Hồ ĐÌnh Nguyên

MSSV:1050080025

Lớp :CNPM1

Linkyoutube:<https://www.youtube.com/channel/UCEPuvFCAI79ySOkCeCdgRyw>

Linkgithup:https://github.com/Dinhnguyen16122003/BaoMatMangMayTinhVaHeThong.git

Câu 1: Khai thác lỗi Cross-site scripting (XSS) trên DVWA

1.1. Tham khảo các tài liệu đính kèm để khai thác lỗi Dom-based XSS trên DVWA (security ở mức

độ low) để lấy cookie của người dùng đang đăng nhập.

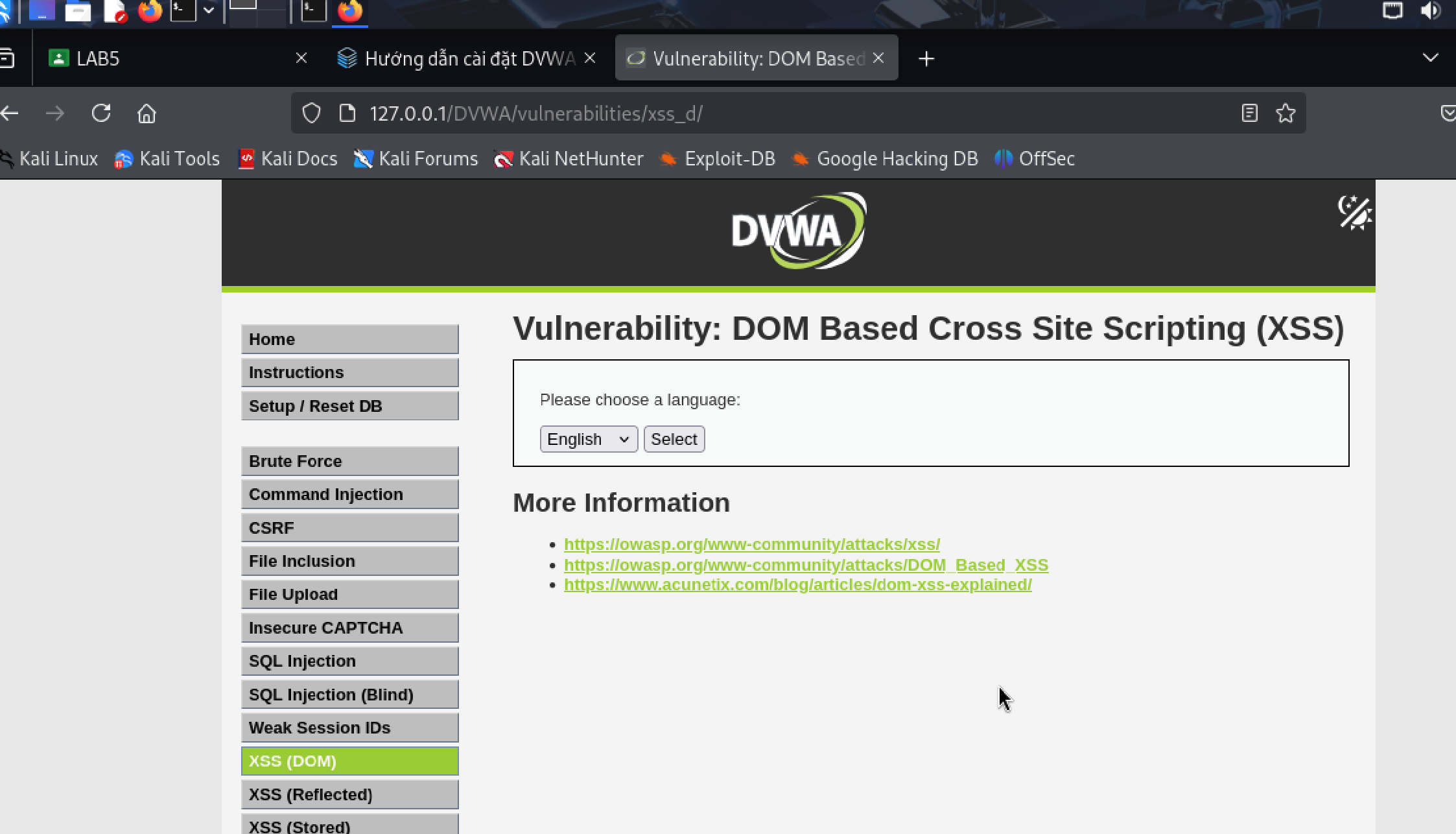
https://github.com/mrudnitsky/dvwa-guide-2019/blob/master/low/Challenge%2010:%20XSS%20(DOM).md

https://bughacking.com/dvwa-ultimate-guide-first-steps-and-walkthrough/#XSS\_DOM

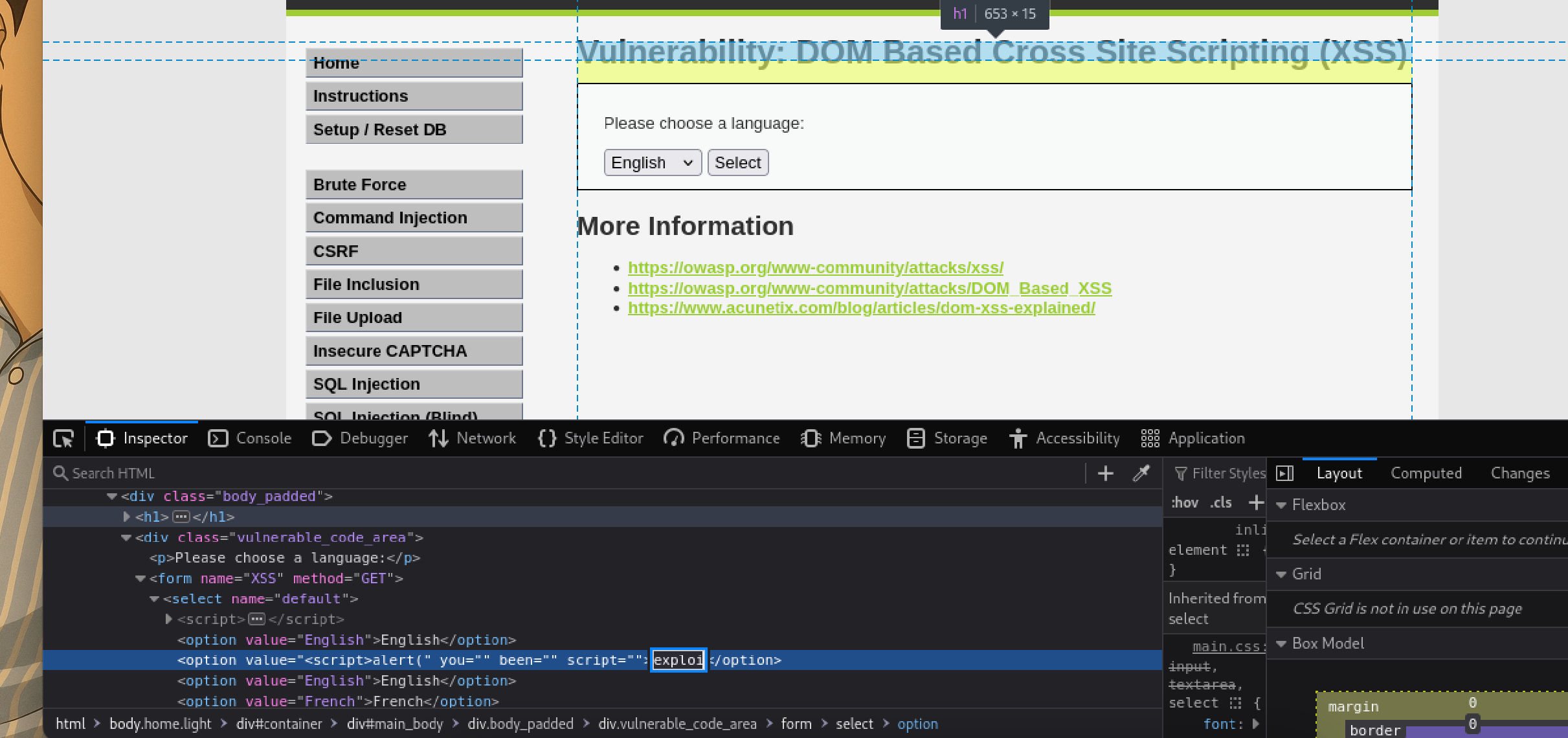
https://youtu.be/X87Ubv-qDm4?list=PLHUKi1UlEgOJLPSFZaFKMoexpM6qhOb4Q

(Chụp hình minh họa quá trình thực hiện)

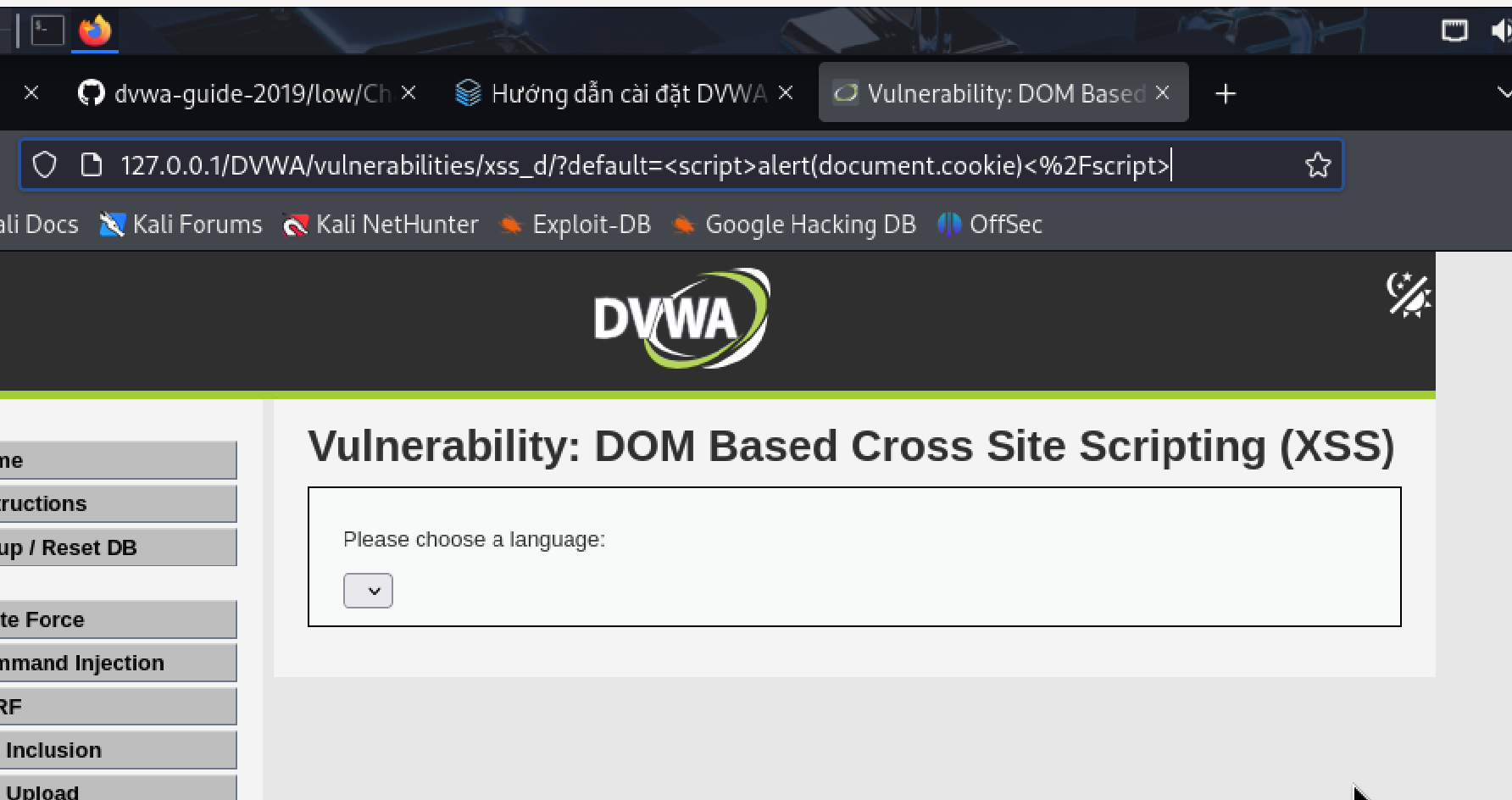
- Mở trang DVWA và chọn XSS, nhập thanh địa chỉ như trong hình bên dưới:

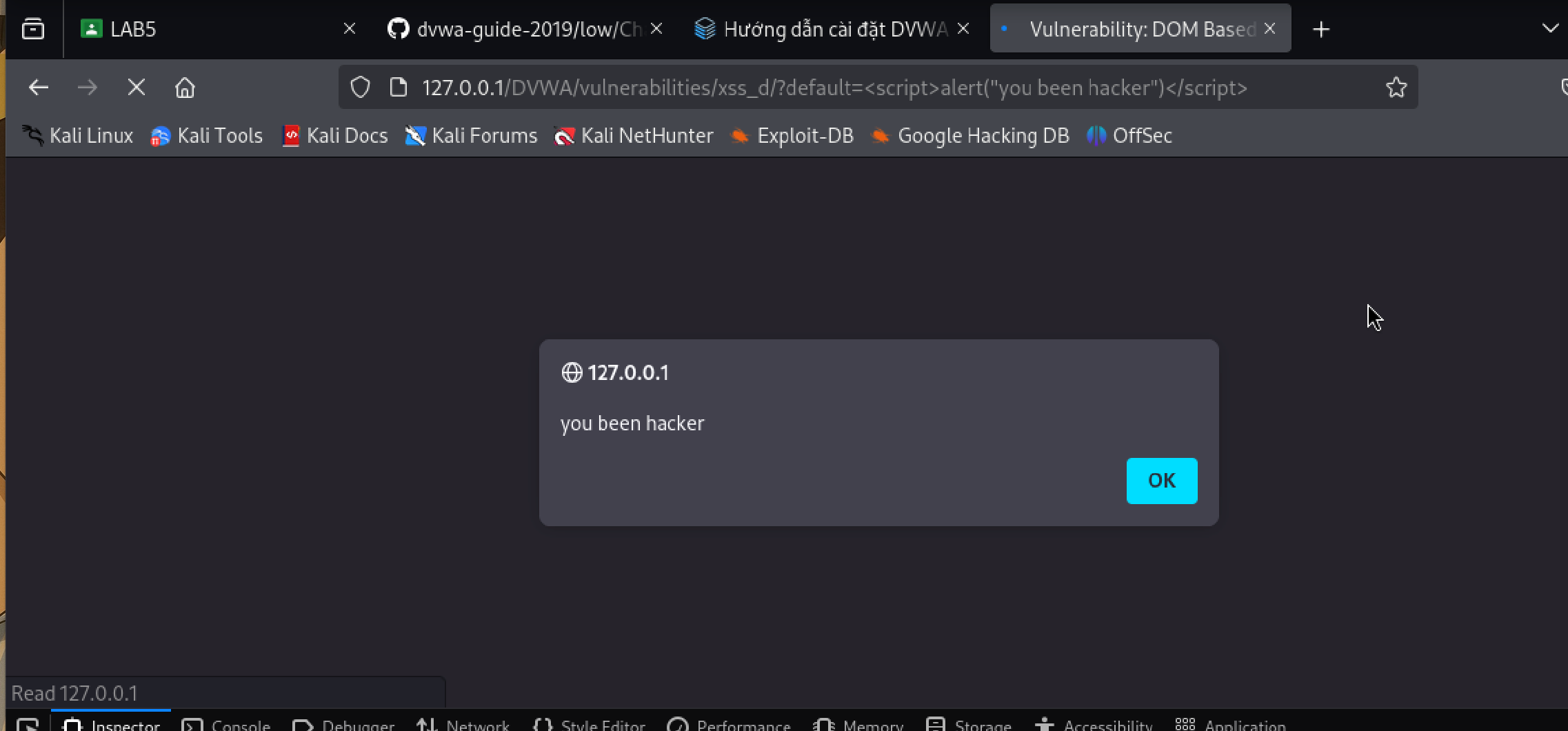


-mở mà nguồn và chỉnh sửa:

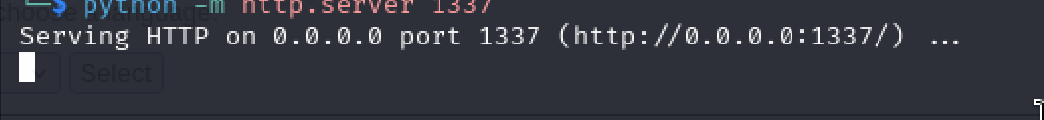


- Kết quả cookie hiện lên:

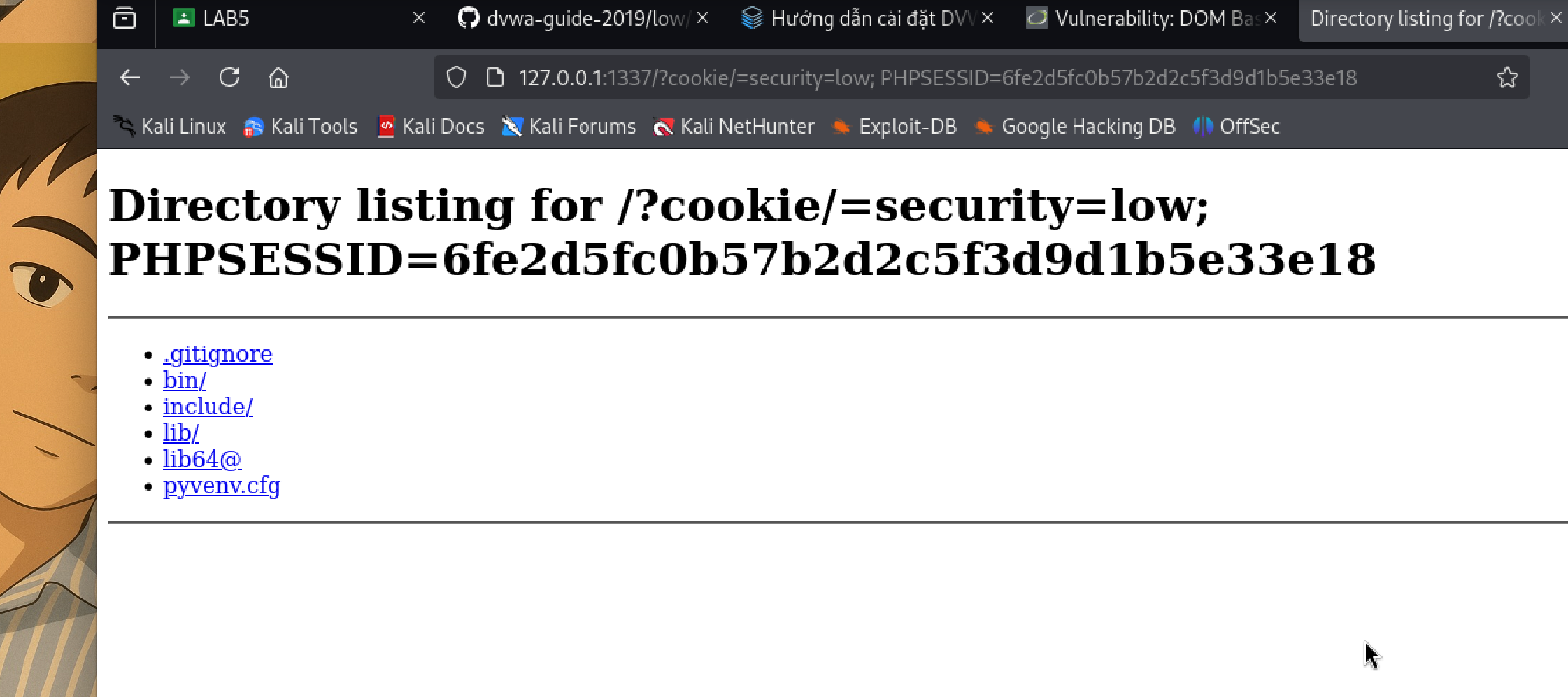




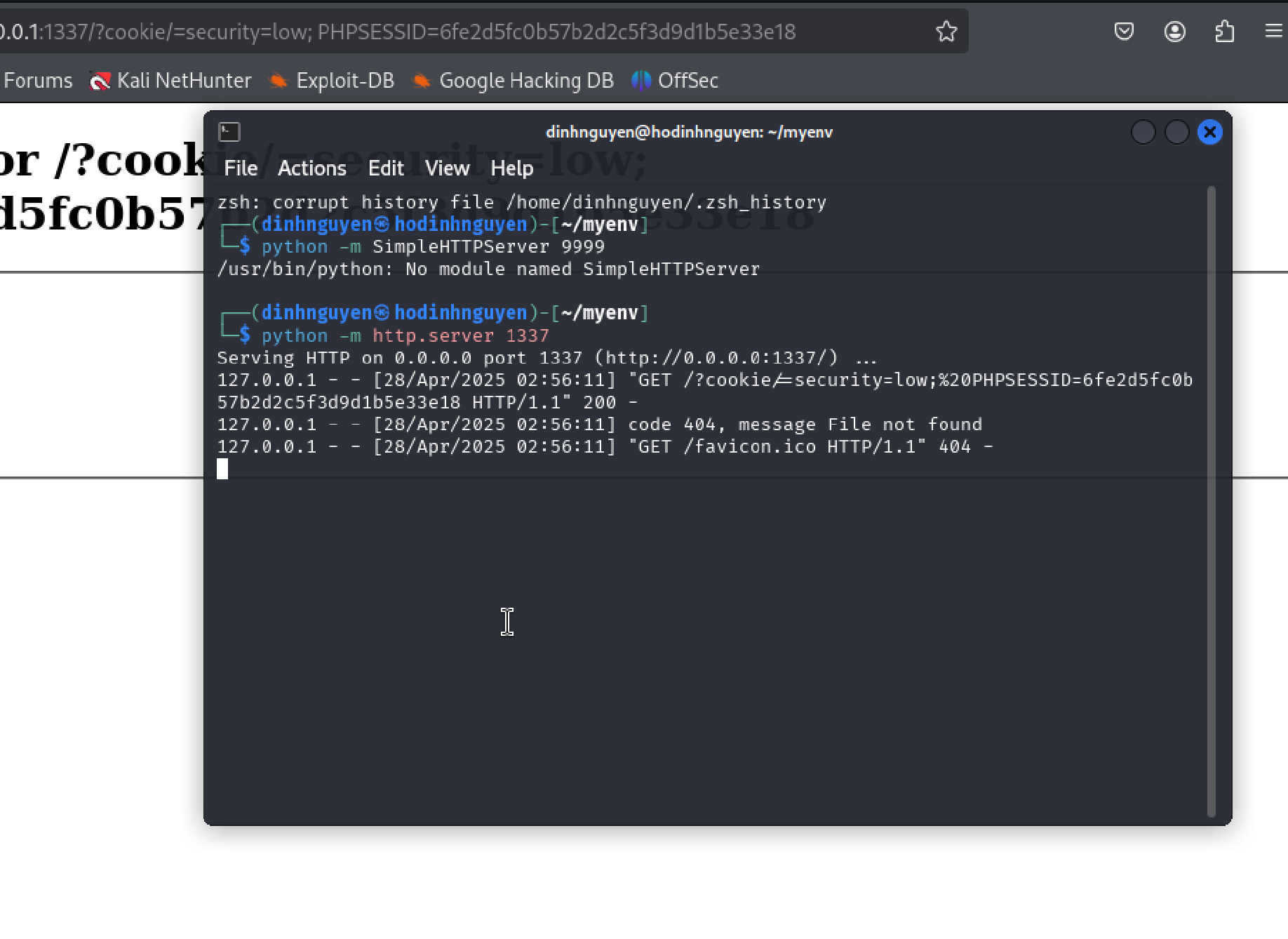
- Chạy server:



- Chèn script vào và chạy:



hiển thị



1.2. Tham khảo các tài liệu đính kèm để khai thác lỗi Stored XSS trên DVWA (security ở mức độ

low) để lấy cookie của người dùng đang đăng nhập. Sau đó dùng công cụ Burp Suite (hoặc chức

năng Inspect trên Firefox) để truy cập vào DVWA sử dụng cookie vừa lấy được.

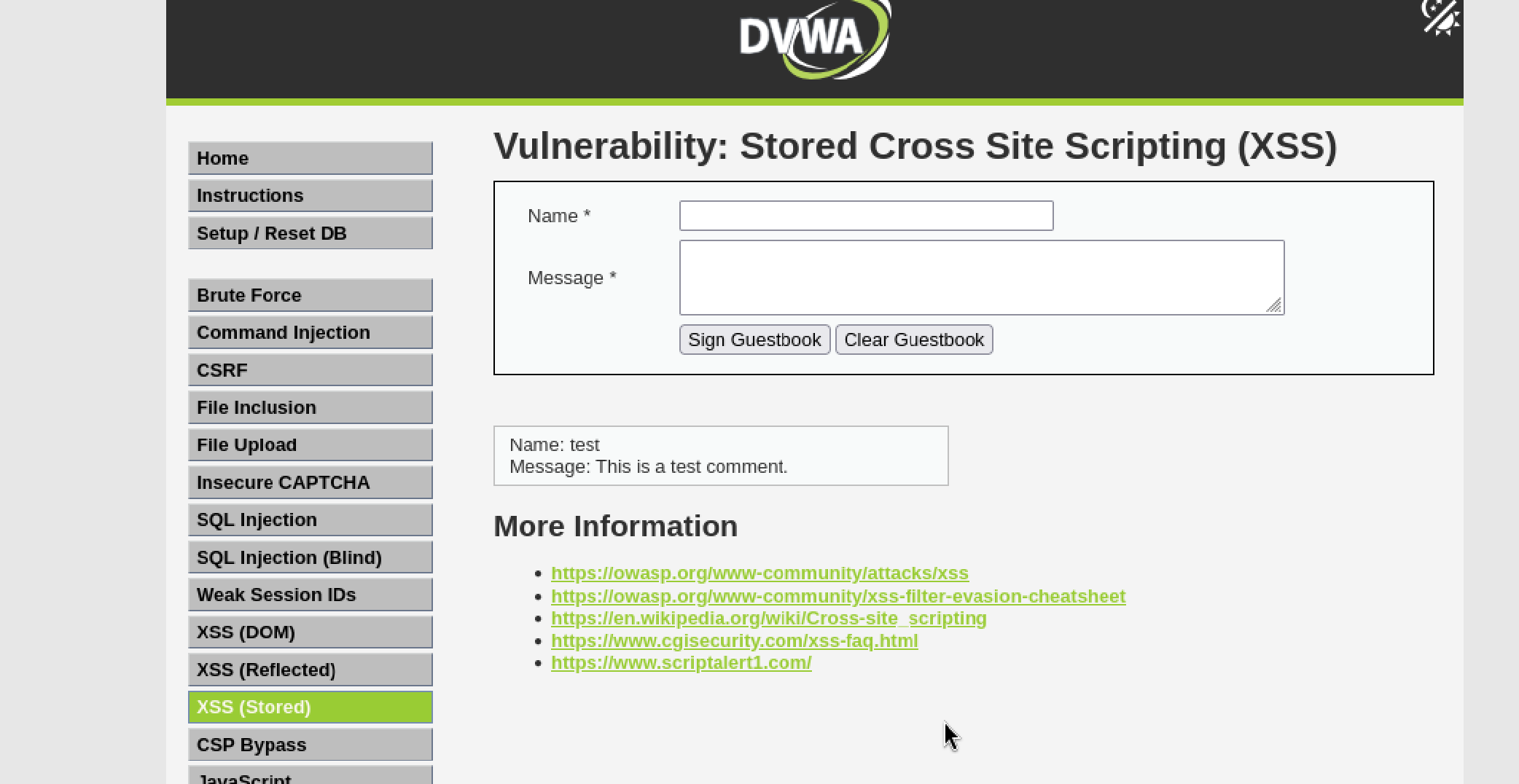
https://github.com/mrudnitsky/dvwa-guide-2019/blob/master/low/Challenge%2012:%20XSS%20(Stored).md

https://bughacking.com/dvwa-ultimate-guide-first-steps-and-walkthrough/#XSS\_Stored

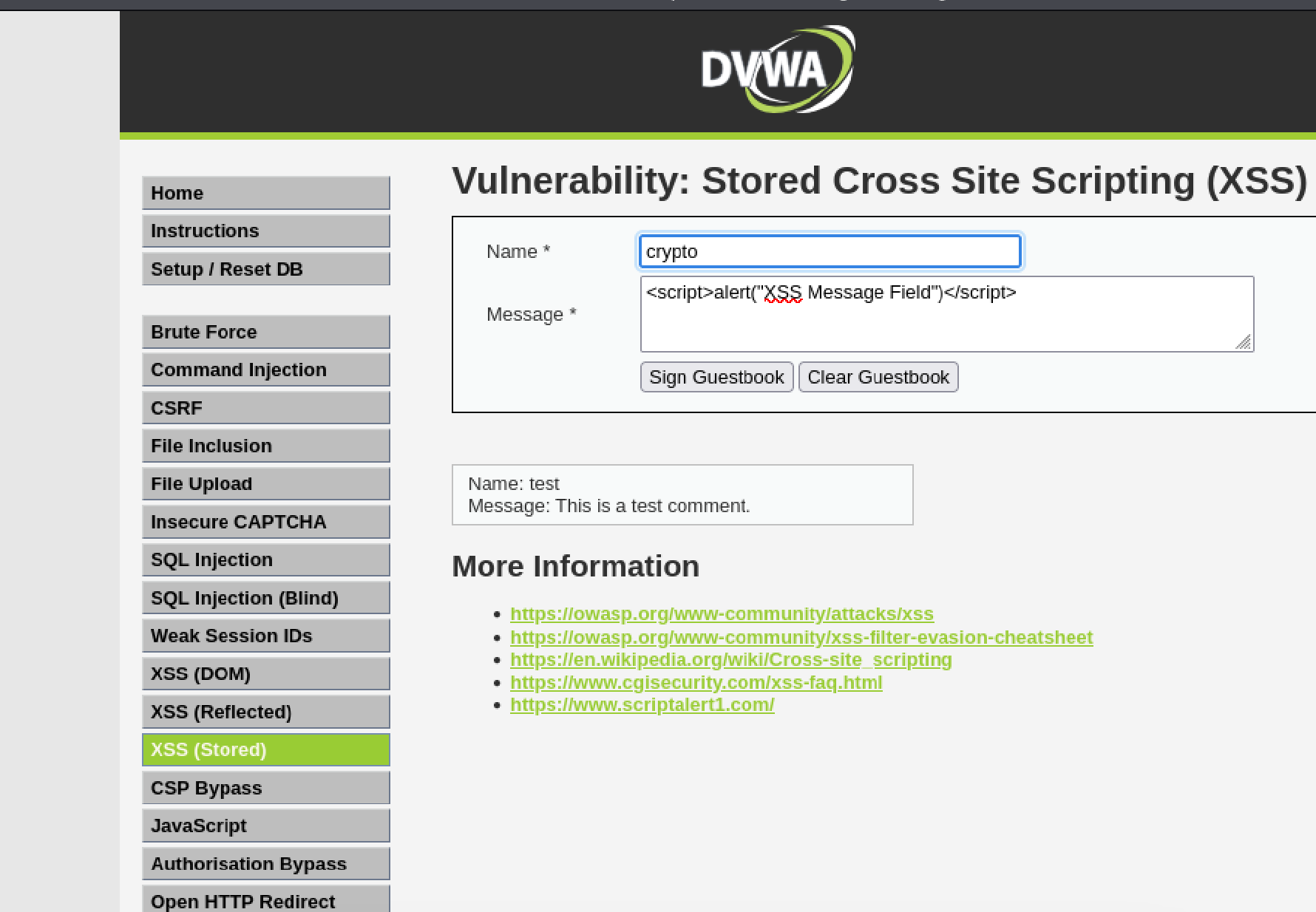
https://youtu.be/P1I9UGpGdrU?list=PLHUKi1UlEgOJLPSFZaFKMoexpM6qhOb4Q

(Chụp hình minh họa quá trình thực hiện)

