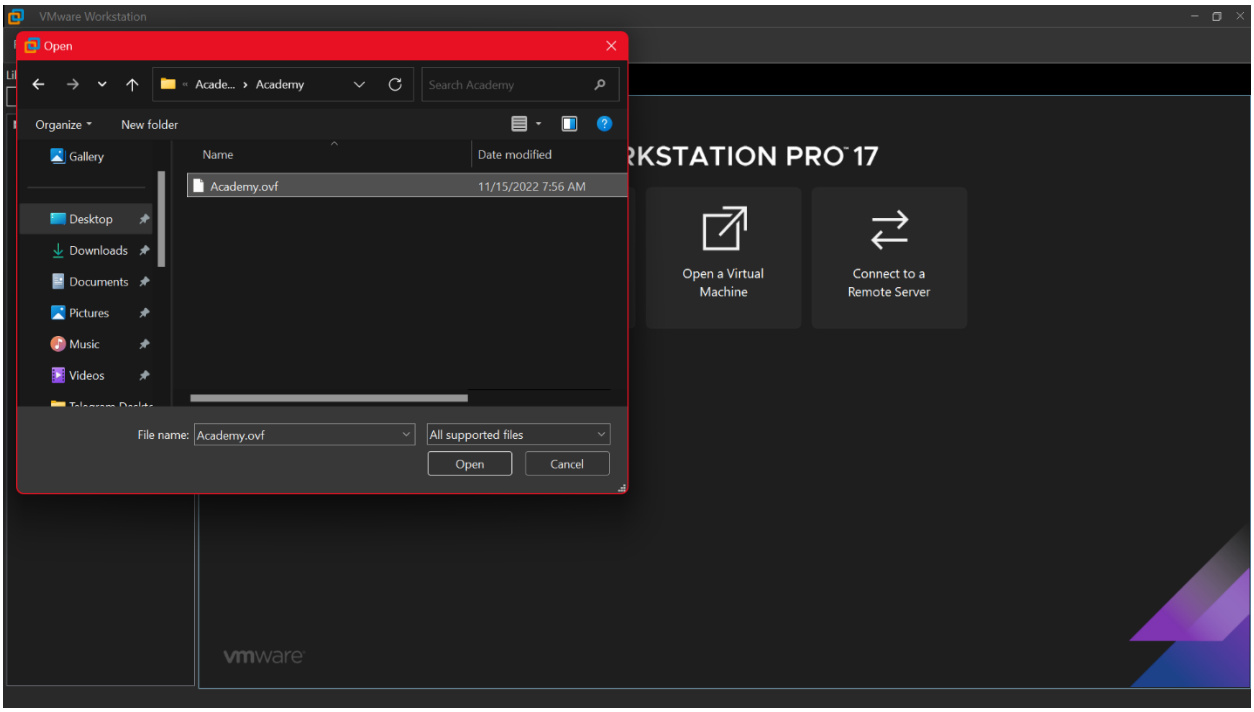
Capstone Report: Hands-On Exploitation of Academy VM – Lab Report

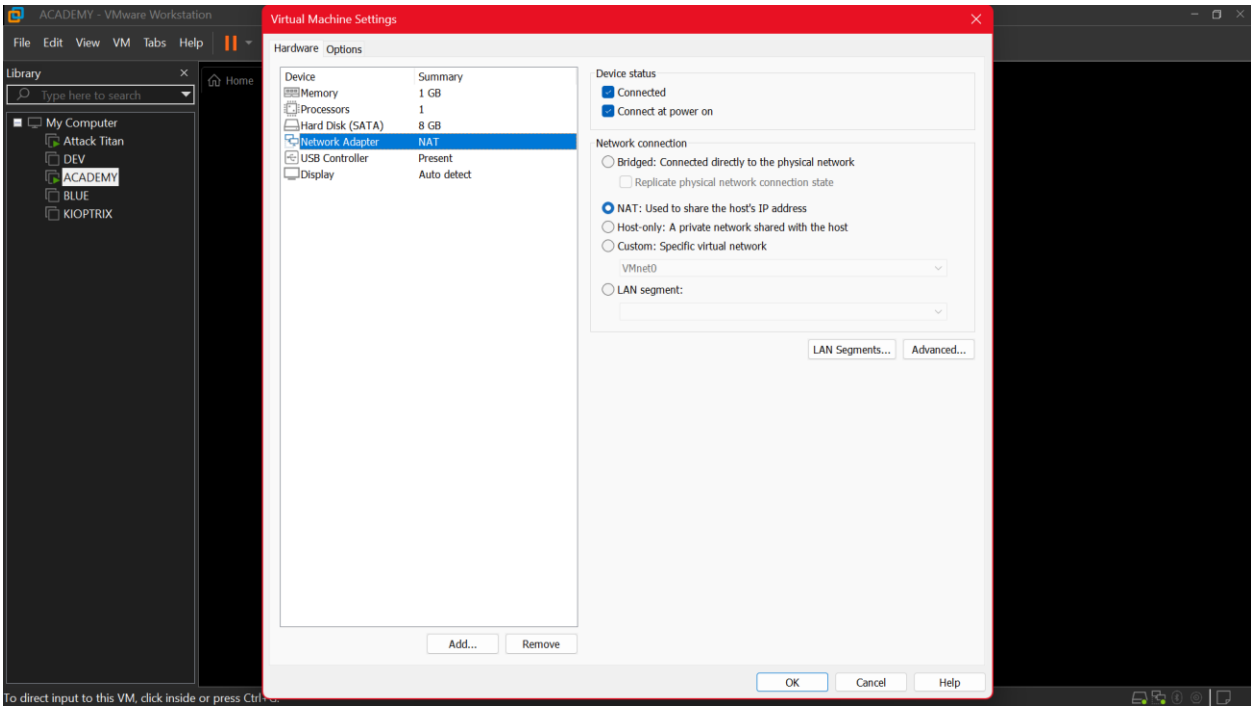
By: Lloyd Ensor Azumah

Overview

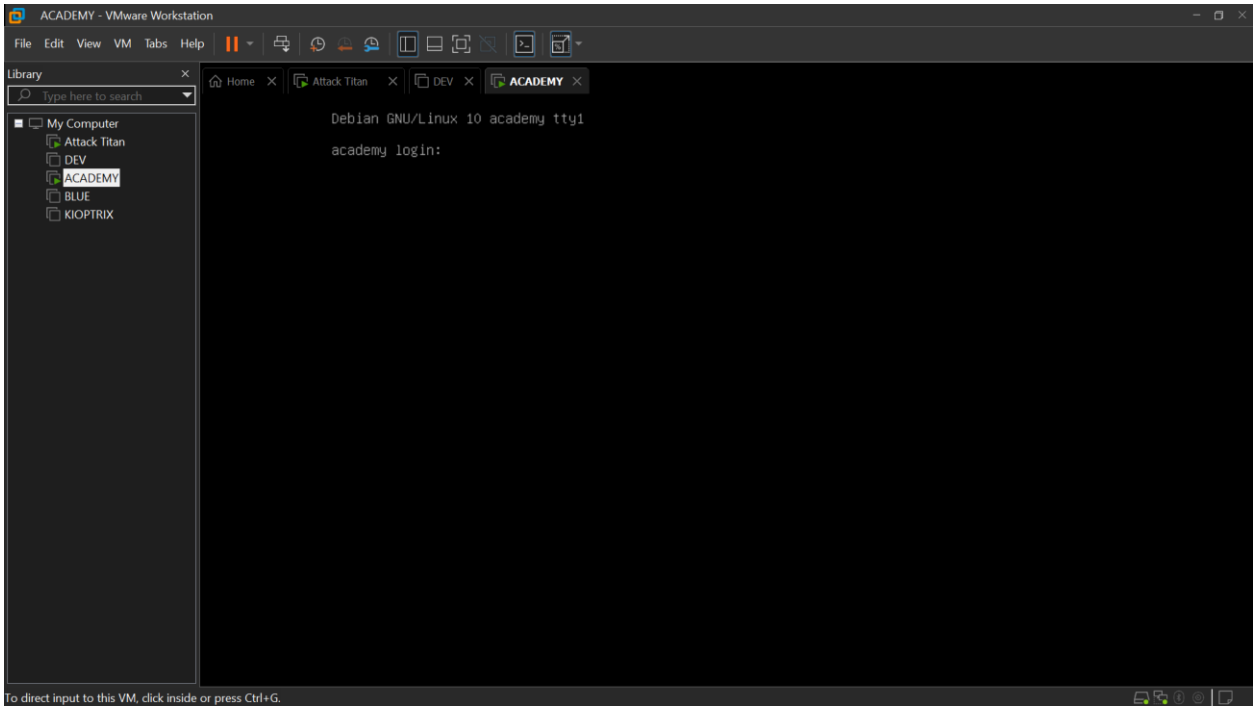
This report documents my hands-on experience compromising the Academy virtual machine in a simulated lab. Through a combination of FTP misconfiguration abuse, hash cracking, web directory discovery and privilege escalation techniques, I achieved full system compromise and root access using tools like Hashcat, FFuF, LinPEAS and PSPY.



