|  |  |
| --- | --- |
| Class | IAW301 |
| Name | Trần Ngọc Anh |
| IC No. | HE151461 |
| Date & Time | 10/6/2022 |

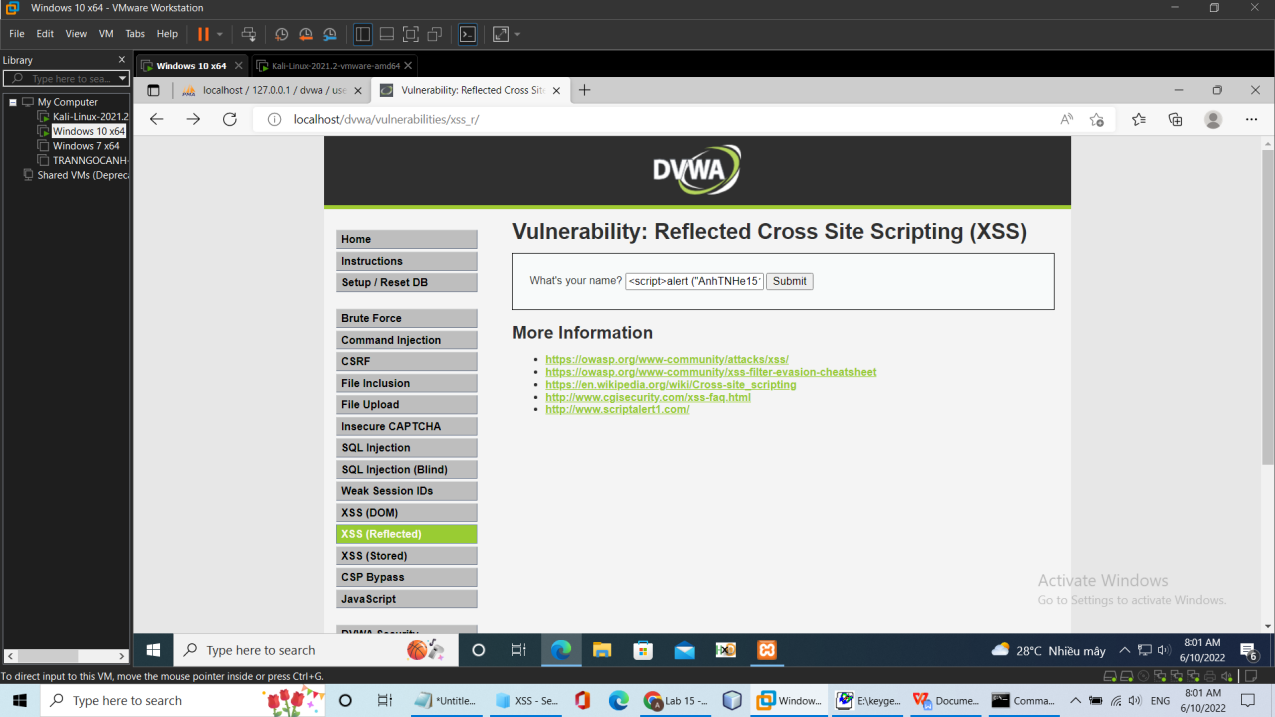
# LAB 15 - Reflected Cross Site Scripting

