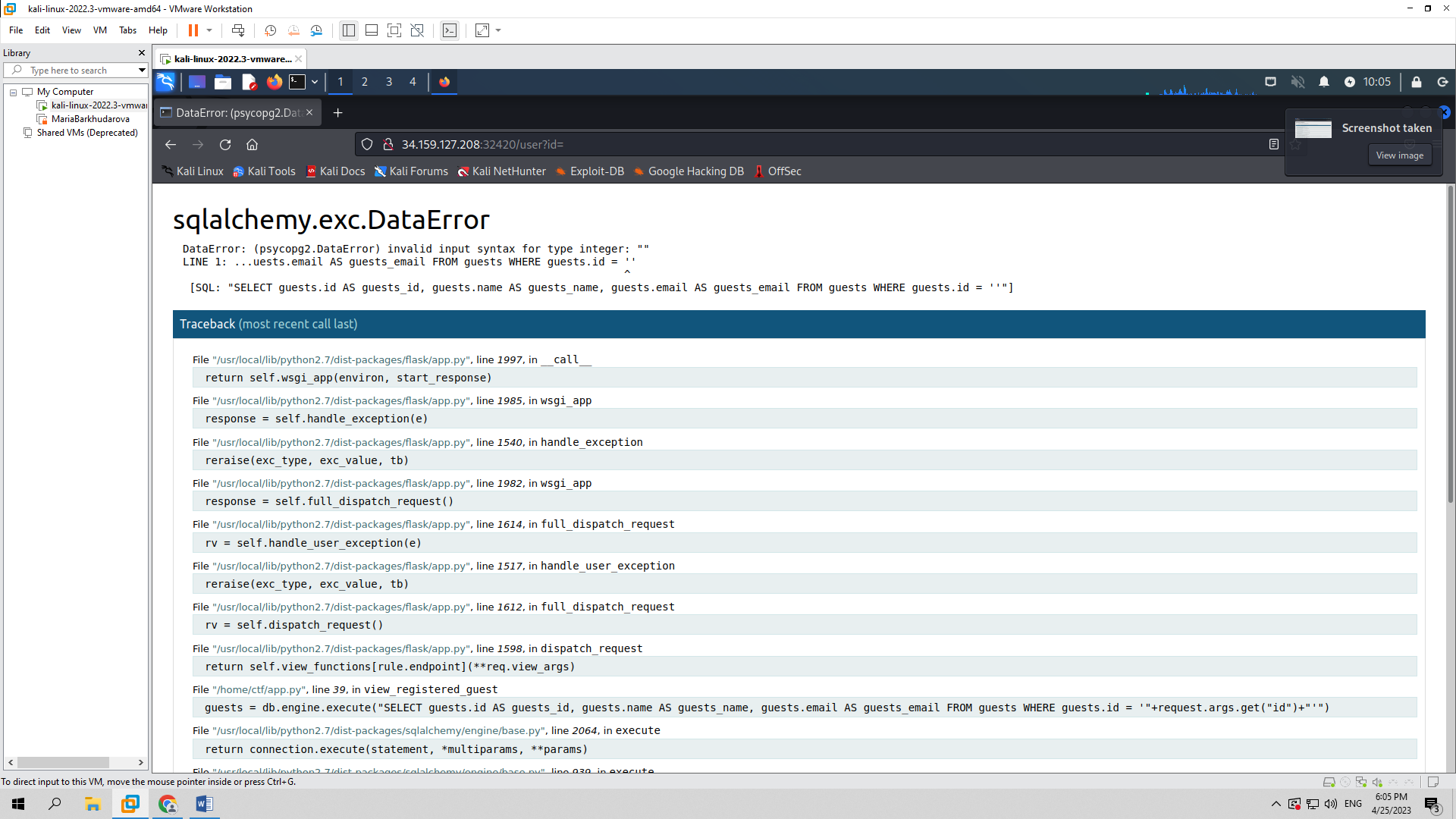