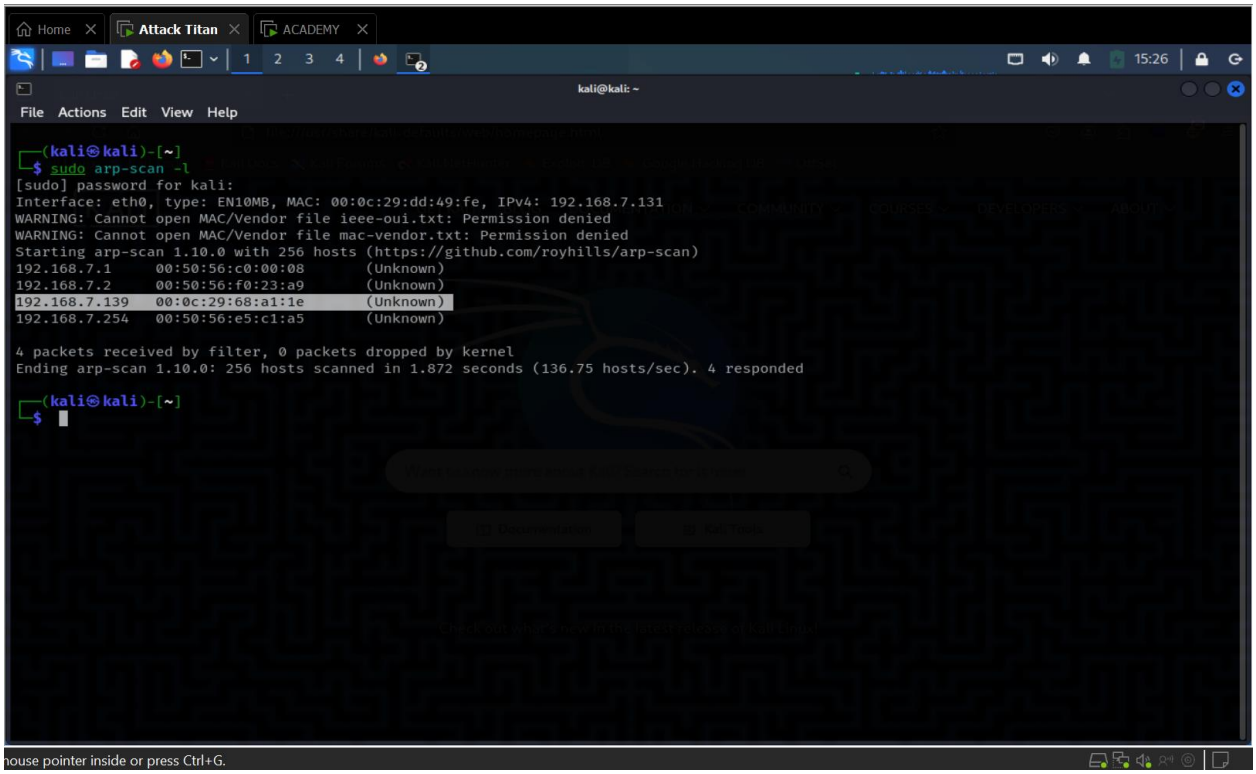
Once we have downloaded our virtual machine, we can click “Open a virtual machine” as this is an already built and configured machine and select “open” to import it.



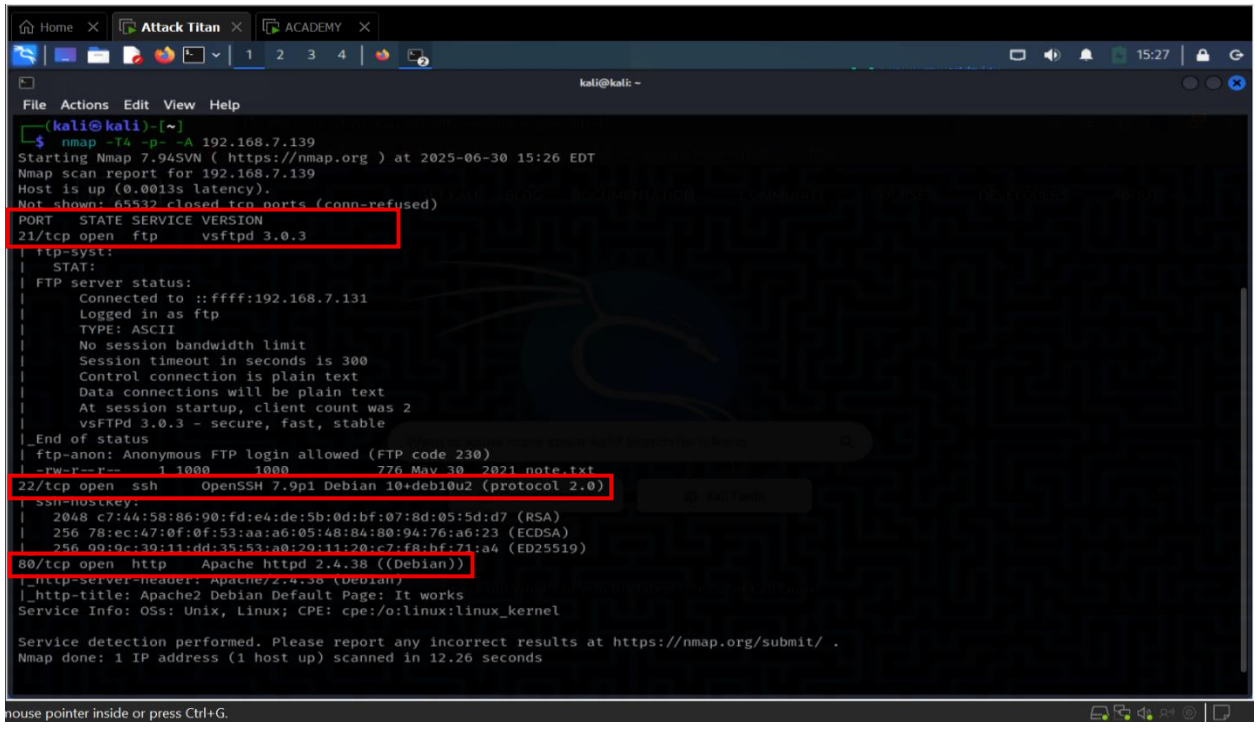
Once it has successfully imported, we will only need to make changes to the “Network Adapter” settings by converting it from “Bridged” to “NAT”. This is to ensure our attack machine is able to communicate with it (on the same virtual network).