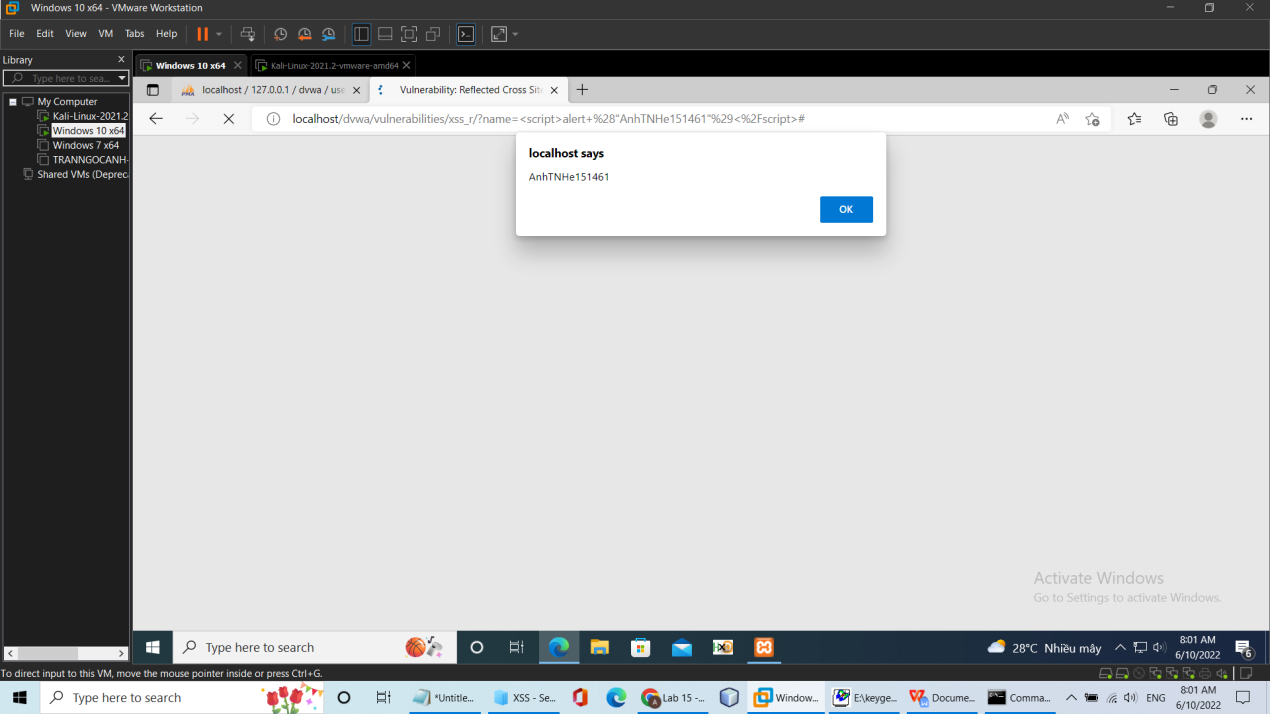
1. **Low:**

- Ở mức low nhìn vào sourcode chúng ta thấy chỉ cần người dùng đã nhập giá trị cho trường thông tin name là nó sẽ được in ra.