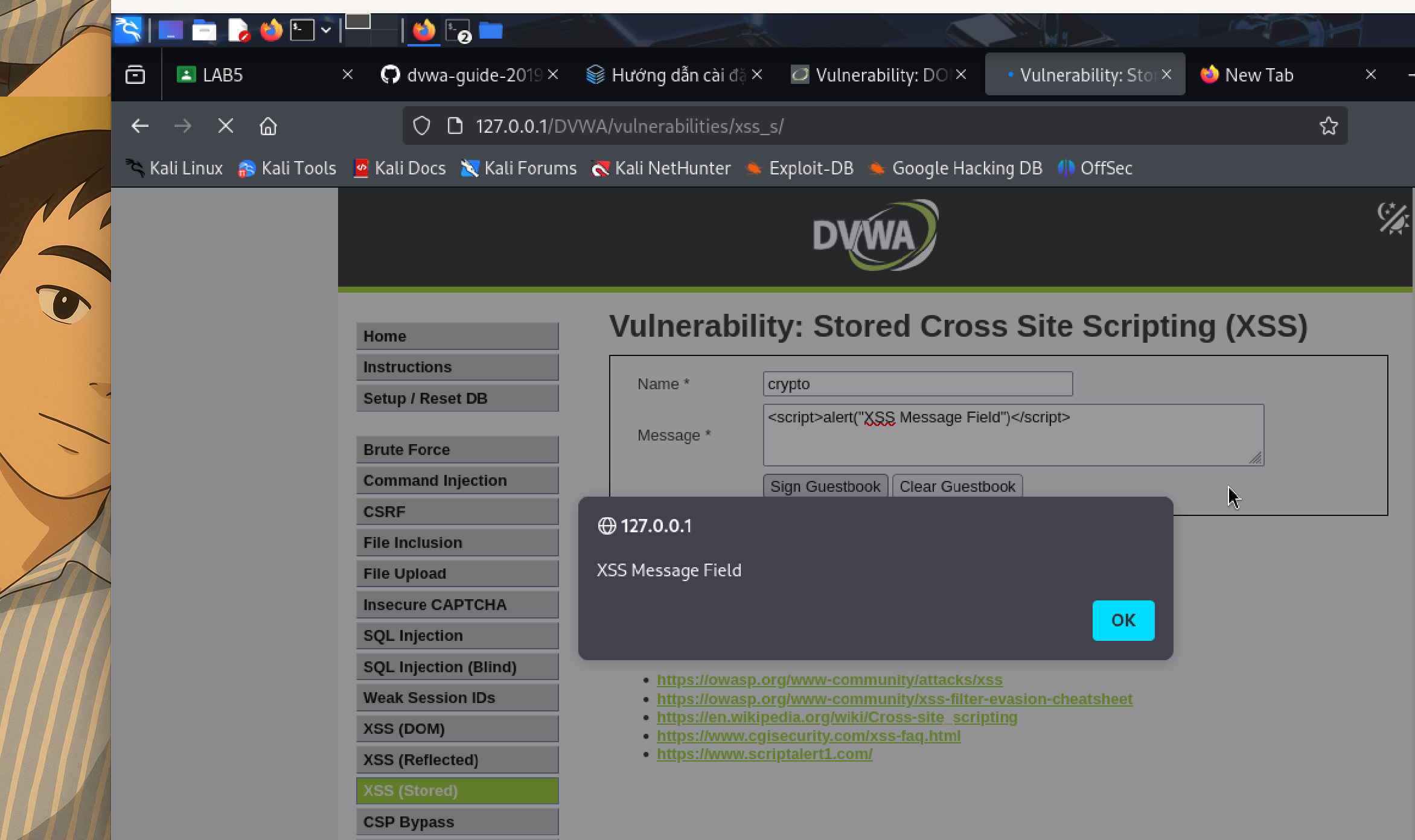
-vào dwva



-nhập <script>alert("XSS Message Field")</script>

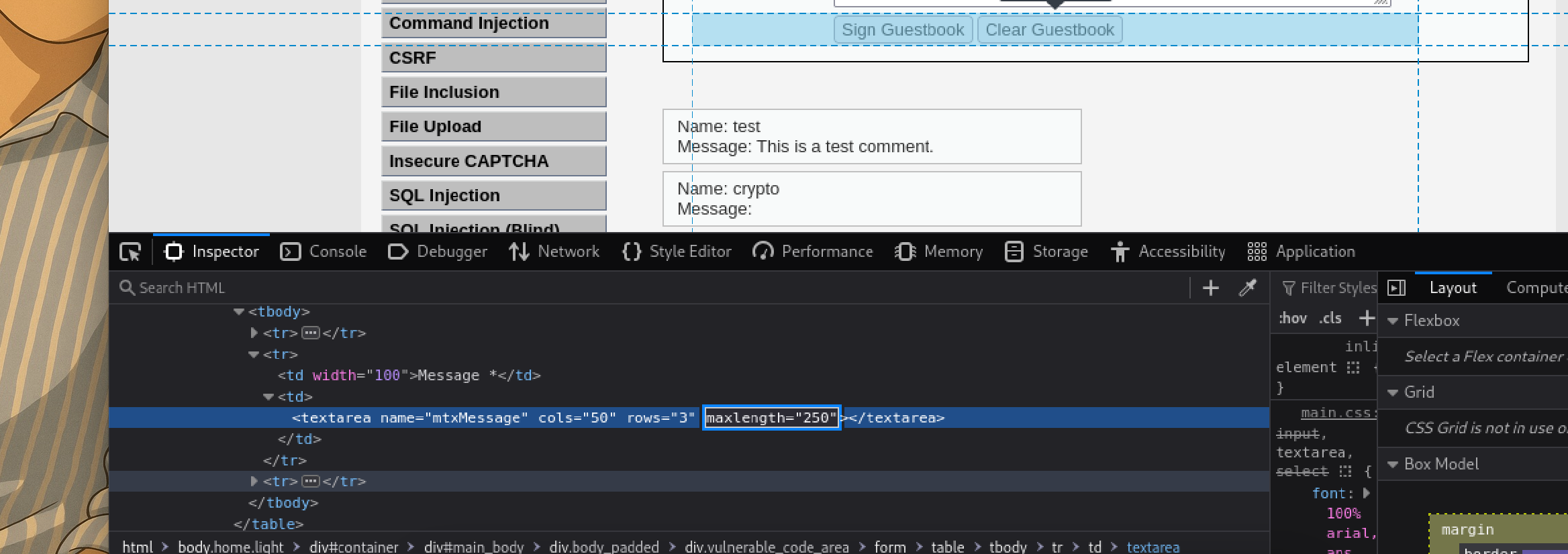


-hiển thị cookie

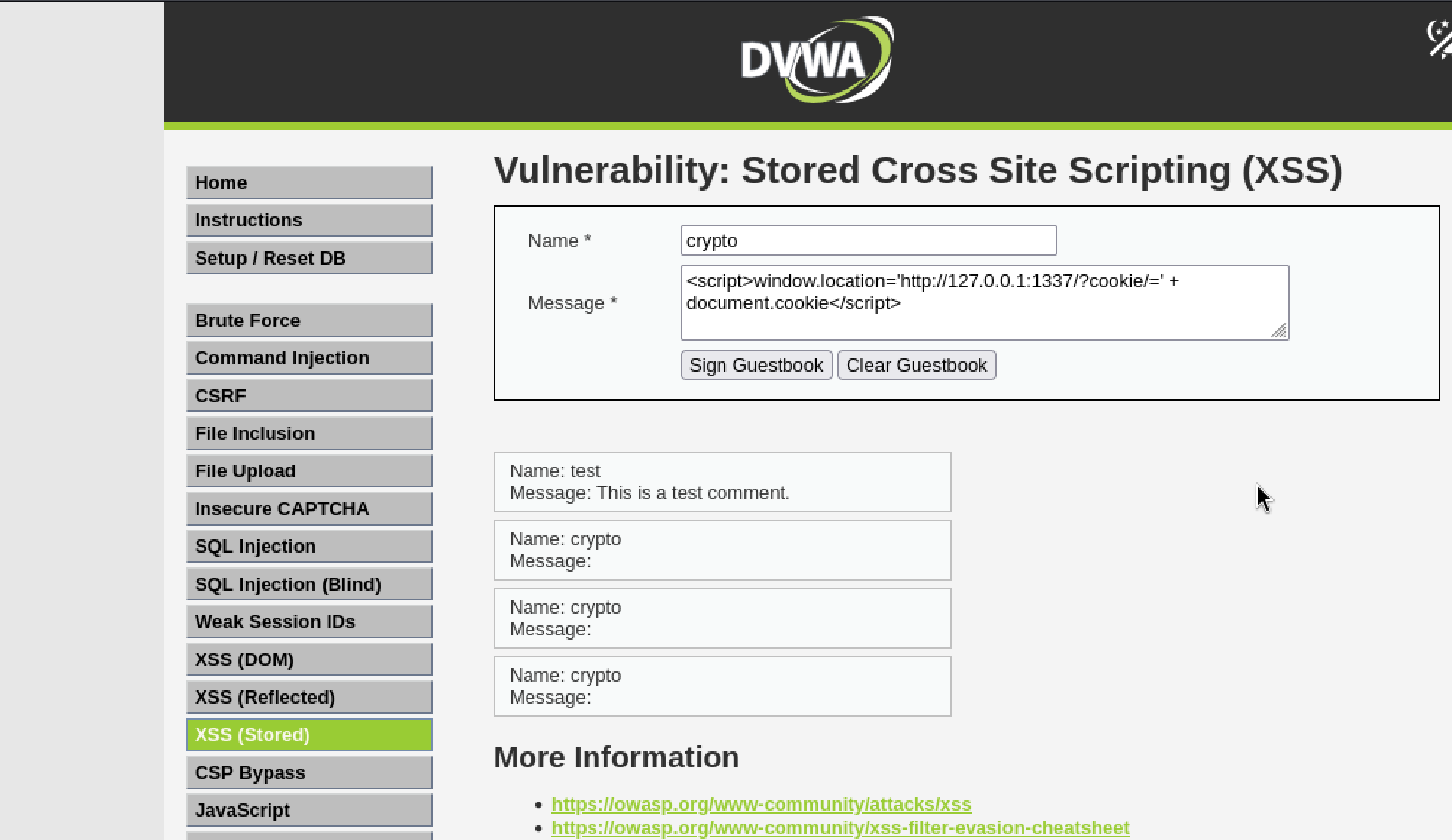


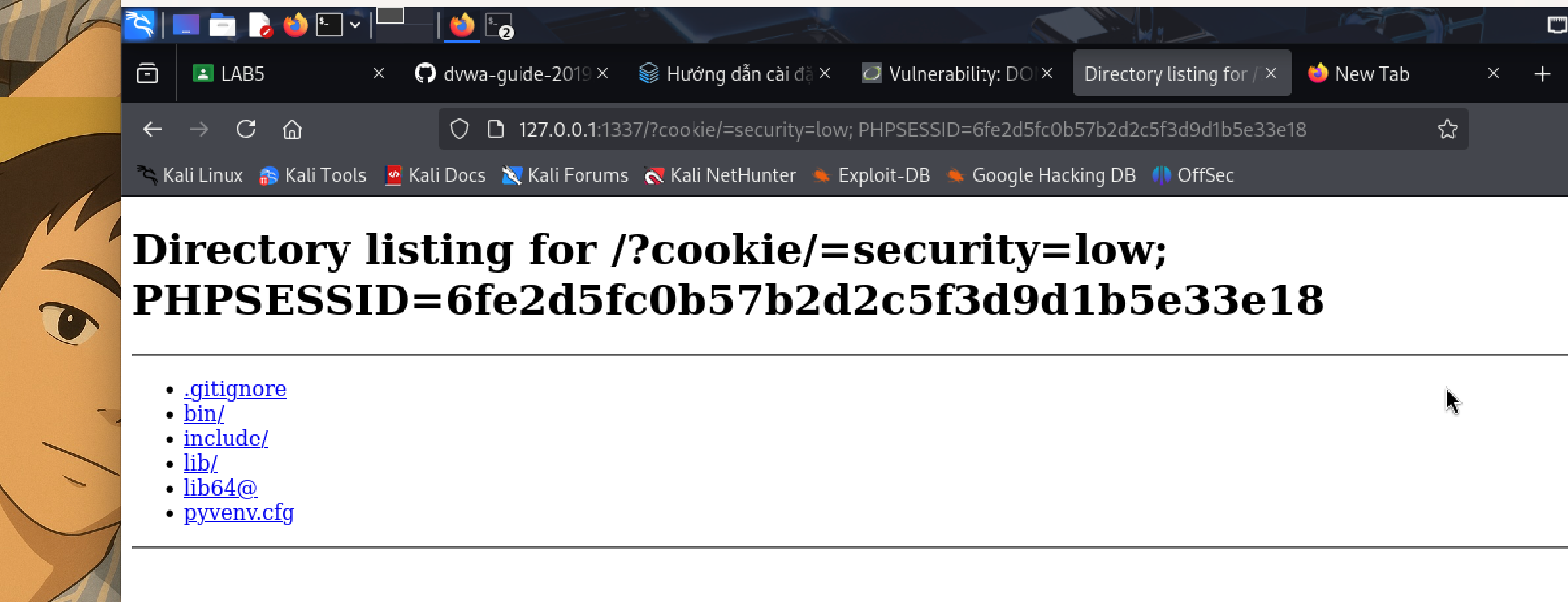
-chạy server 1337

-chỉnh sửa 50 thành 250

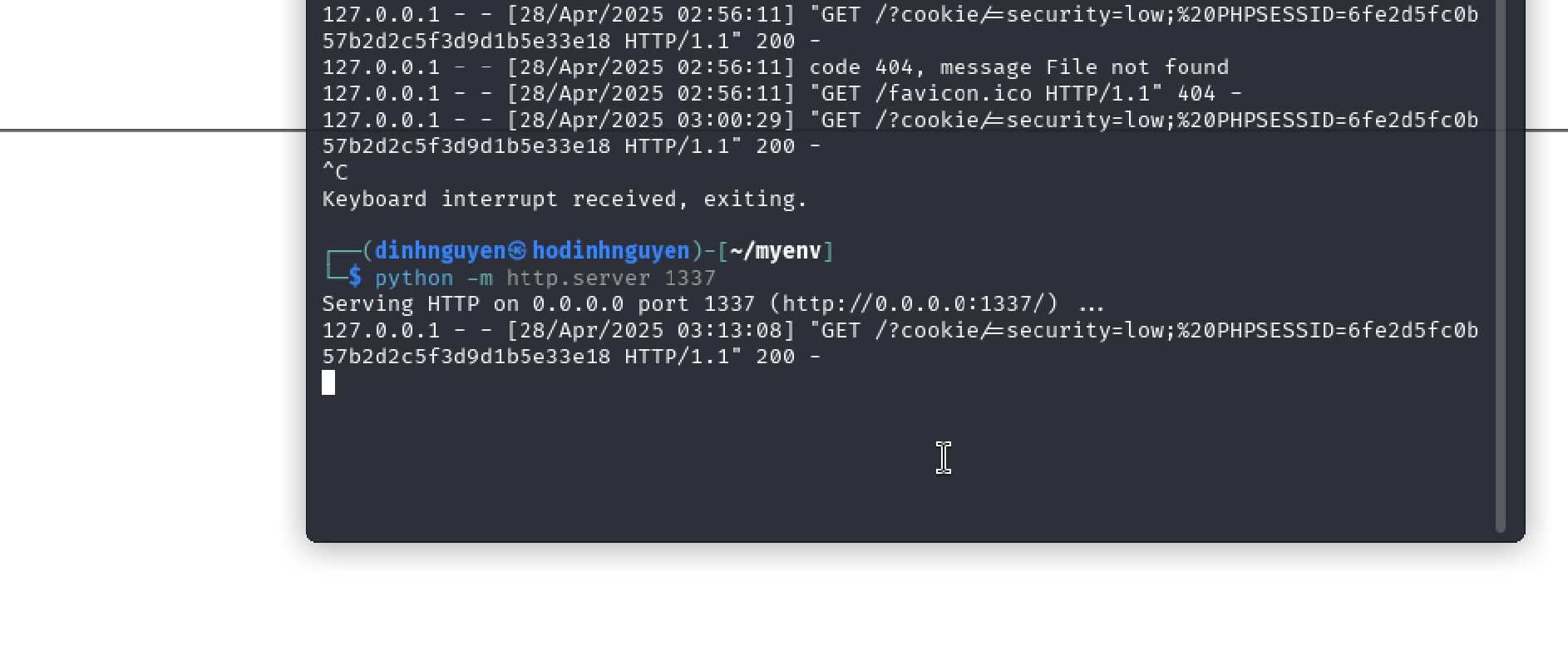


-điền scrip





hiển thị trên server



1.3. Tham khảo mã nguồn (view source), giải thích ngắn gọn cách DVWA với security ở mức độ

medium, high, impossible ngăn chặn tấn công khai thác lỗi Stored XSS.

- Medium: hàm strip\_tags loại bỏ các thẻ HTML và PHP ra khỏi chuỗi và sẽ trả về chuỗi đã loại bỏ

hết các thẻ HTML và PHP trong $message. Hàm str\_replace sẽ loại bỏ tất cả &lt;script&gt;trong $name.

- High: hàm strip\_tags loại bỏ các thẻ HTML và PHP ra khỏi chuỗi và sẽ trả về chuỗi đã loại bỏ hết

các thẻ HTML và PHP trong $message. Hàm addslashes() thêm một dấu gạch chéo ngược (\) phía

trước ký tự đặc biệt: dấu nháy kép, dấu nháy đơn và dấu gạch chéo ngược trong $message.Loại

bỏ tất cả các kí tự mà script có và không cho phép chữ T trong https.

- Impossible: Hàm htmlspecialchars () được triển khai lọc tất cả các ký tự nguy hiểm từ $message

và $name.

Dưới đây là giải pháp rút gọn cách DVWA ngăn chặn tấn công XSS được lưu trữ ở các mức bảo mật Medium , High , và Impossible :

Giải thích:

Trung bình :

Sử dụng Strip\_tags() để loại bỏ các thẻ HTML và PHP khỏi $message , trả về việc dọn dẹp chuỗi.