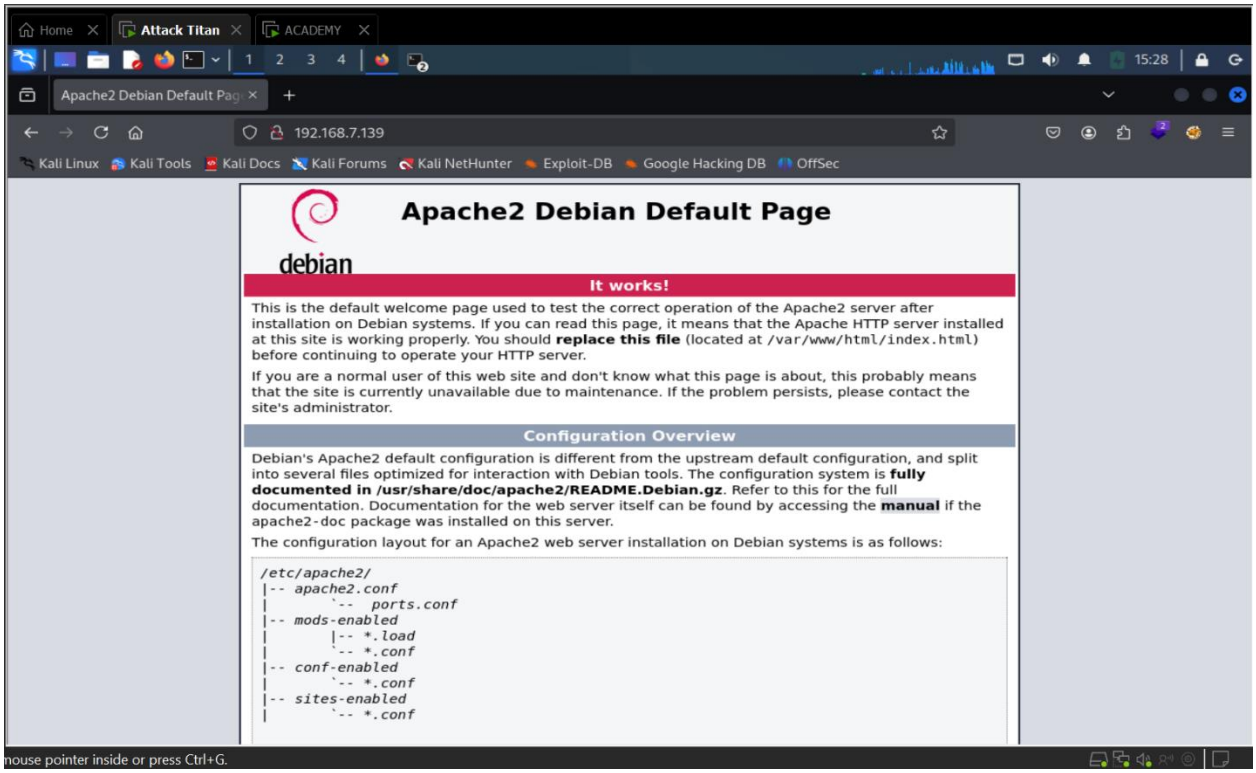
Now, we boot the machine which should to load to this interface



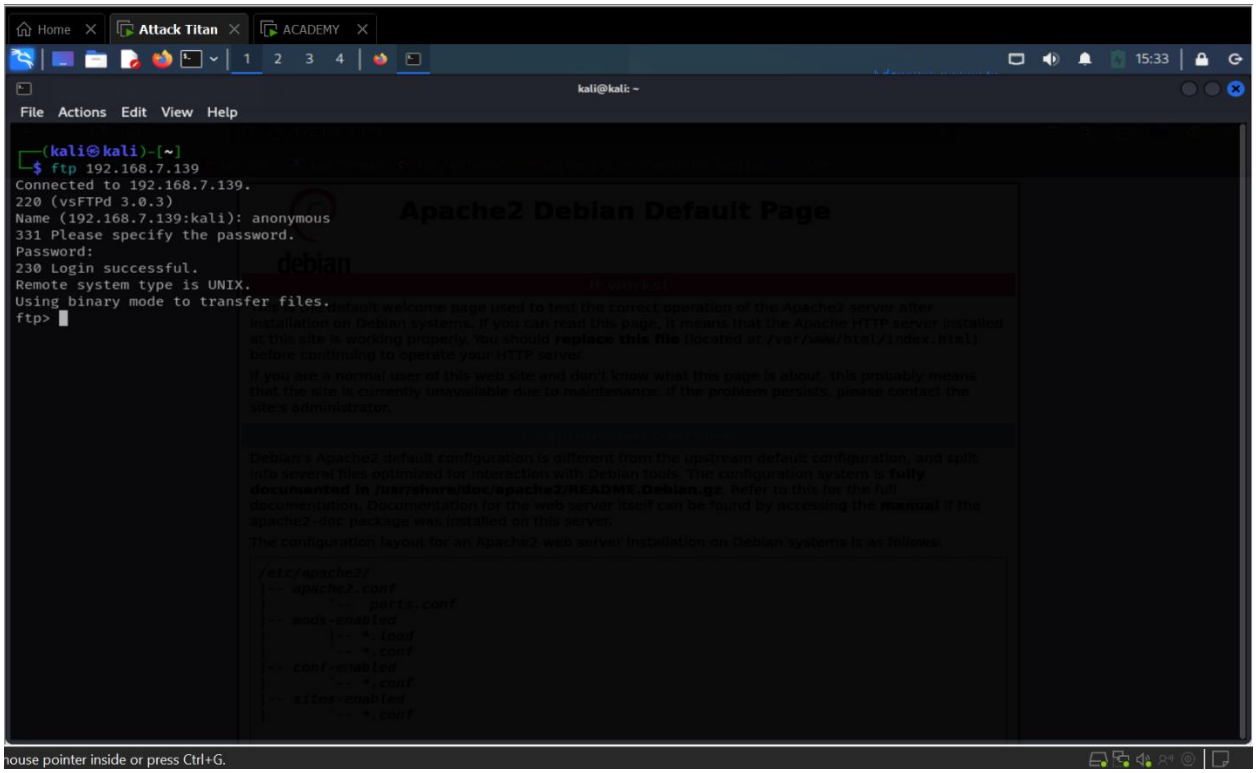
Here, we were able to spot the IP address of the target machine (Academy) on the network.