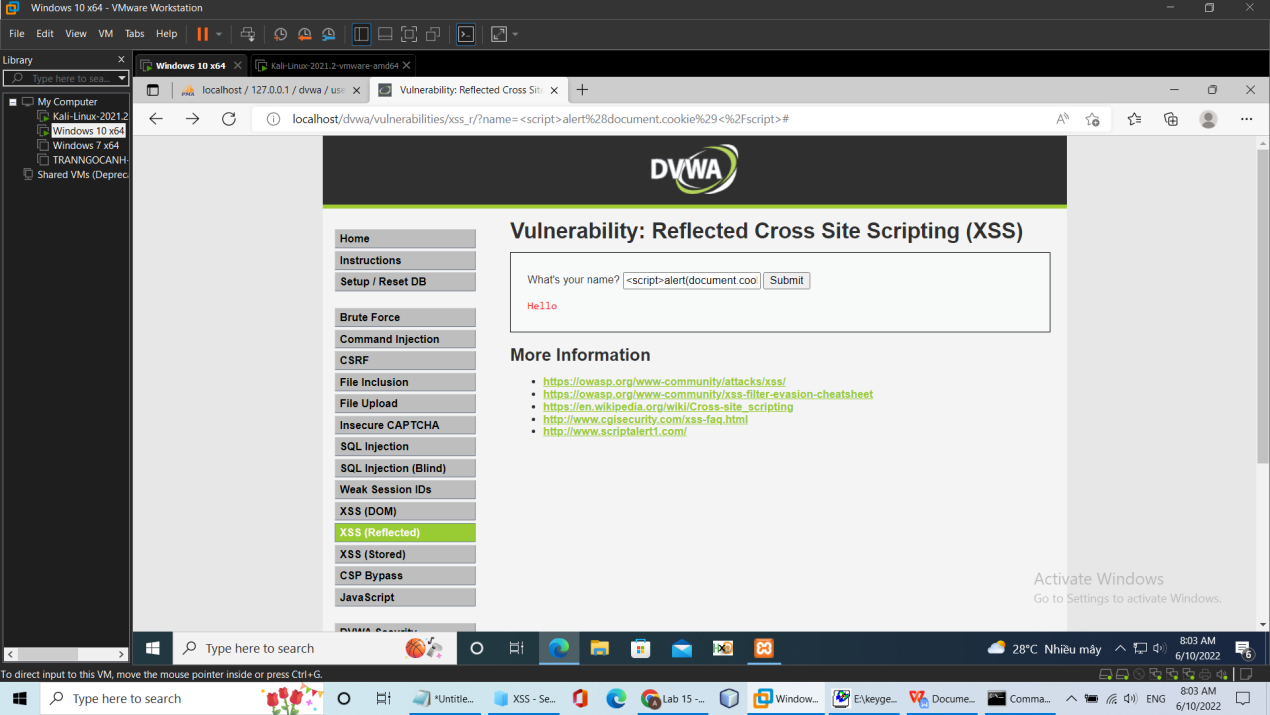


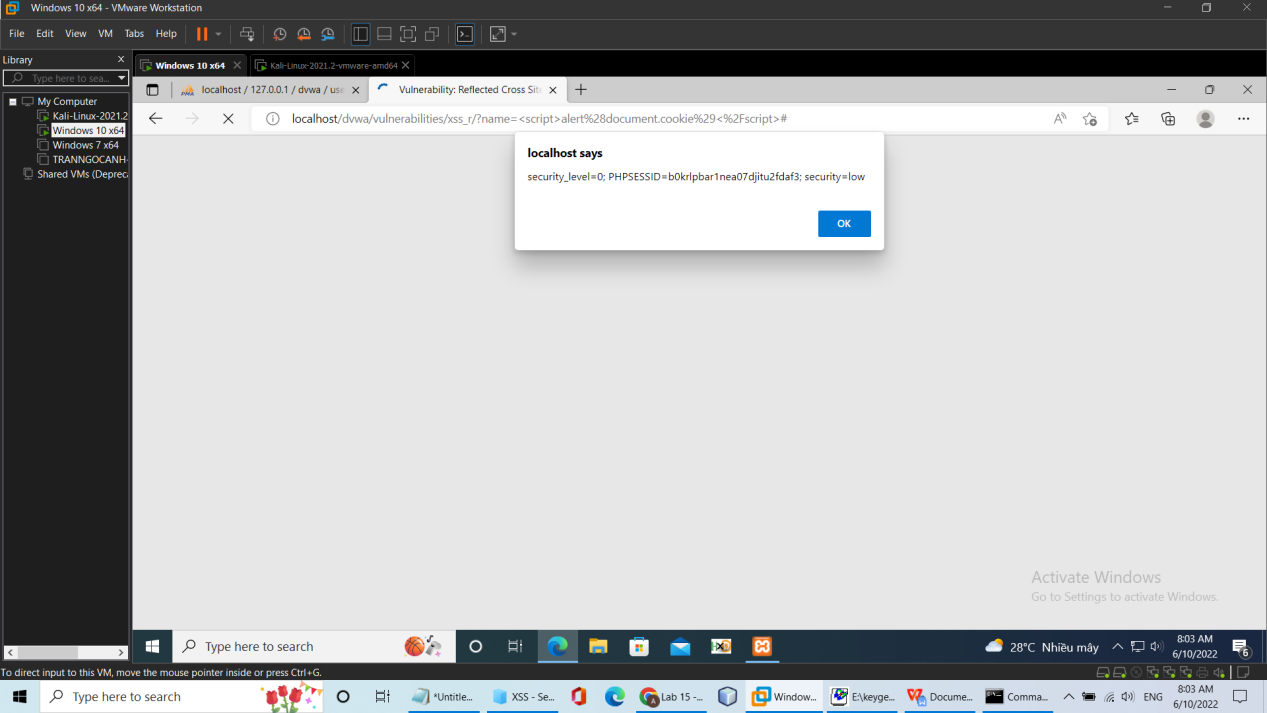
- Chúng ta thử inject một đoạn script in ra mã số sinh viên của mình để chứng minh có lỗ hổng Reflected XSS và đã thành công. <script>alert("AnhTNHe151461")</script>