Sử dụng str\_replace() để xóa chuỗi <script> khỏi $name .

Chế độ : Chỉ loại bỏ thẻ <script> cụ thể, các cách viết mã độc khác (như onerror , onload ) vẫn có thể khai thác thác.

Cao :

Sử dụng Strip\_tags() để xóa thẻ HTML và PHP khỏi $message .

Sử dụng addedlashes() để thêm dấu \ trước các ký tự đặc biệt ( " , ' , \ ) trong $message , ngăn chặn thoát chuỗi.

Loại bỏ ký tự trong <script> và không cho phép chữ T trong https (ngăn chặn URL độc hại).

Chế độ : Vẫn có thể bỏ qua bằng cách sử dụng HTML sự kiện thuộc tính hoặc mã hóa khác.

Không thể nào :

Sử dụng htmlspecialchars() để chuyển đổi các ký tự nguy hiểm (như < , > , & , " , ' ) trong $message và $name thành dạng HTML mã hóa (ví dụ: < thành phố < ), ngăn chặn hoàn toàn việc thực thi mã JavaScript.

Hiệu quả : Gần như không thể khai thác XSS vì mọi ký tự nguy hiểm đều được mã hóa, đảm bảo đầu vào không thể trình duyệt hiểu là mã thực thi.

Tóm tắt : Level Impossible là an toàn nhất nhờ htmlspecialchars() , trong khi Medium và High vẫn có lỗi do chỉ lọc một phần mã độc.

Câu 2: Khai thác lỗi File upload trên DVWA

2.1 Tham khảo các tài liệu đính kèm để khai thác lỗi File upload trên DVWA (security ở

mức độ low) để chiếm shell của máy chạy DVWA.

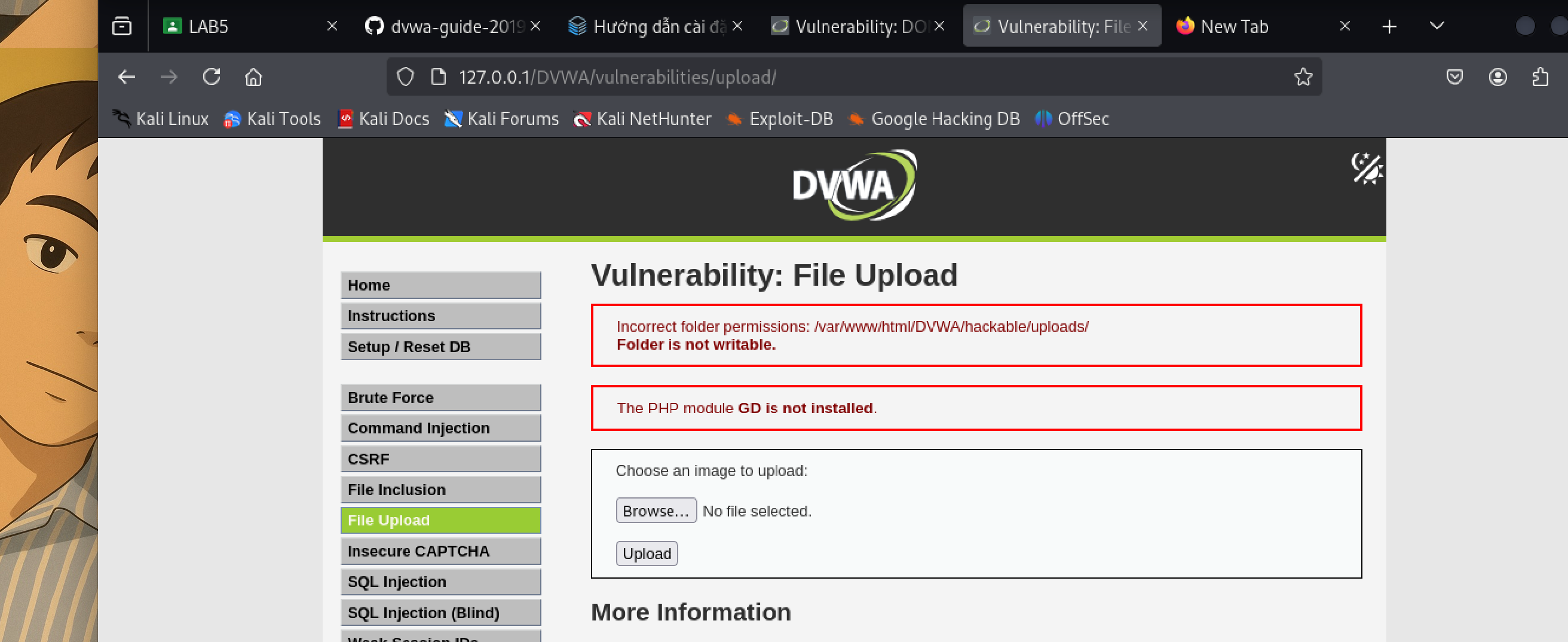
https://bughacking.com/dvwa-ultimate-guide-first-steps-and-walkthrough/#File\_Upload

https://github.com/mrudnitsky/dvwa-guide-2019/blob/master/low/Challenge%2005:%20File%20Upload.md

