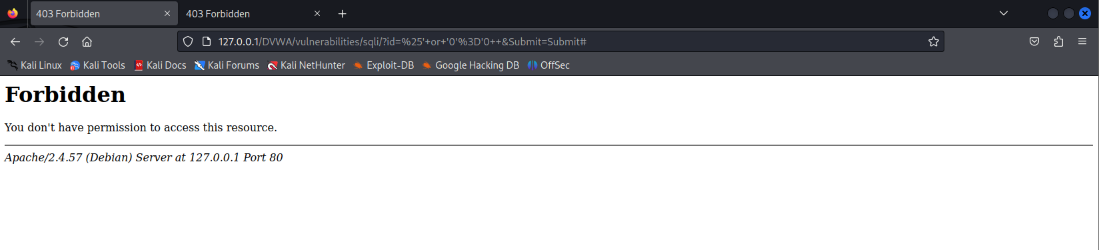
1. **ModSecurity**

Sau khi cấu hình ModSecurity, ta tiến hình test ModSecurity;



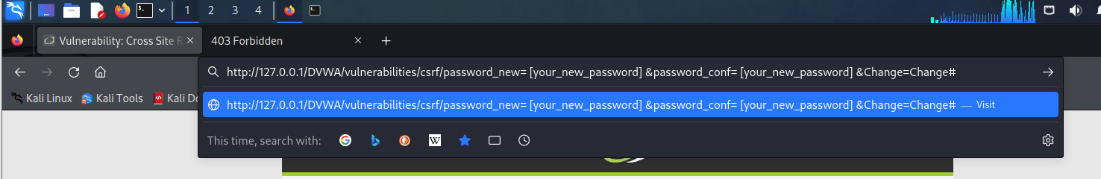
Tiếp theo, ta thử tiến hình tấn công như ở phần I

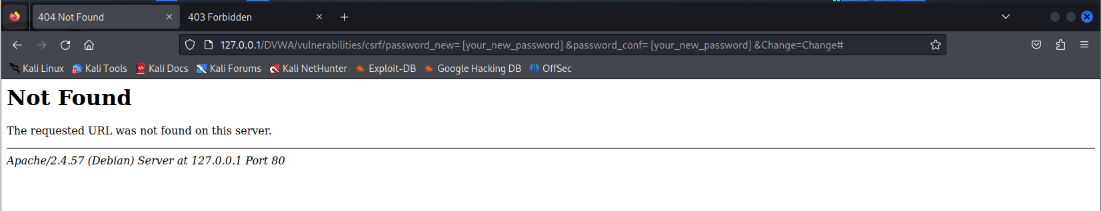




Ta nhận thấy SQL Injection đã được phát hiện và chặn bởi ModSecurity.

Tiếp theo, ta tiến hành tấn công CSRF, ta cũng thấy rằng tấn công đã bị chặn bởi ModSecurity





Ta thử tấn công Command Injection thì ta thấy không tấn công được

A screenshot of a computer

Description automatically generated