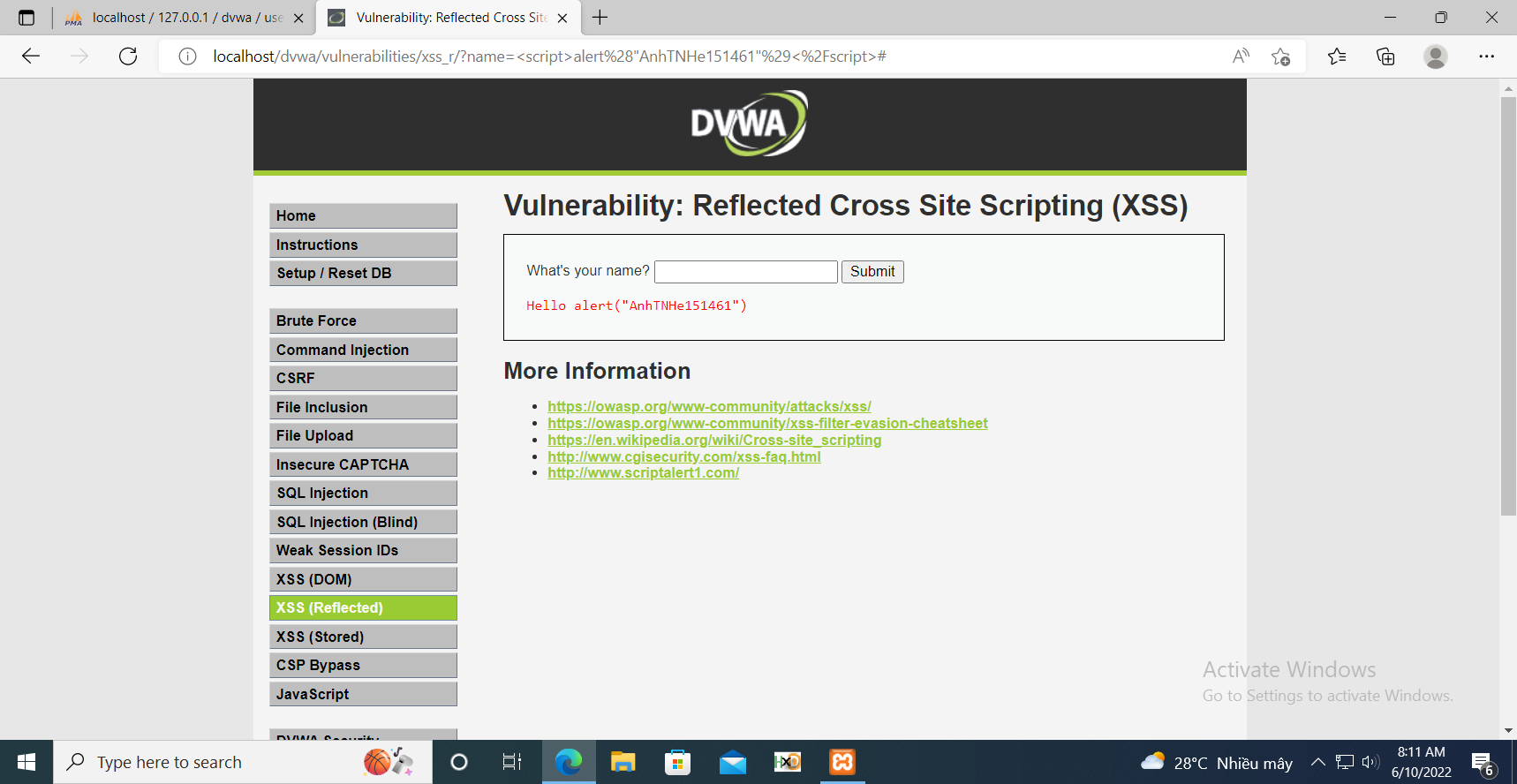
- Sau đó chúng ta sẽ inject đoạn code ( <script>alert(document.cookie);</script> ) để lấy cookie và hoàn thành mức low.