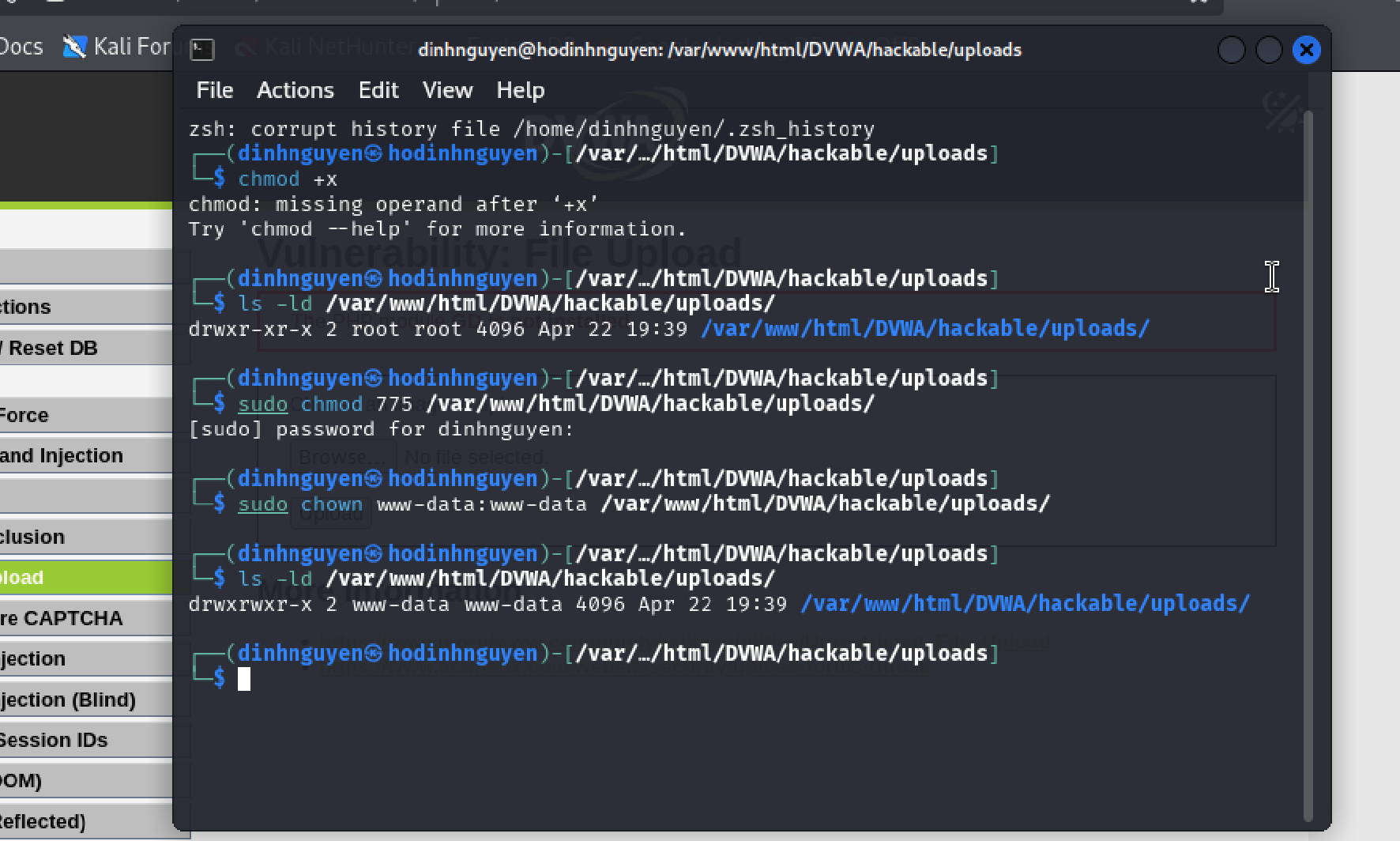
https://youtu.be/K7XBQWAZdZ4?list=PLHUKi1UlEgOJLPSFZaFKMoexpM6qhOb4Q

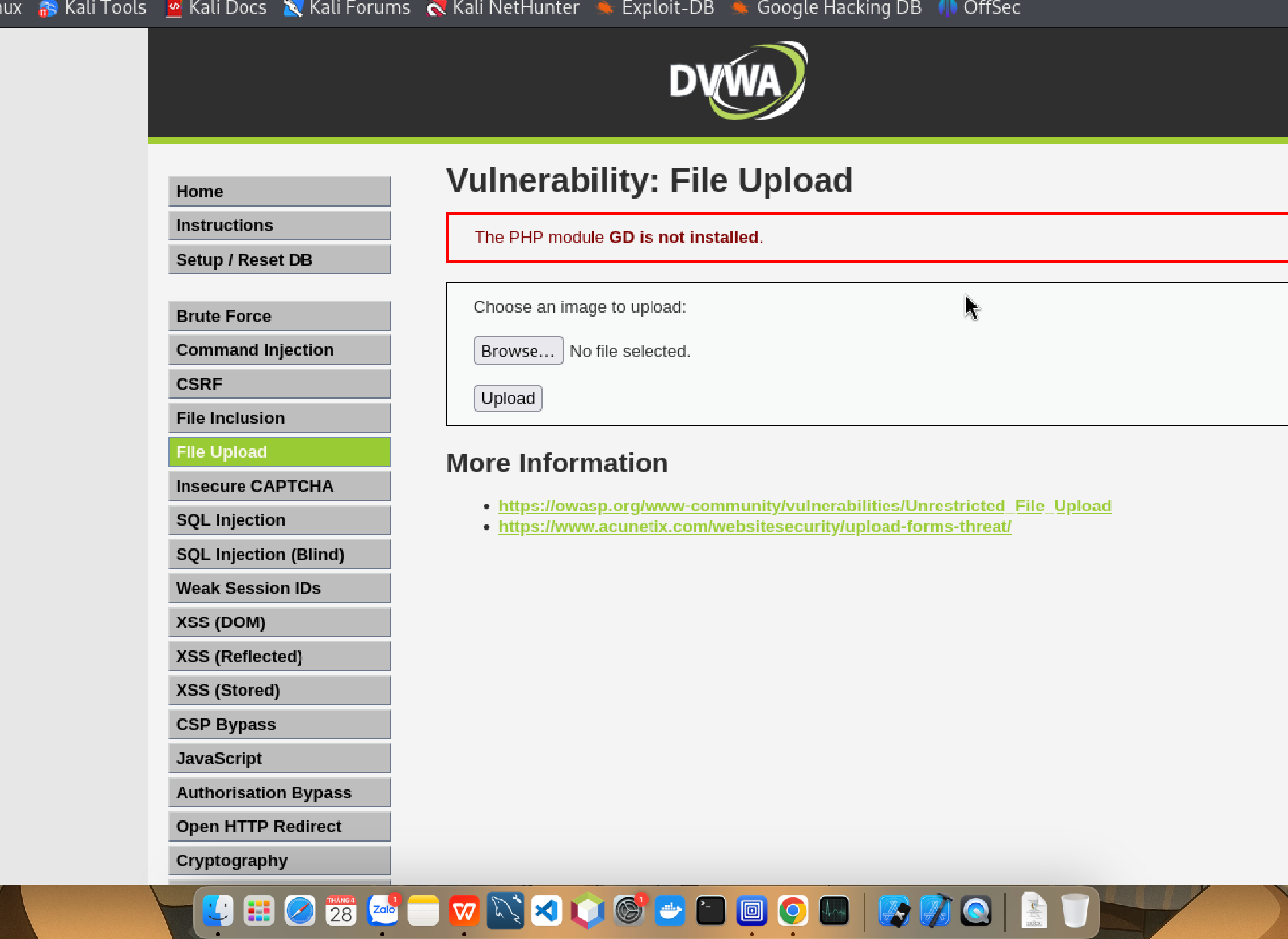
(Chụp hình minh họa quá trình thực hiện)