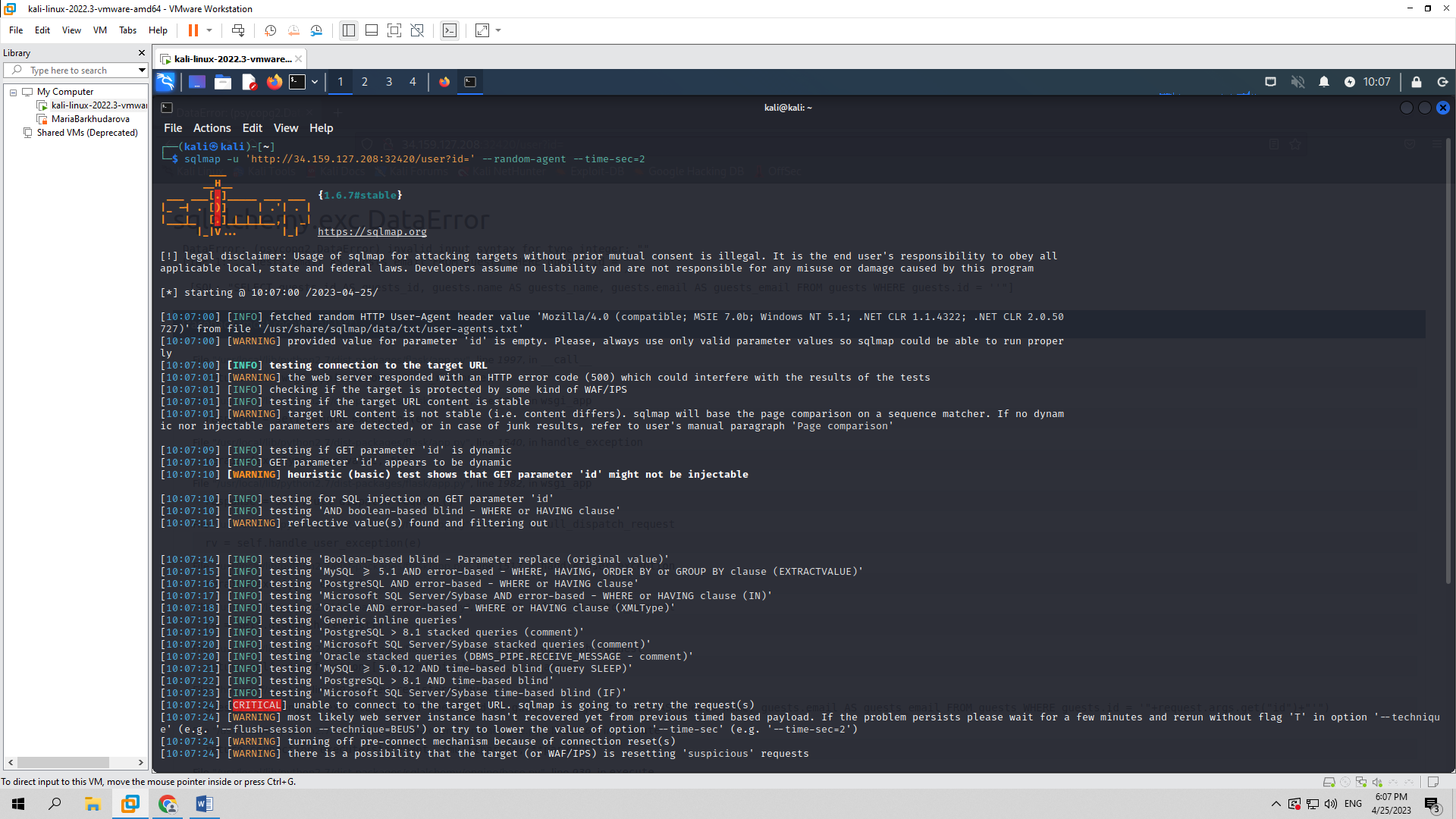