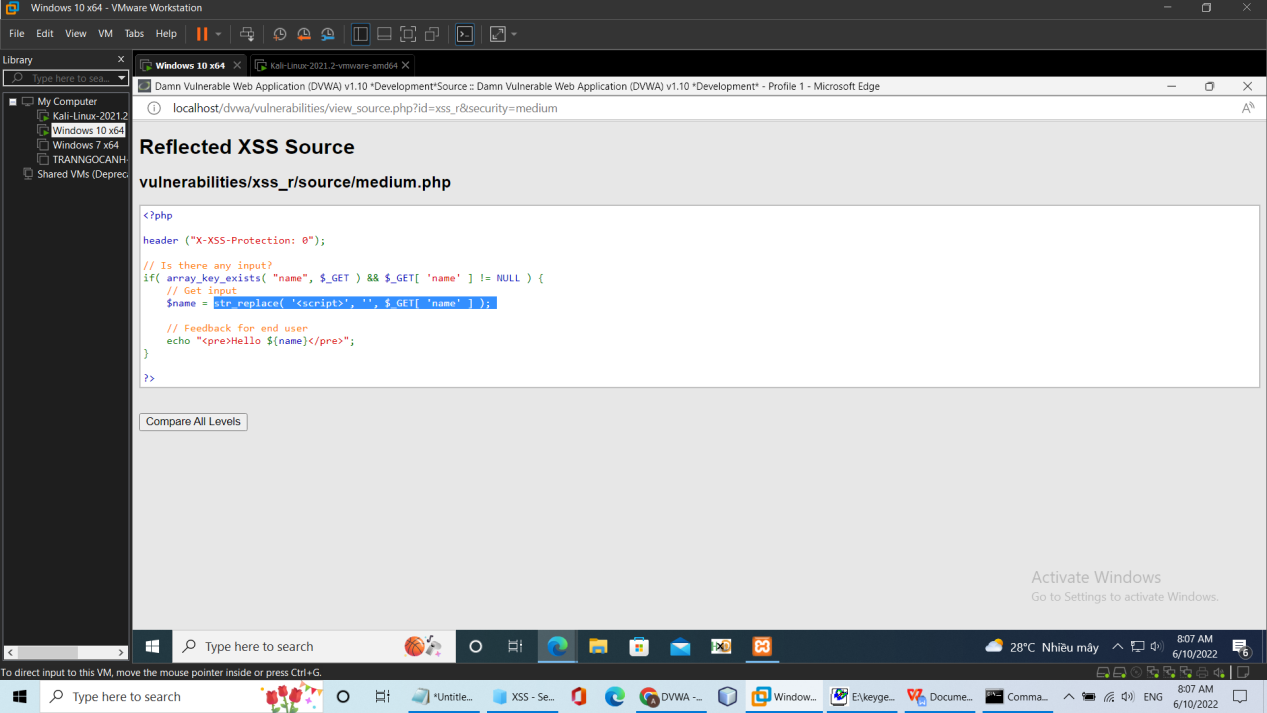


1. **Medium**

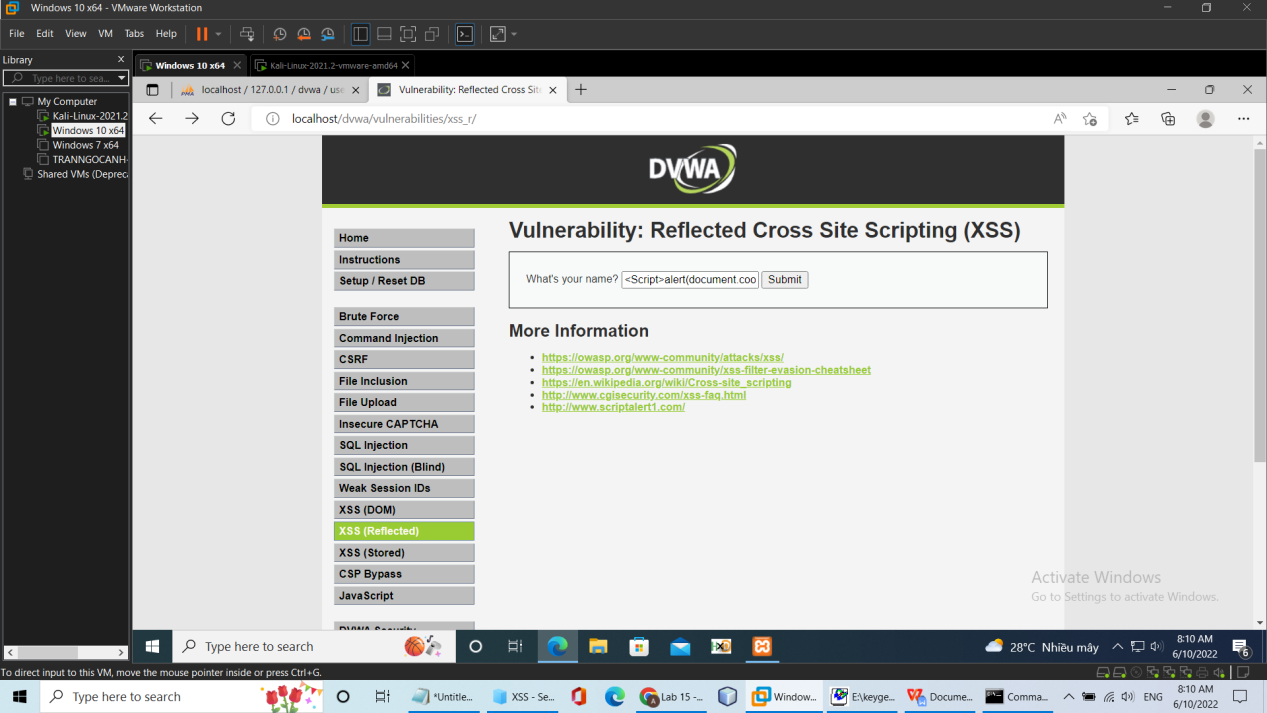
- Sang mức medium đầu tiên chúng ta thử như mức low để chứng minh có lỗi nhưng không thành công.( <script>alert("AnhTNHe151461")</script> )