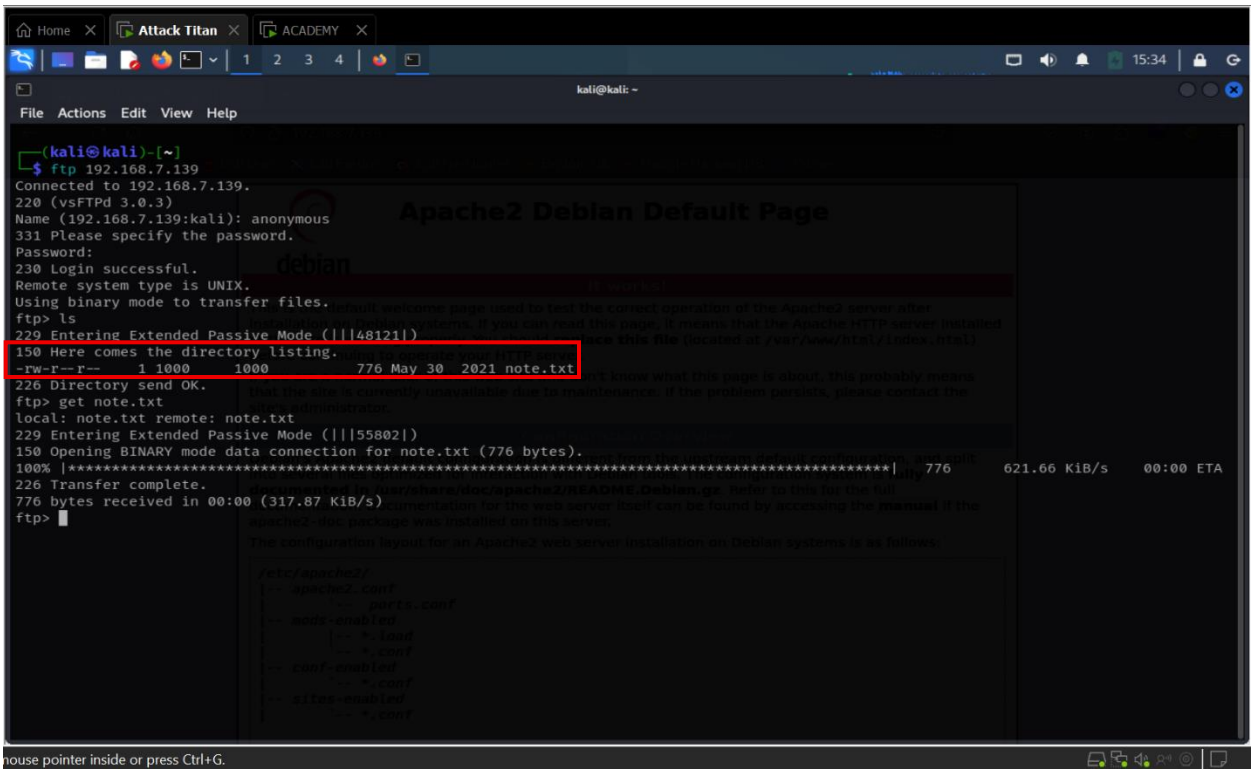
After running our **Nmap** scan, we get some interesting information. Ports such as 21(ftp), 22(ssh) and 80(http) are open. So, we start from port 80 as it tends to provide us more valuable information by searching http://192.168.7.139 on our browser.



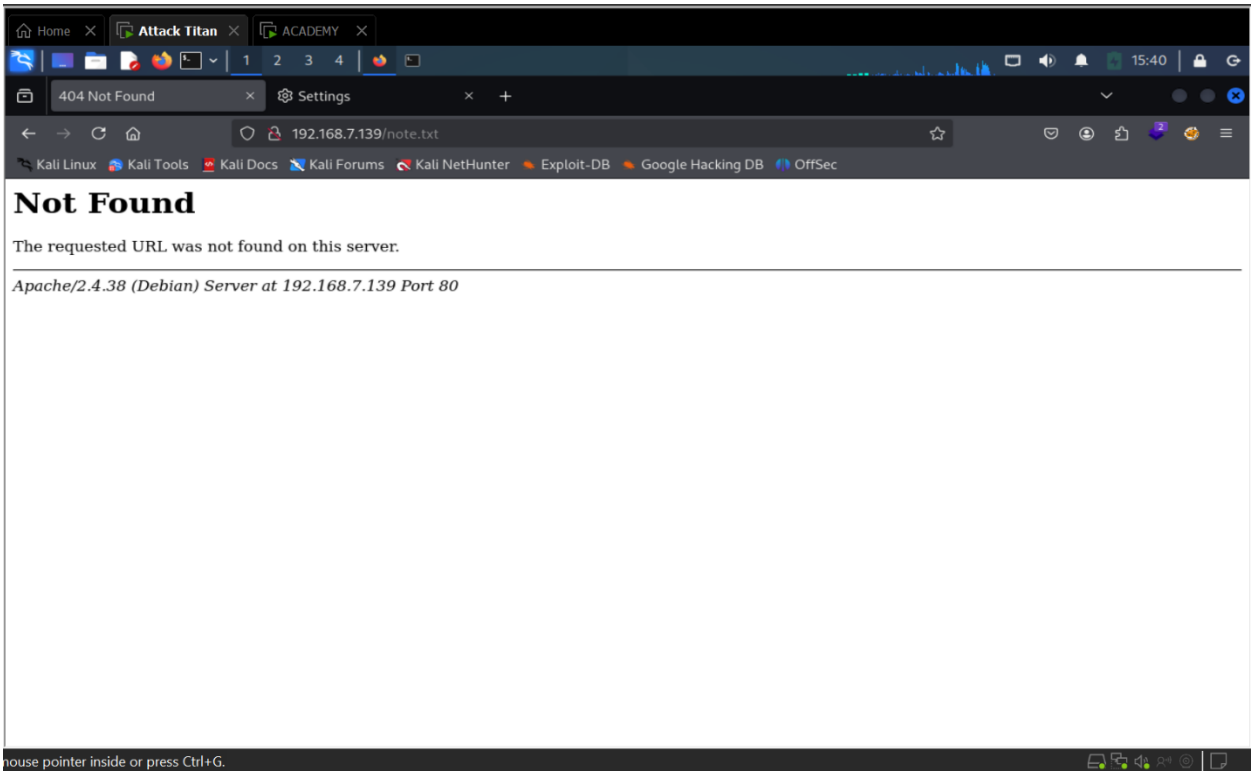
Looking at the website, there isn't much we can put together that could give us a lead. So we try our next port which is going to be port 21.