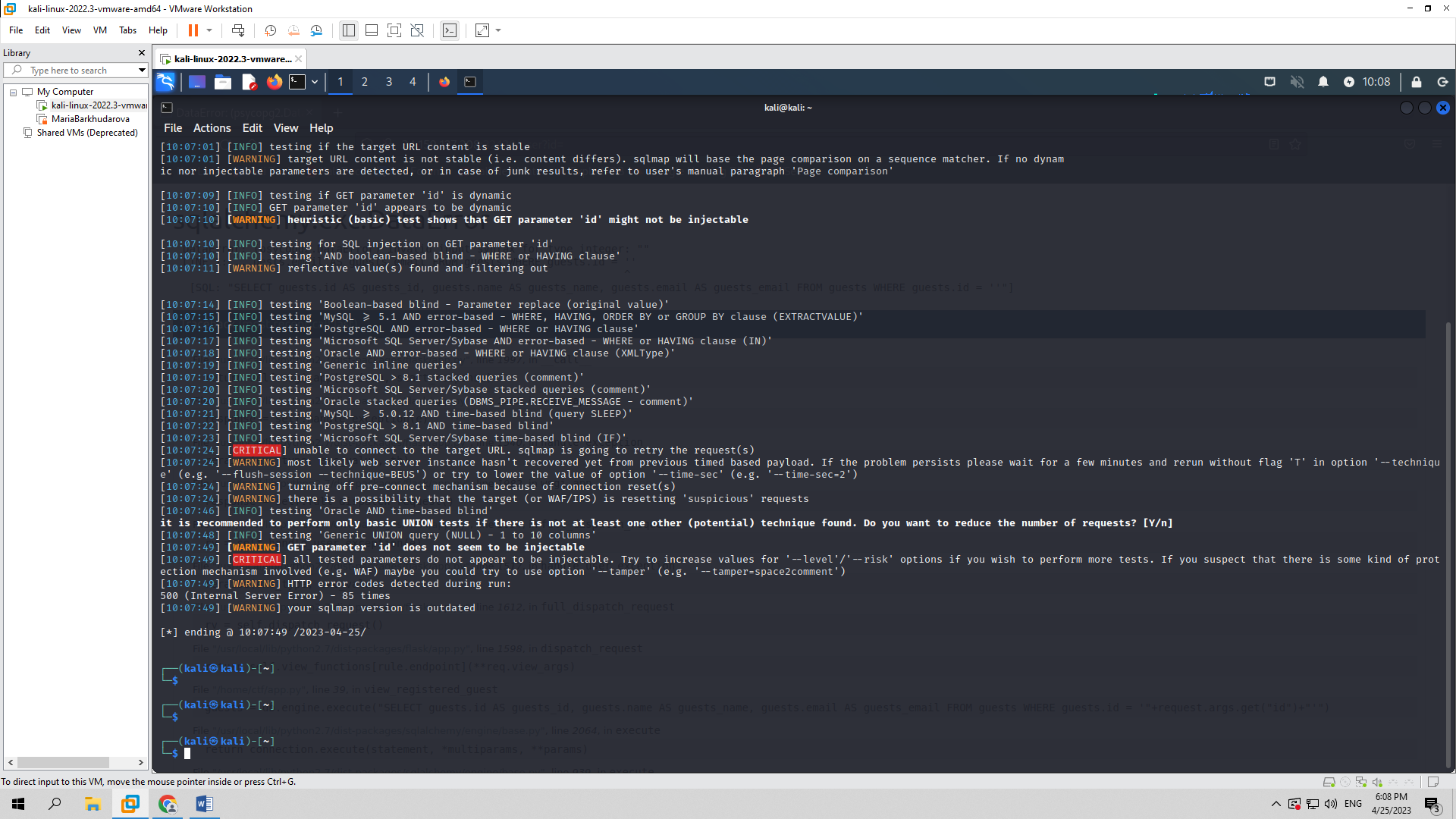
1. **Small data leak**