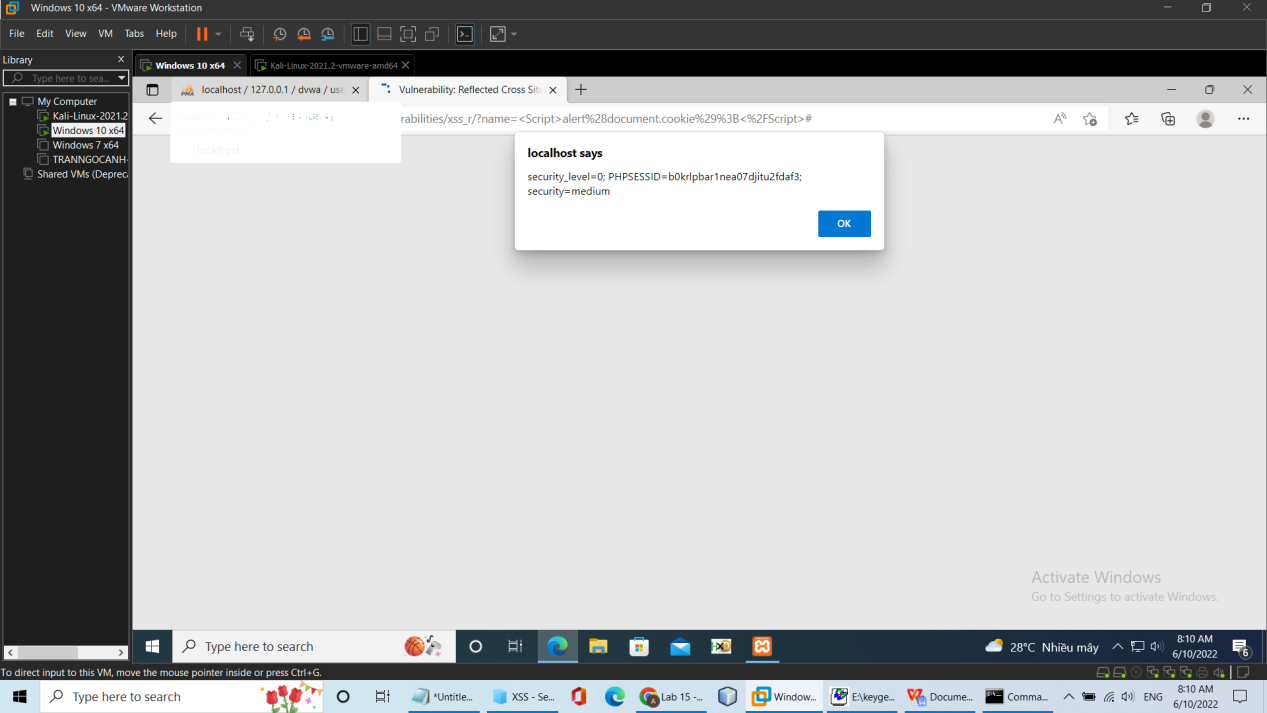


- Chúng ta thử phân tích chút về source code xem nguyên nhân ở đâu . Ở đây chúng ta nhìn thấy sau khi người dùng nhập giá trị name rồi thì sẽ được đưa vào một hàm str\_replace() để thay thế thẻ <script> thành kí tự rỗng