-vào trang

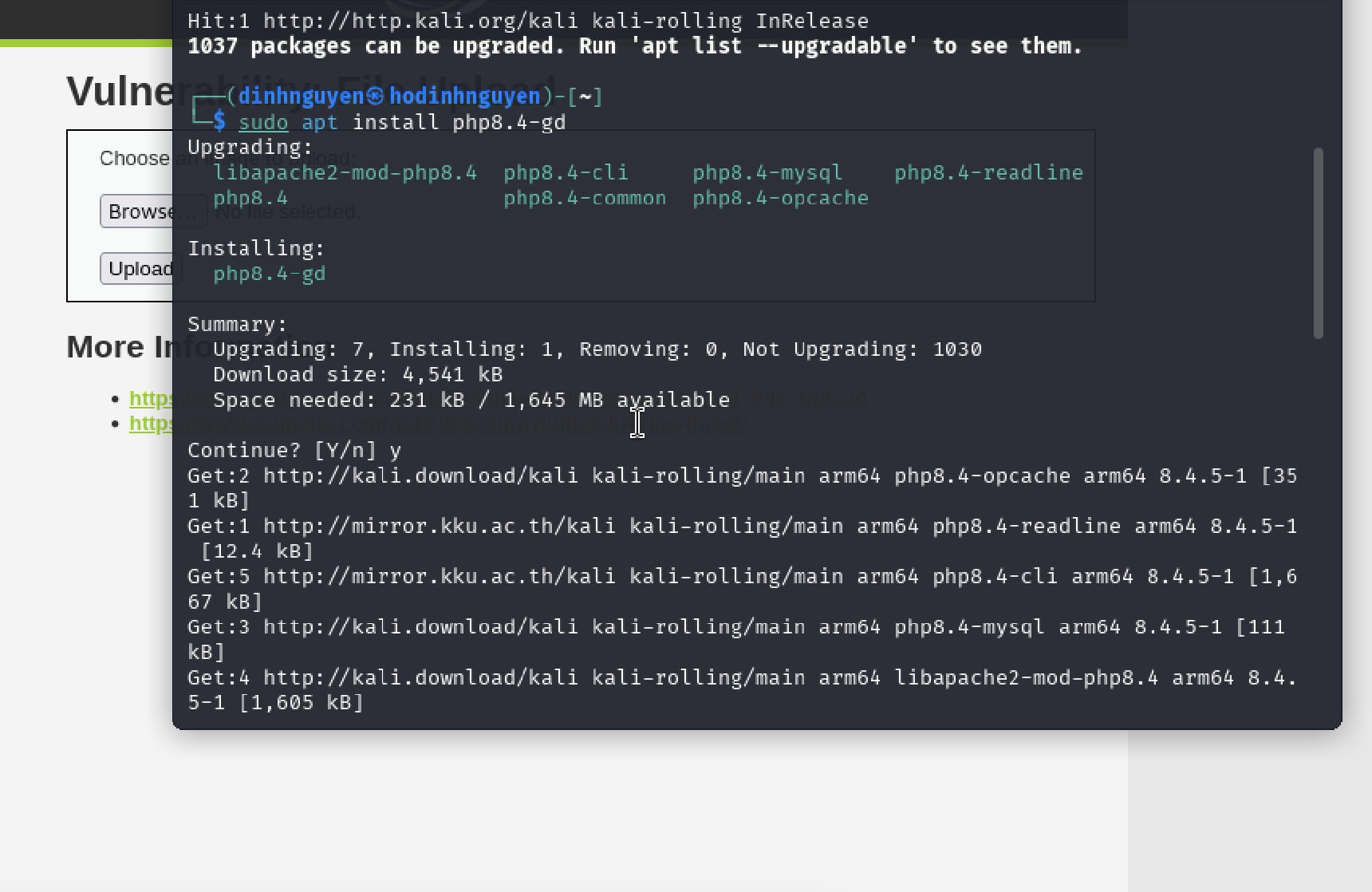


-cấp quyền ghi cho thư mục upload

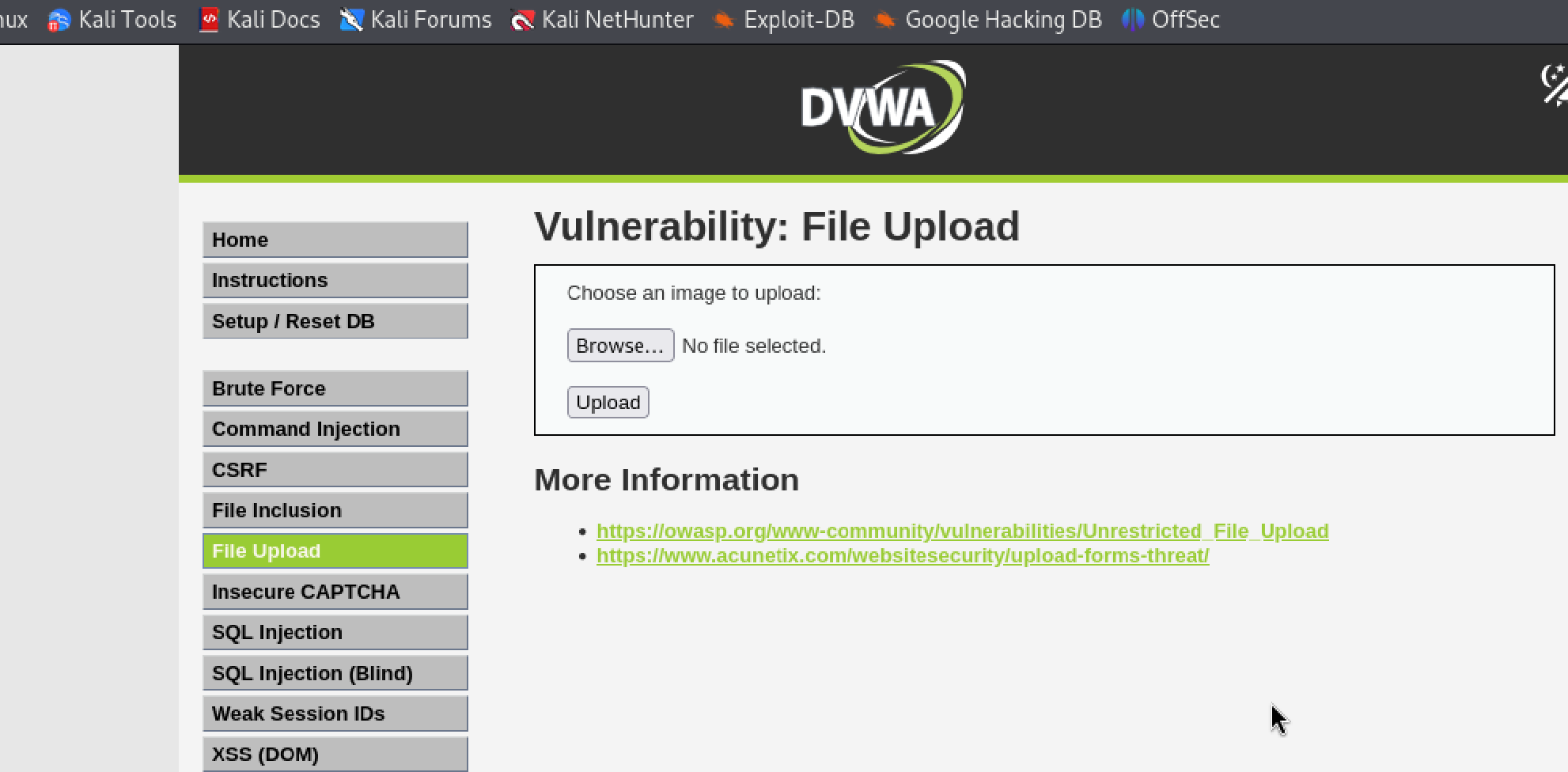




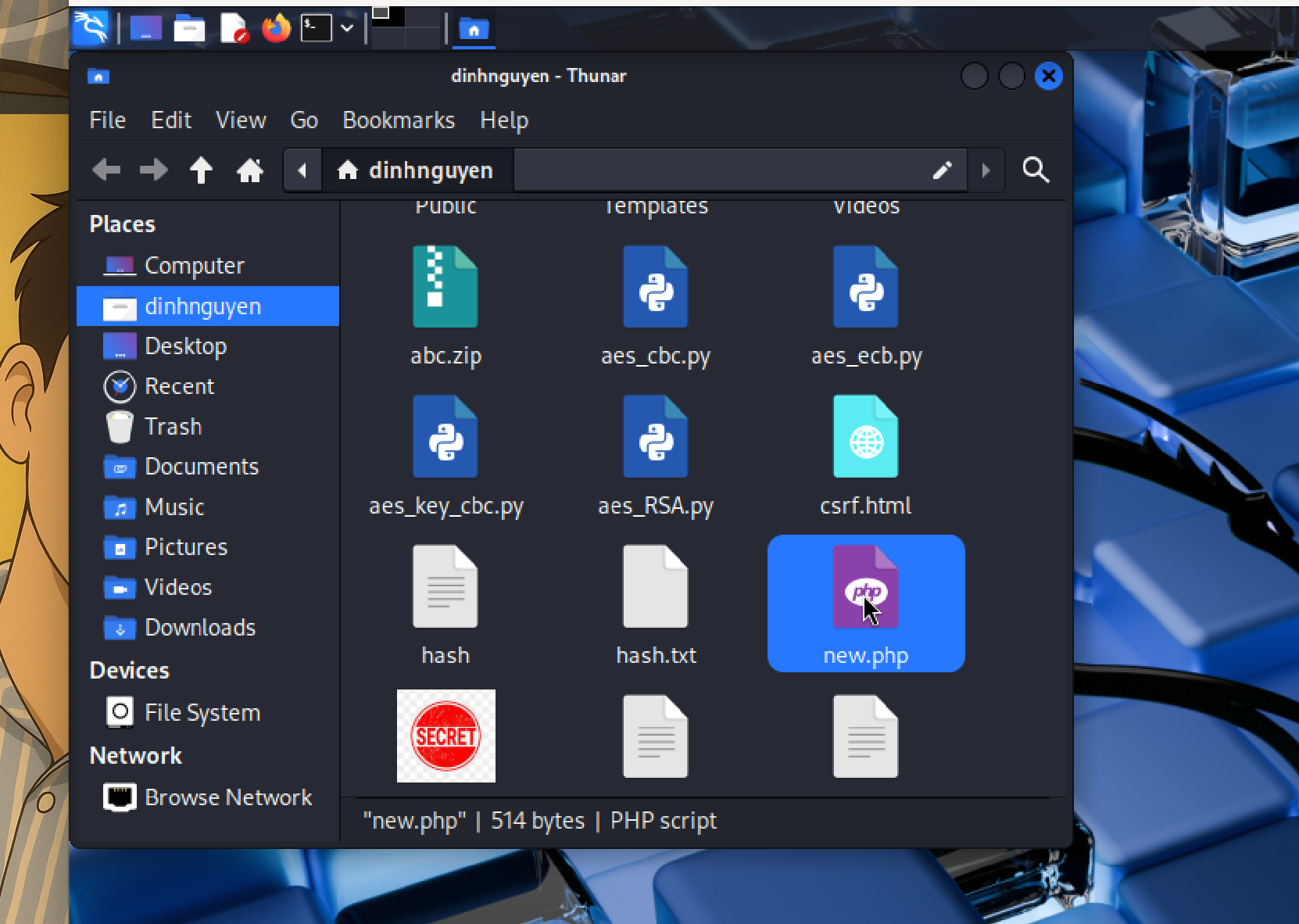
-dowload php gd



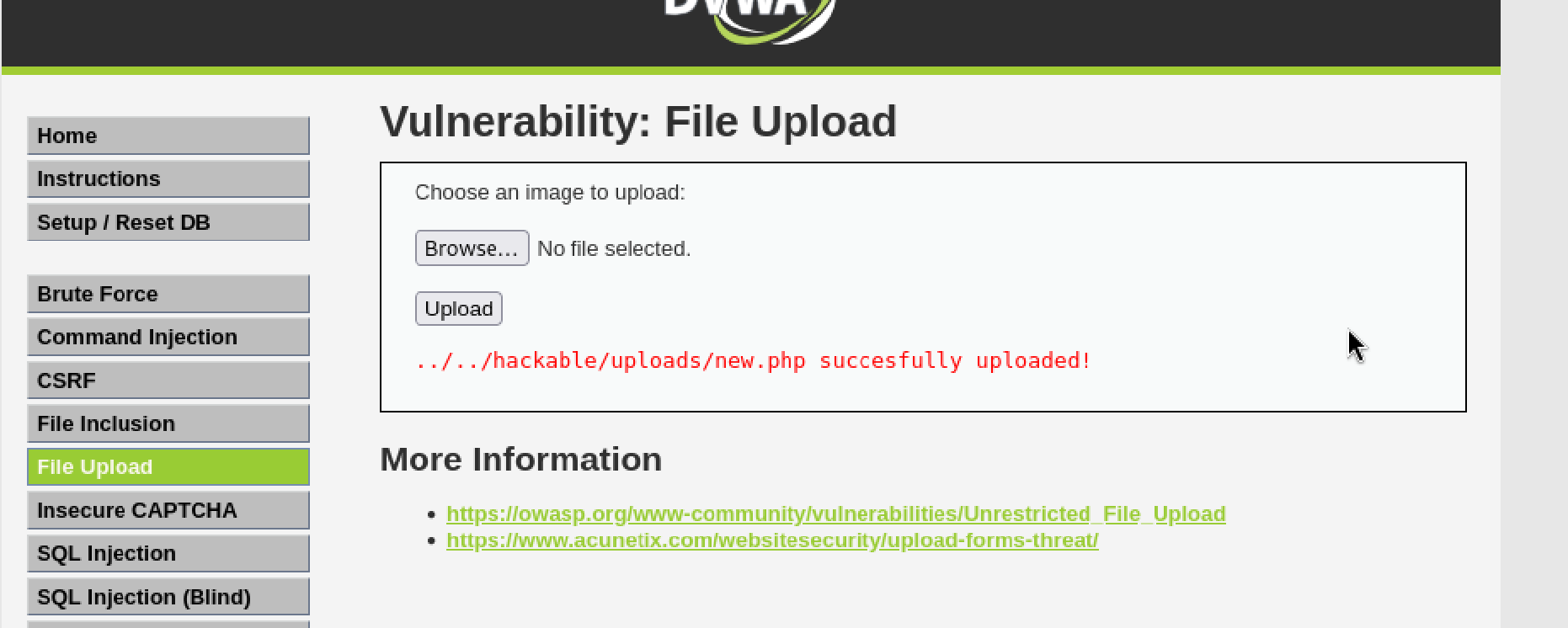
-sau khi cài đặt gd thì mở lại trang web