When accessing the website, we have two input fields, where one of them is not validated. It creates a reasonable doubt that application is vulnerable to sql injection attack. I will try to write in the url /user?id= which gives us an error. From the screenshot below we find out that sqlachemy is python based used to communicate with sql databases.



From the gathered information I will try simple sqlmap command, specifically at the url given above to inspect that point and to see If I can gain access and information about this application and as seen in the screenshots below, my suspicion is right, this website is vulnerable.





After this website was down, it was not working, probably because of the traffic, therefore I couldn’t access the website. I will just write what process I would have gone through without the screenshots, since I couldn’t access the application.

Again using sqlmap command I would try to find out about existing databases, this command would be:

sqlmap – u ‘<http://34.159.127.208:32694/>user?id=’ --random-agent --time-sec=2 –dbs

This command would give me all the databases and then I would try to read all of the tables in these databases by writing the command:

sqlmap – u ‘http://34.159.127.208:32694/user?id=’ --random-agent --time-sec=2 –D public --tables

This command would give us all the tables, if one of the table had suspicious name, maybe some part of flag, I would inspect that table’s columns and flag or last part of flag most probably would be there, so the last command I would have run would be:

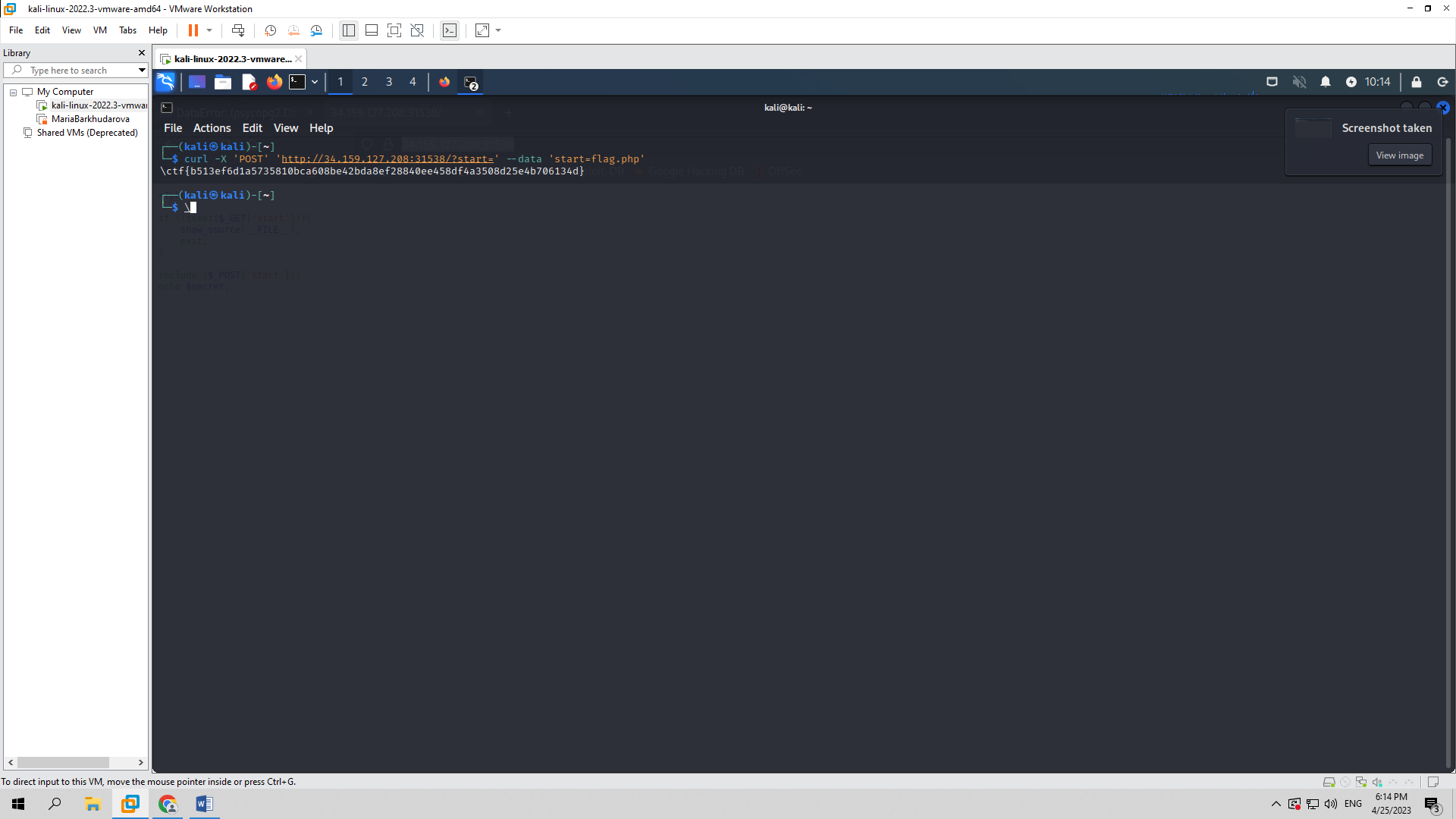
sqlmap – u ‘http://34.159.127.208:32694/user?id=’ --random-agent --time-sec=2 –D public –T ctf{23143 –columns

This would have given us the flag.