để đoạn code ta inject không hoạt động.



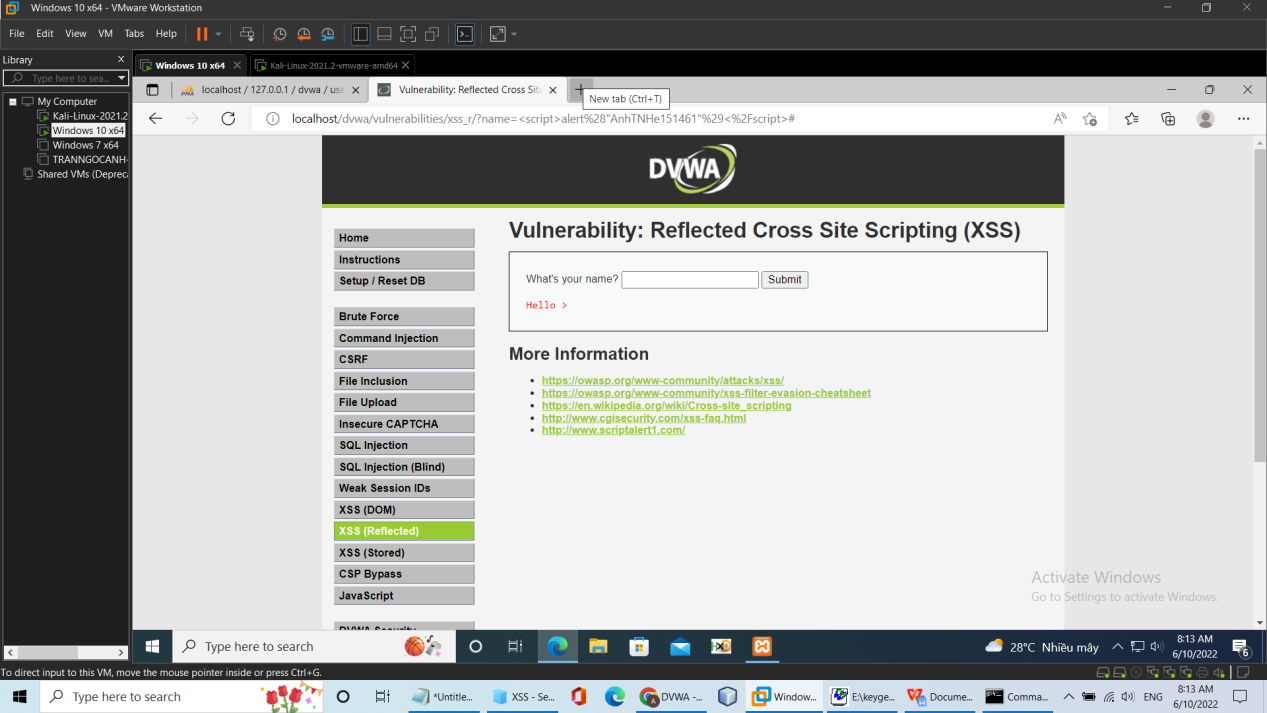
- Nhưng validate như vậy là chưa đủ và ta inject theo cách khác đó là <Script>alert(document.cookie);</Script> viết hoa chữ cái đầu của thẻ <script> để bypass hàm thay thế .