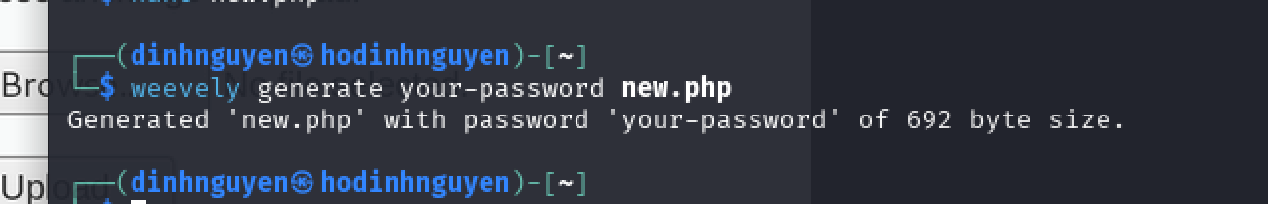


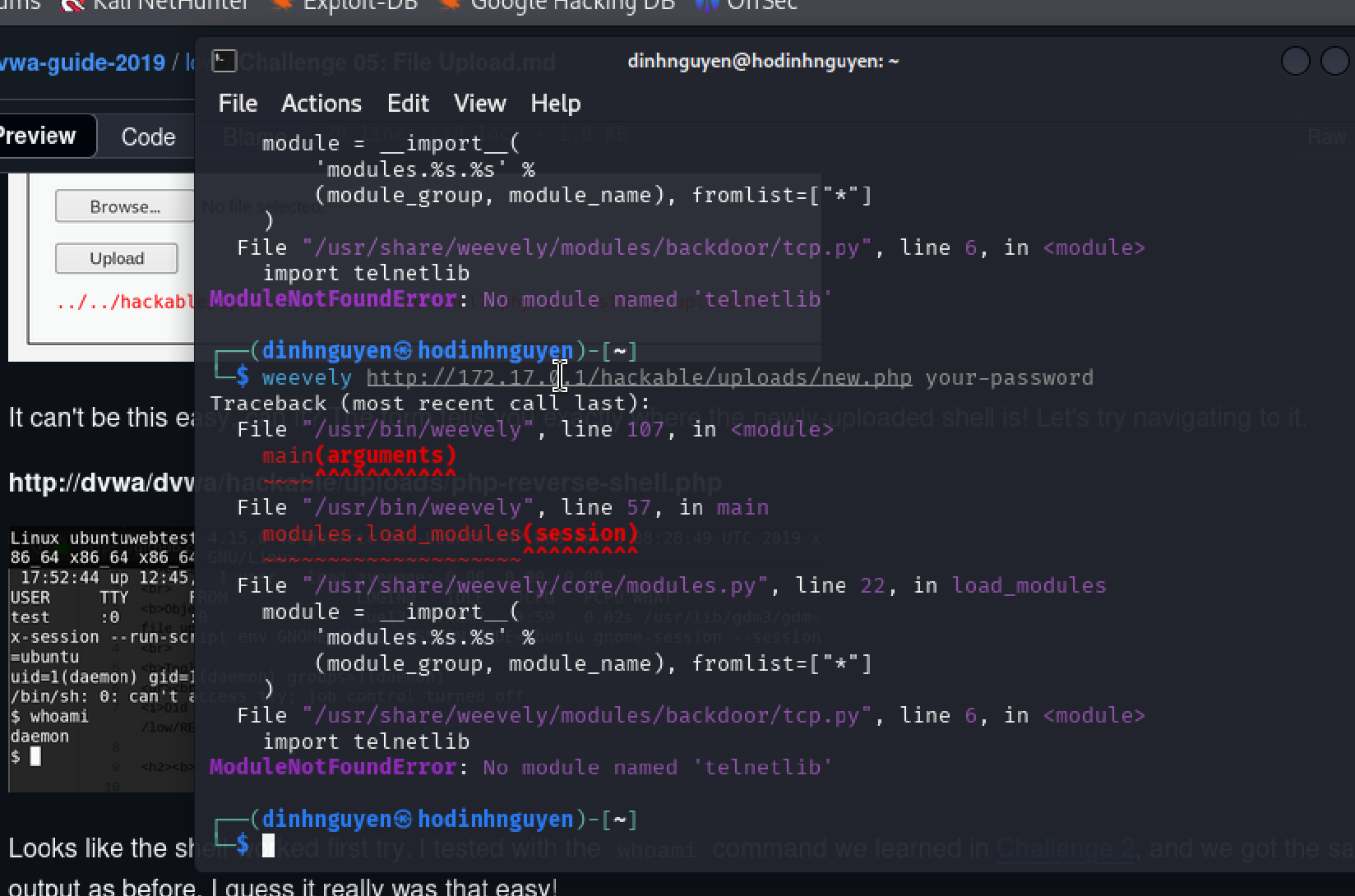
-tạo file new.php



-uoload file







2.2. Tham khảo mã nguồn (view source), giải thích ngắn gọn cách DVWA với security ở mức độ

medium, high, impossible ngăn chặn tấn công khai thác lỗi File upload.

- Medium: loại file được phép tải lên là image/jpeg bằng cách kiểm tra tên tệp, kích thước tệp, kiểu

tệp và lấy chuỗi sau dấu chấm “.” để lấy được loại file trong cú pháp thẻ img. Ví dụ: &lt;img src=&quot;img\_

girl.jpg&quot; alt=&quot;Girl in a jacket&quot; width=&quot;500&quot; height=&quot;600&quot;&gt; sẽ lấy được jpg.

- High: kiểm tra có phải là ảnh không nếu không thì không tải lên bằng cách phân tích các thông

tin file, cho phép jpg, jpge, png file.

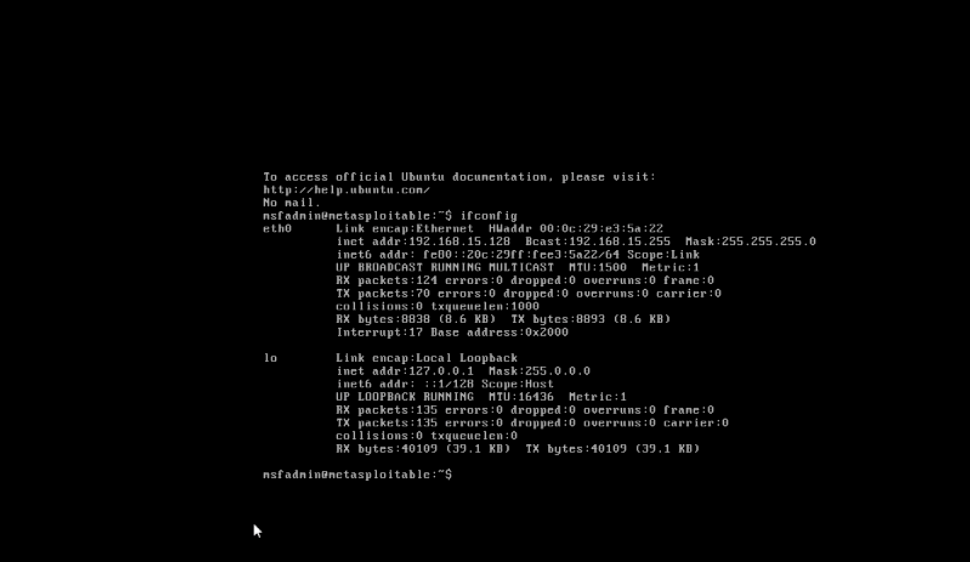
- Impossible: có nhiều biện pháp bảo mật làm cho việc tải file không phải ảnh là không thể.

Câu 4: Sử dụng công cụ OWASP ZAP để tìm lỗ hổng bảo mật của ứng dụng web

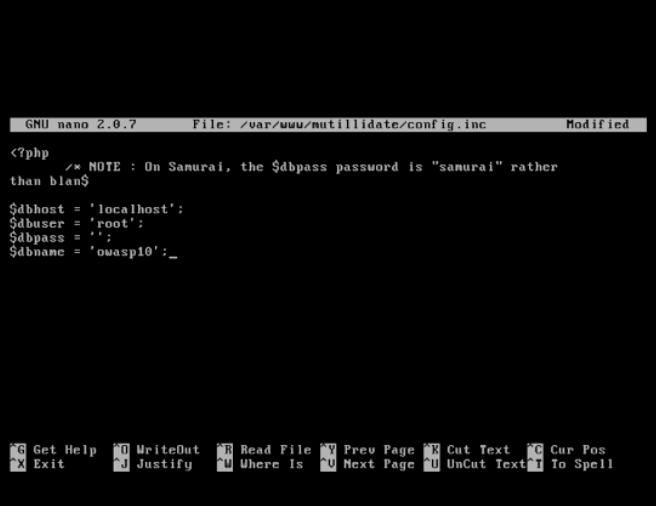
4.1. Sửa lỗi cấu hình cơ sở dữ liệu của Mutillidae trên máy ảo Metasploitable 2

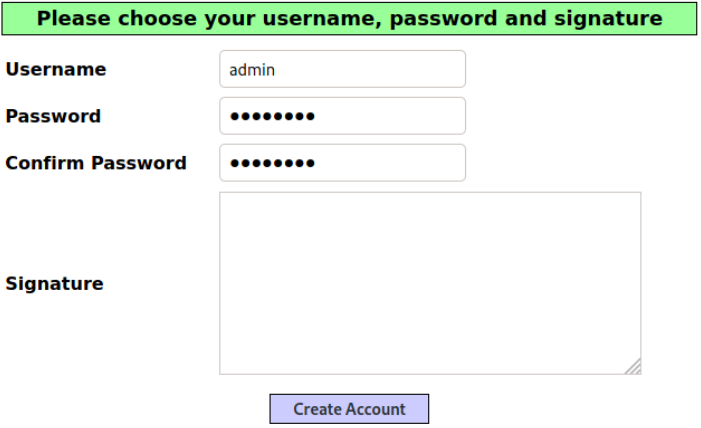
<https://nudesystems.com/fix-mutillidae-database-error-in-metasploitable-2/>

khởi động máy ảo



cấu hình





IMG_256



4.2. Khởi động máy ảo Metasploitable 2. Sau đó dùng công cụ OWASP ZAP để tìm các lỗ hổng bảo mật có trên ứng dụng web Mutillidae trên máy ảo Metasploitable 2.

(Chụp hình minh họa quá trình thực hiện)