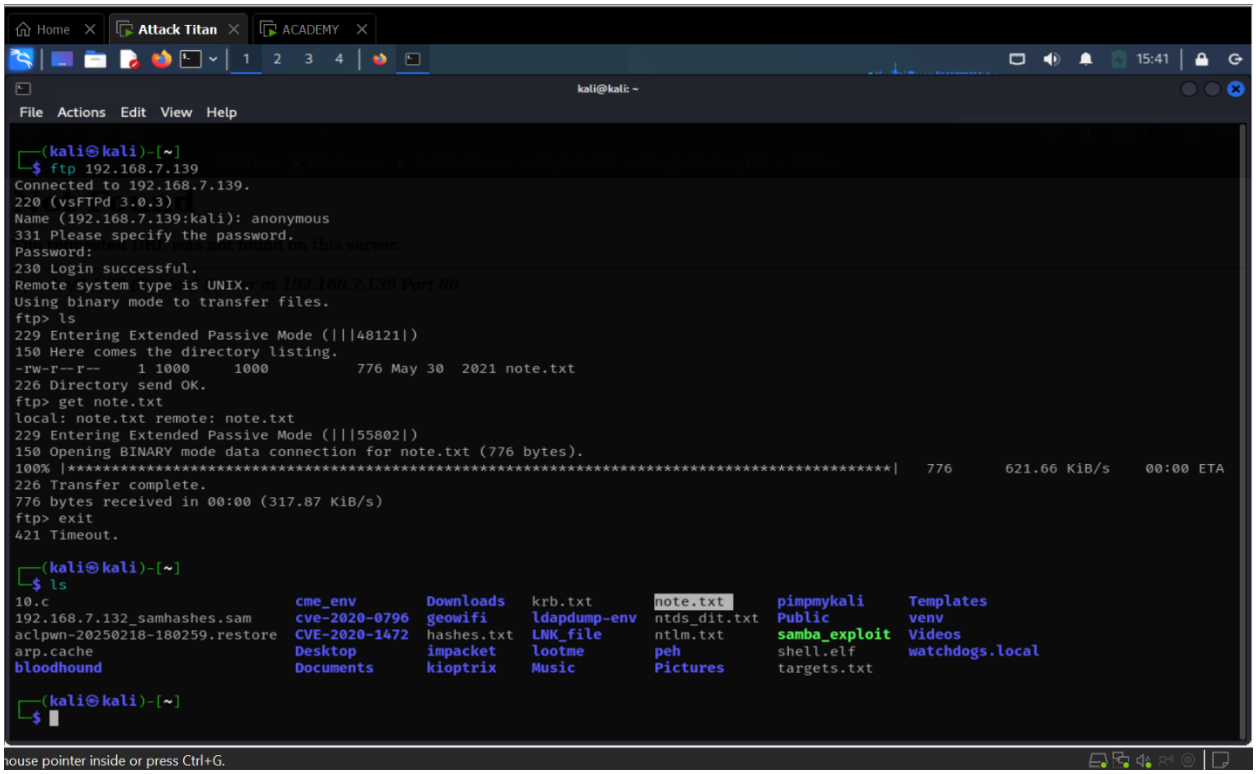
On port 21(file share protocol), this suggests it could contain files with credentials of users or provide us a clue we could use to our advantage, so we connect using "ftp ip_address". NB: If you remember our **Nmap** results, we were told "*Anonymous FTP login allowed*" meaning we can login with the username "anonymous" and for curiousity, repeat that for the password.



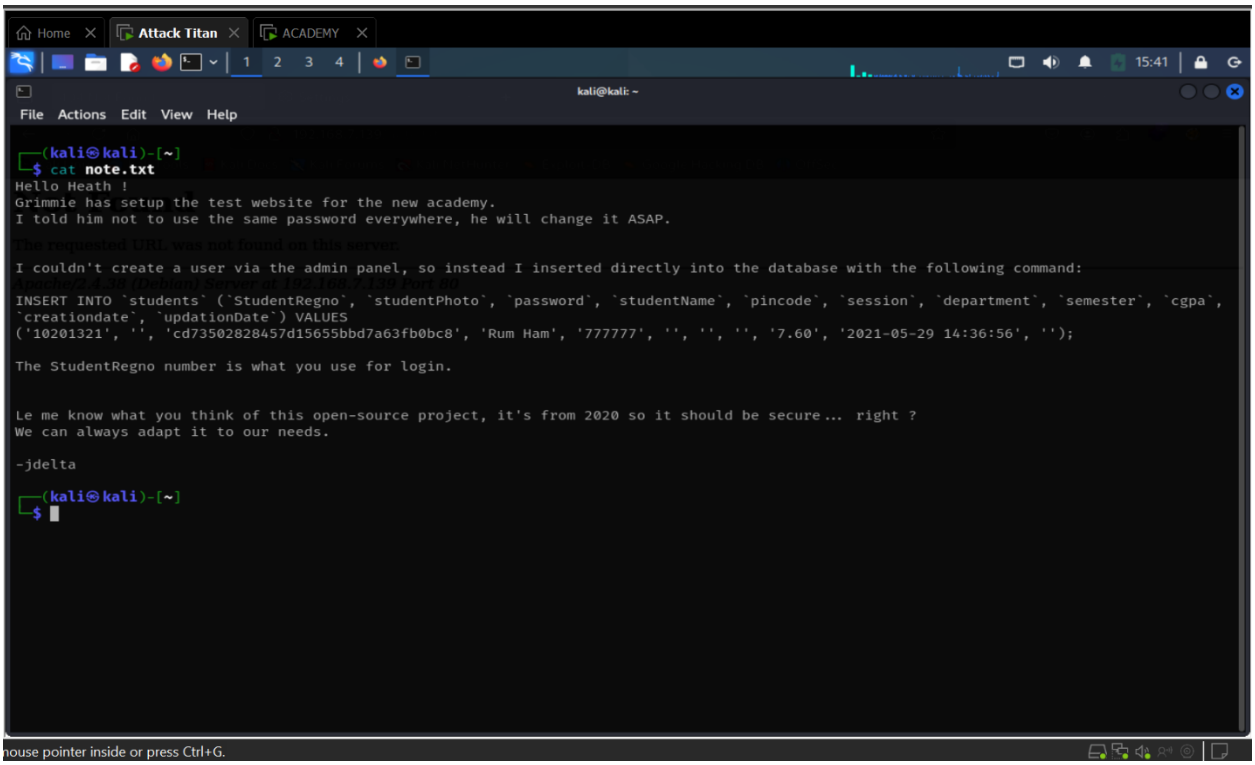
Our login was successful and we can procede by enumerating directories and files in the it. After doing this we notice a file called “note.txt”. We download it using the “get” command



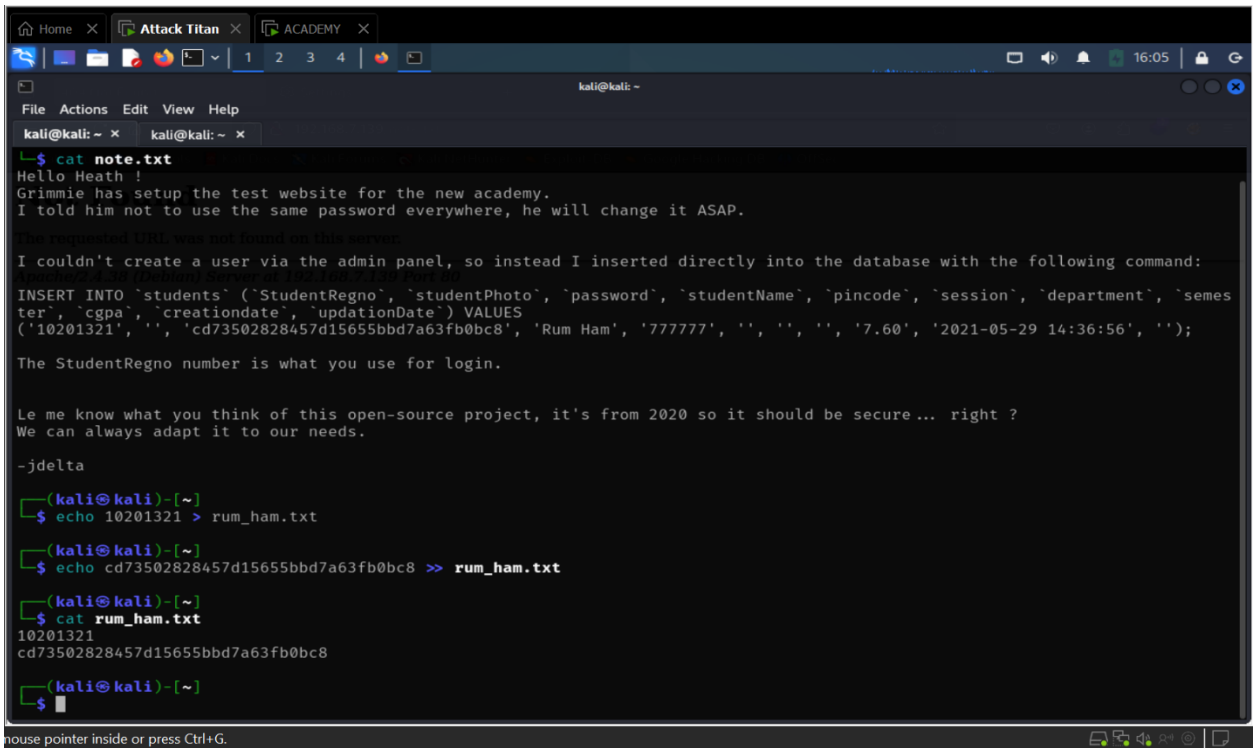
As usual, curiosity gets the best of us so we try to see if the website also hosts this file.