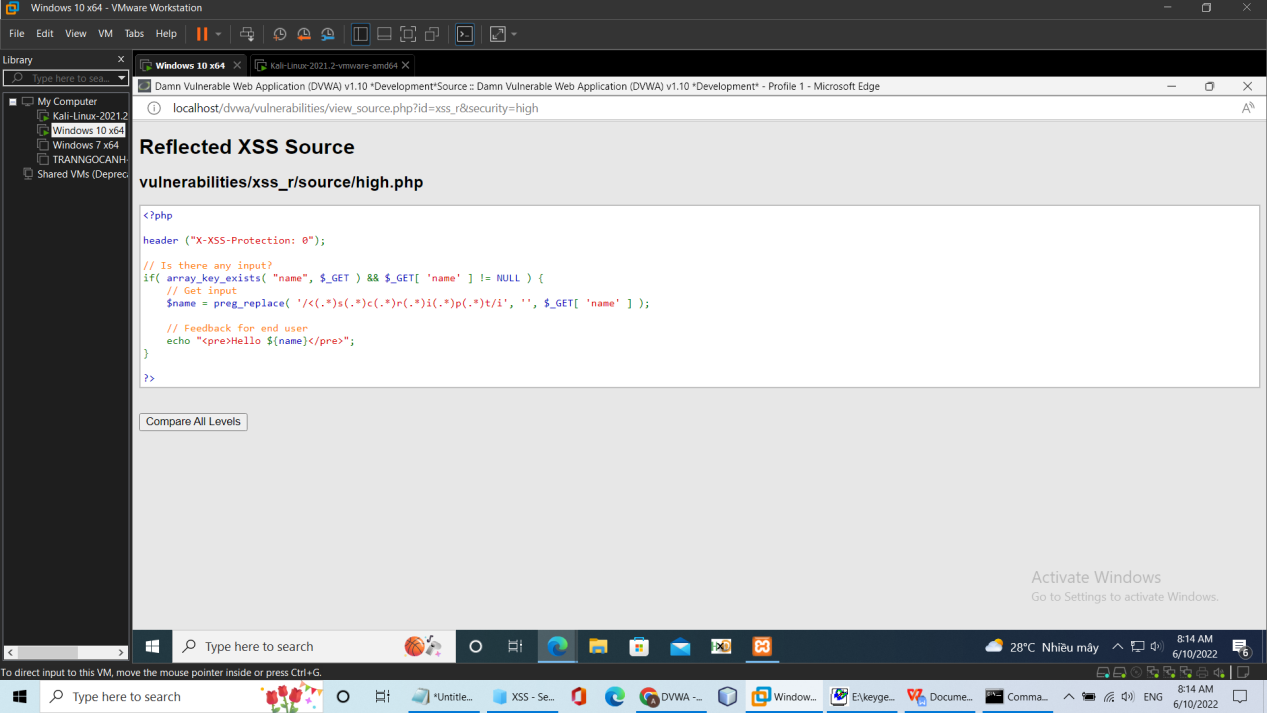


1. **High:**

- Ở mức high thử inject đoạn code như mức low, medium thì đều ra output như ảnh bên dưới.



- Đọc qua source code chúng ta thấy ở đây đã validate dữ liệu name đầu vào tốt hơn medium bằng cách loại bỏ tất cả các kiểu của thẻ <script>.



- Vì việc chèn Reflected XSS có thể được thực hiện thông qua vô số các thẻ khác như img, svg, input, iframe,… nên chúng ta có thể thử inject một đoạn code với thẻ svg ( <svg/onload=alert(document.cookie)> )