1. **Alien-inclusion**

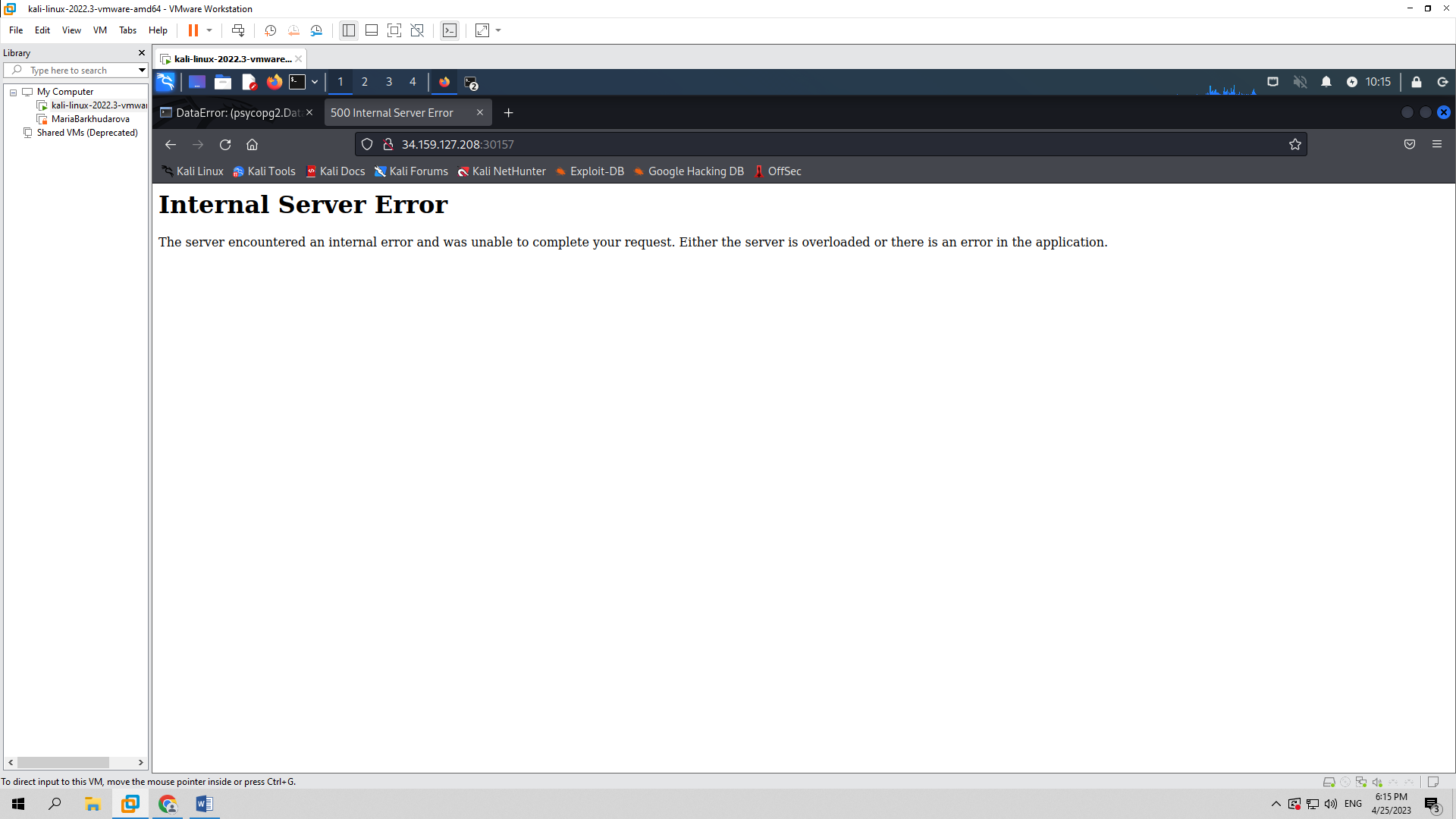
When we access the website, we are given a php code. Reading that php code I will write a curl command that will possibly give us the flag.

As it is seen in the screenshot below we got the flag. In this command –X specifies that request should be POST and –data attribute gives us information from a request. We pass flag.php to a start parameter, since it is where the flag is located and we guessed it.