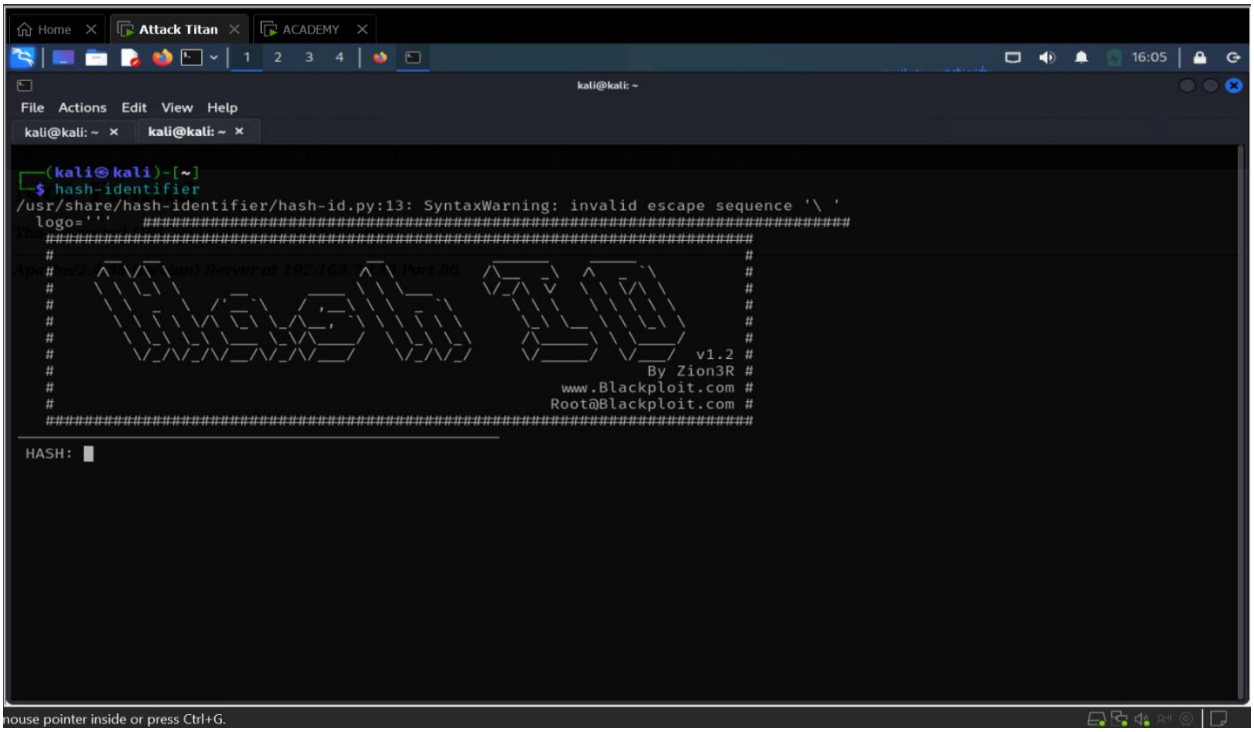
As we can see. the file we downloaded is visible and so we take a peek at it.



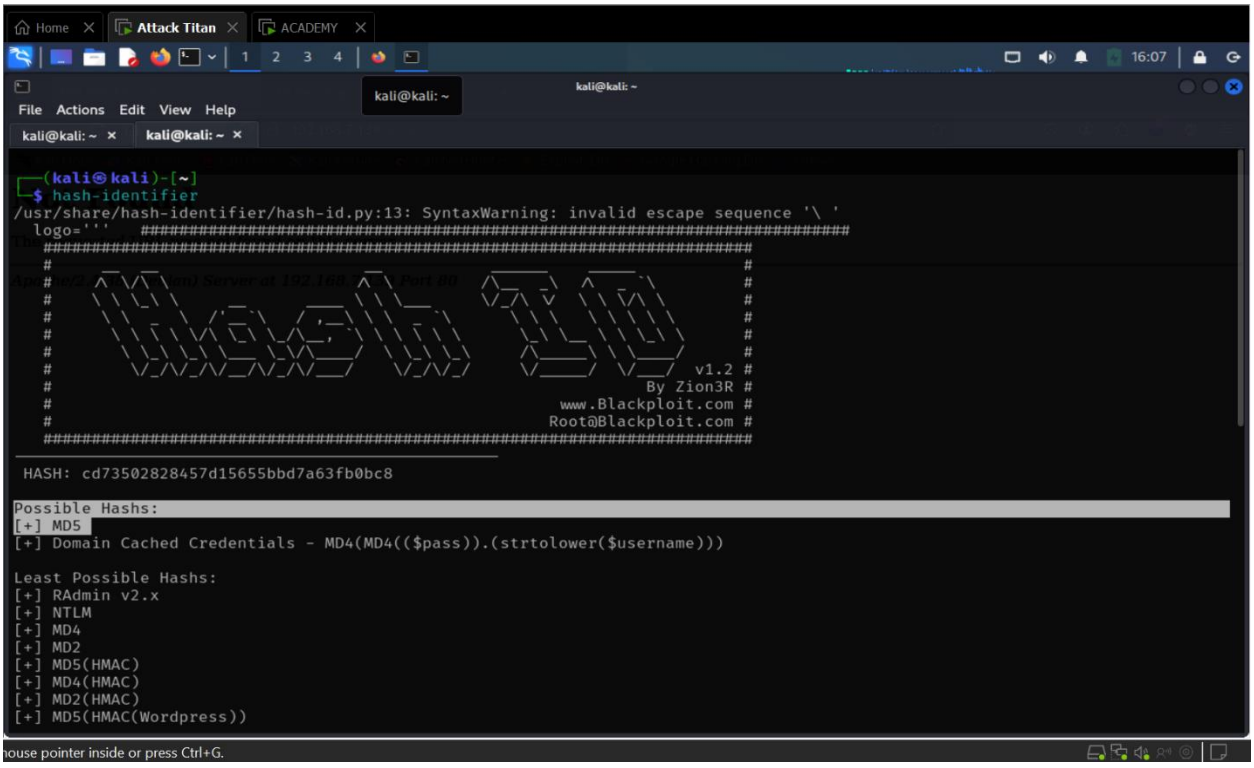
Finally, we got some juicy information talking about an academy website, student registration number, password, student name, etc. It also says the “Studentregno” is to be used for the login.



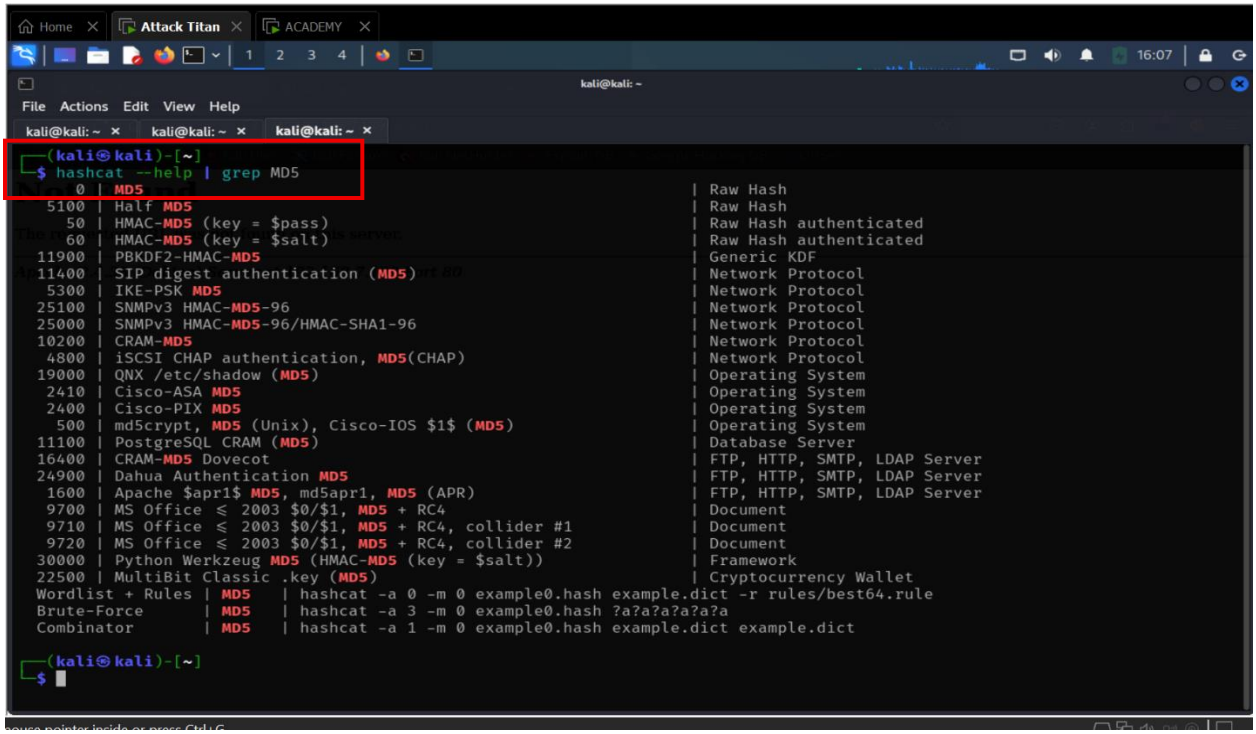
All we have to do is cut out what is needed, if the “studentregno” is the login(username), then the alphanumeric value should be our password but here, it is in a hash format. We will need to crack it before we are able to use it in this context. These credentials are stored in a text file.



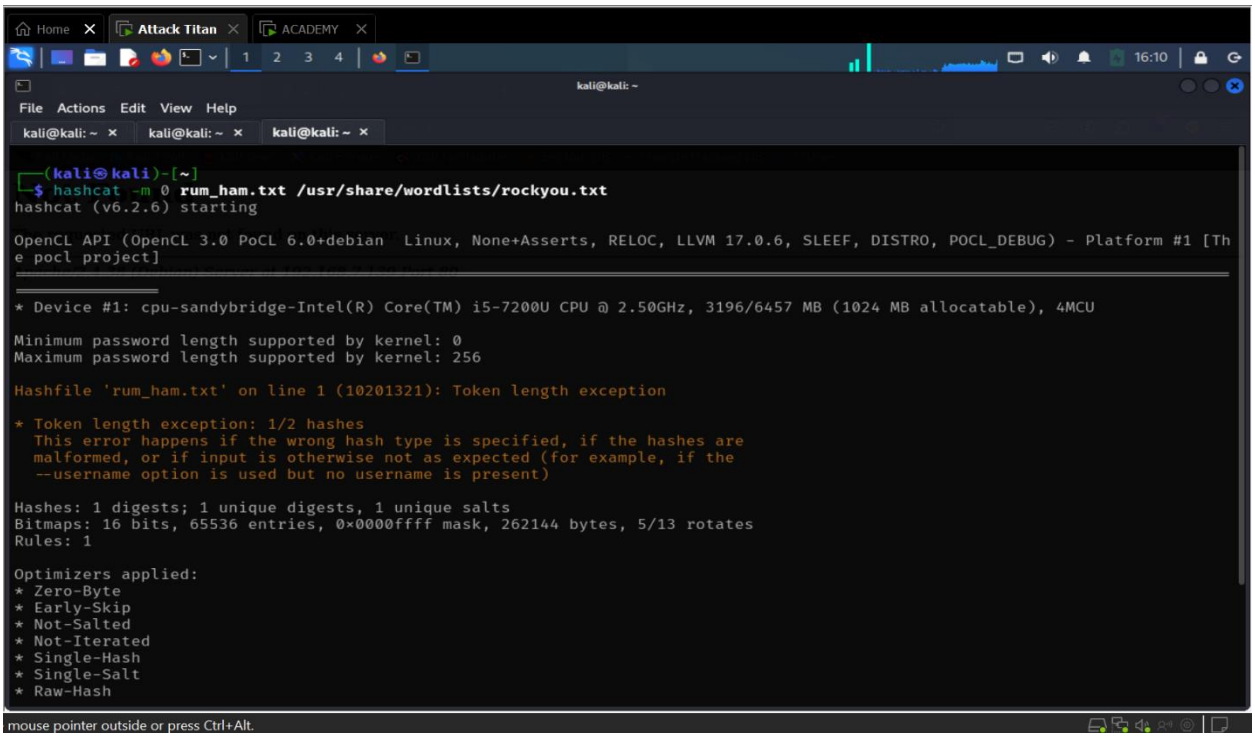
First of all, hashes come in all sorts of format and we need to know what hash format our password is using. To do this, we use a tool called **Hash-ID** to determine that.



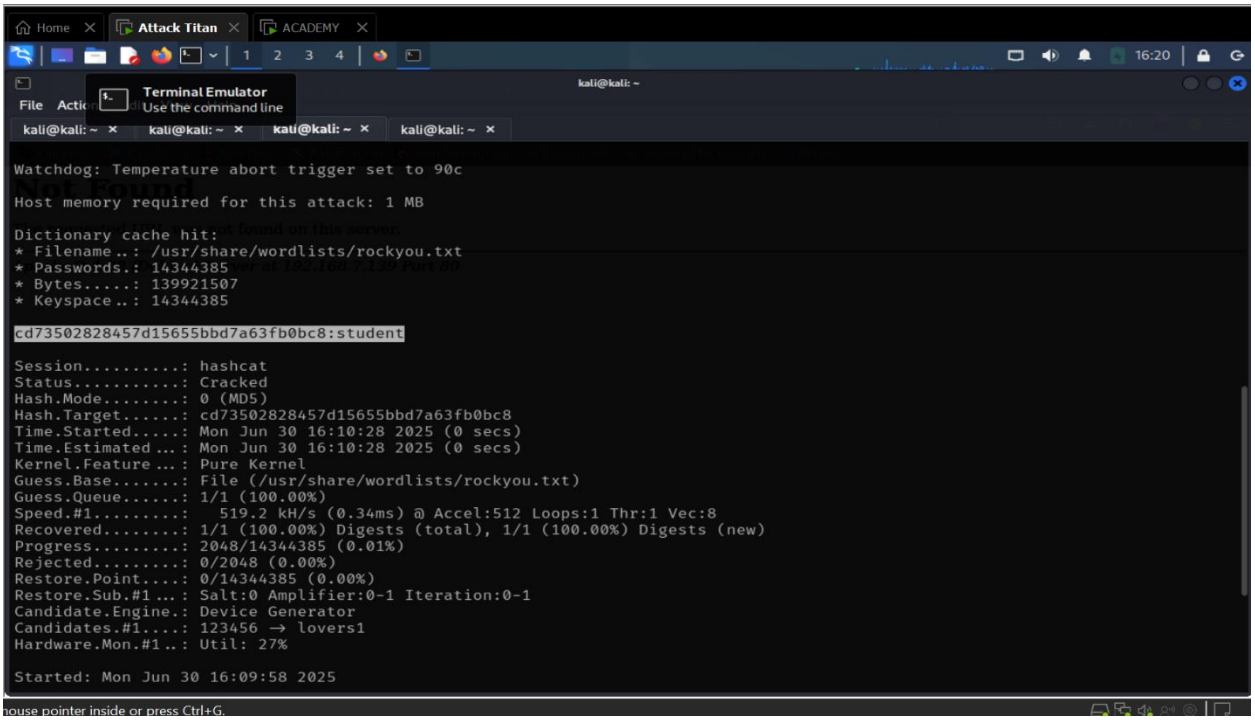
After pasting our hash in there, we are provided our “Possible hashs” (MD5). We will use this information to specify the format of the password (MD5) we want to crack.