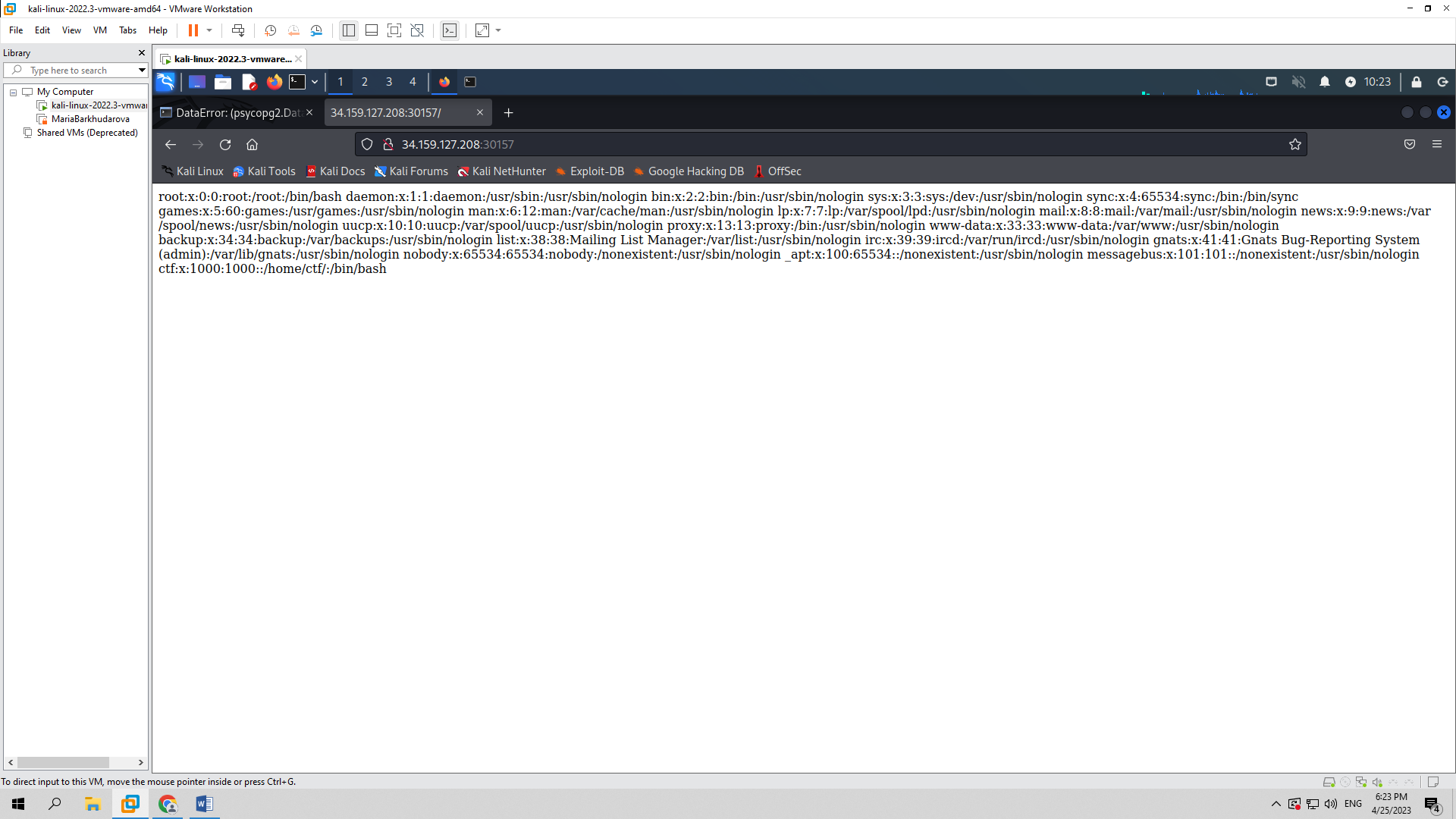


1. **Ultra-Crawler**

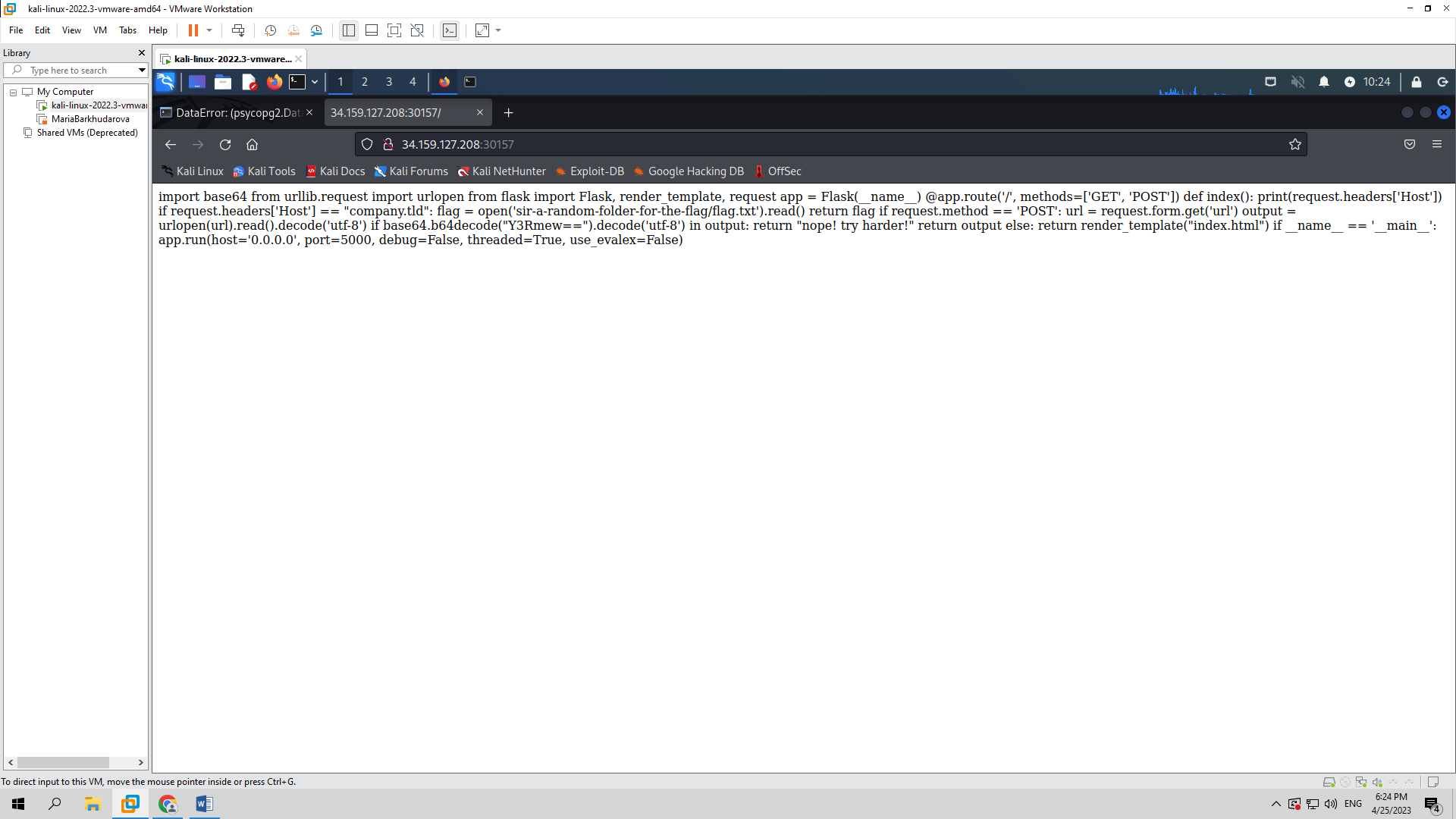
When accessing the website we see an input, where we should type in some kind of url. After typing youtube.com it gives us an error, as seen in the screenshot below. This tells us that either server or the application is not correctly configured. I have a justified suspicion that a possible LFI attack could be done here.



Therefore, I will proceed to write in the input File:/etc/passwd, since it is a folder that nearly every application or website has and as seen in the screenshot below, so does this one. From this we find out that there exists home/ctf folder, where most probably our flag is located.



After inspecting with dev tools in the response header we see that application is based on python, thus, wild guess would be that file name is app.py. From the information I’ve gathered, I will type in input File:/home/ctf/app.py, which will return the result seen in the screenshot below.



In the above code we see that if Host is set to value company.tld, flag will open and we will get the flag. That’s what I will try with curl command and some attributes. As seen in the screenshot below, I was successful and got the flag.