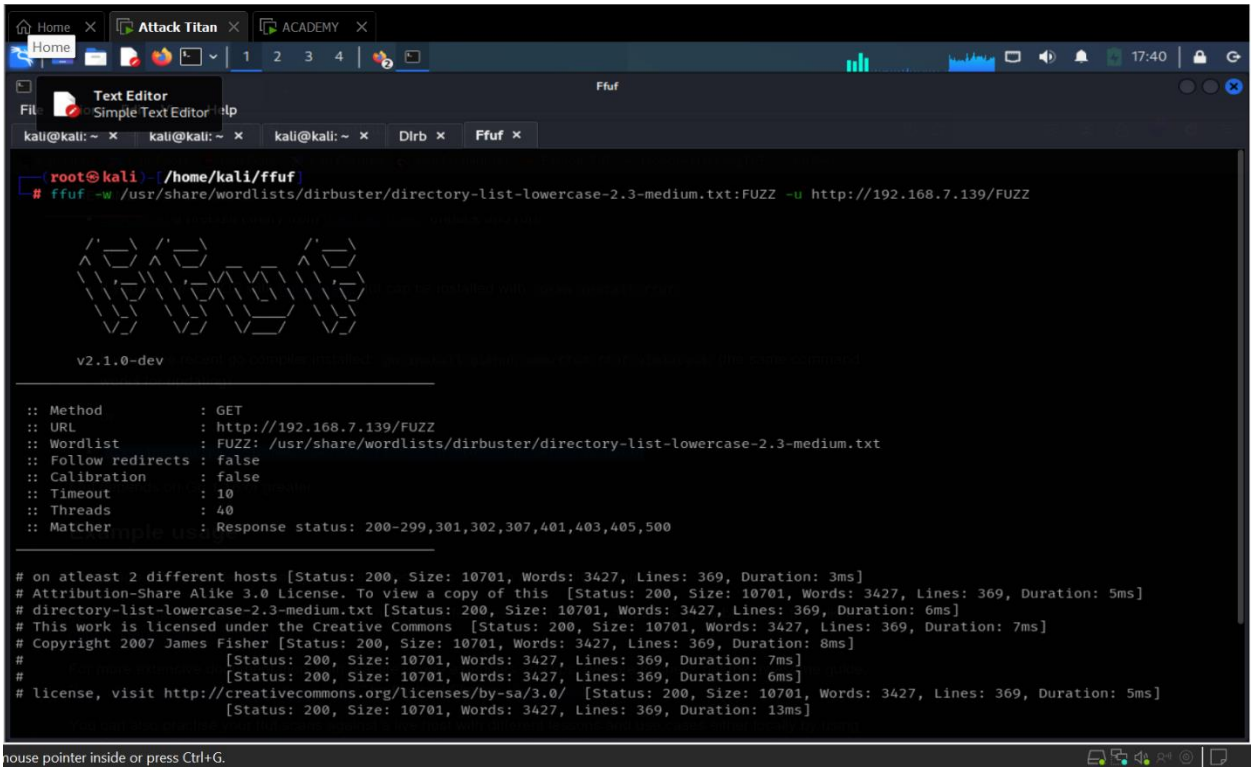
Now for the cracking part, we use the tool called **Hashcat** to crack the password. As we can see, we are looking for the MD5 module we will use to crack our hash.



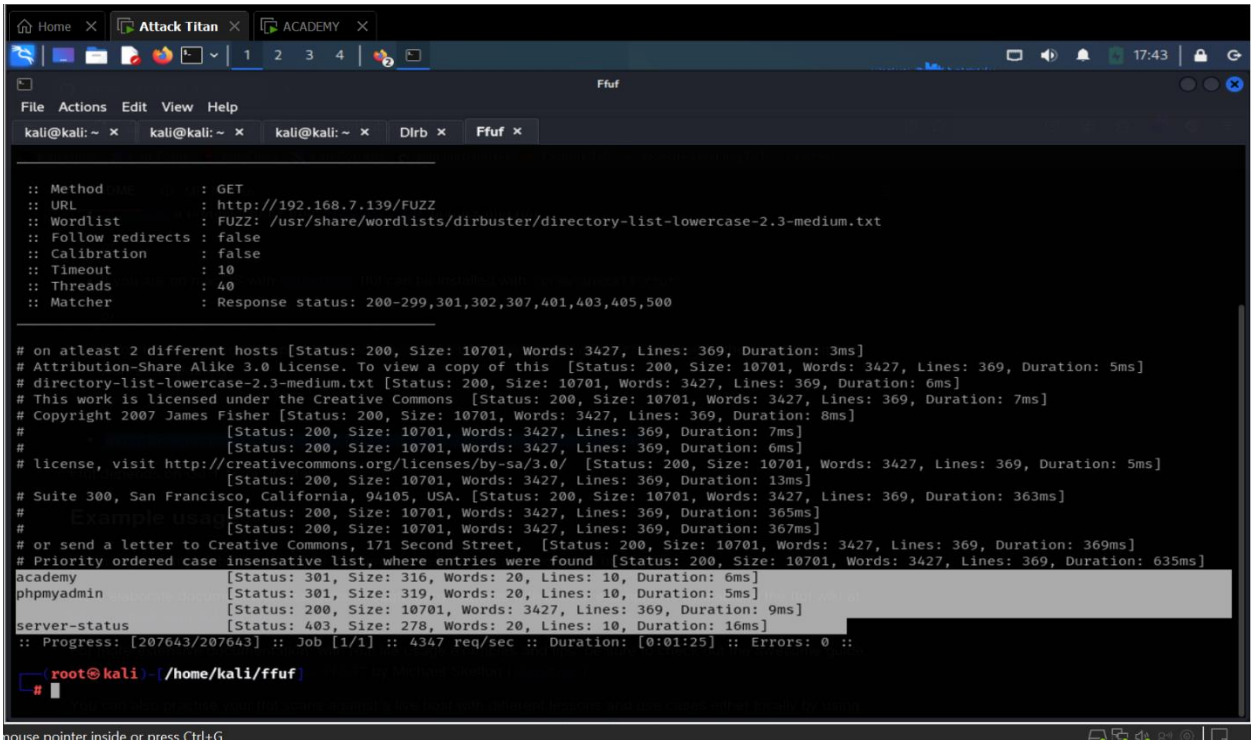
When cracking, “-m” used to specify the module format and “0” the MD5 module we are using. “rum_ham” is the text file we that contains the hash.



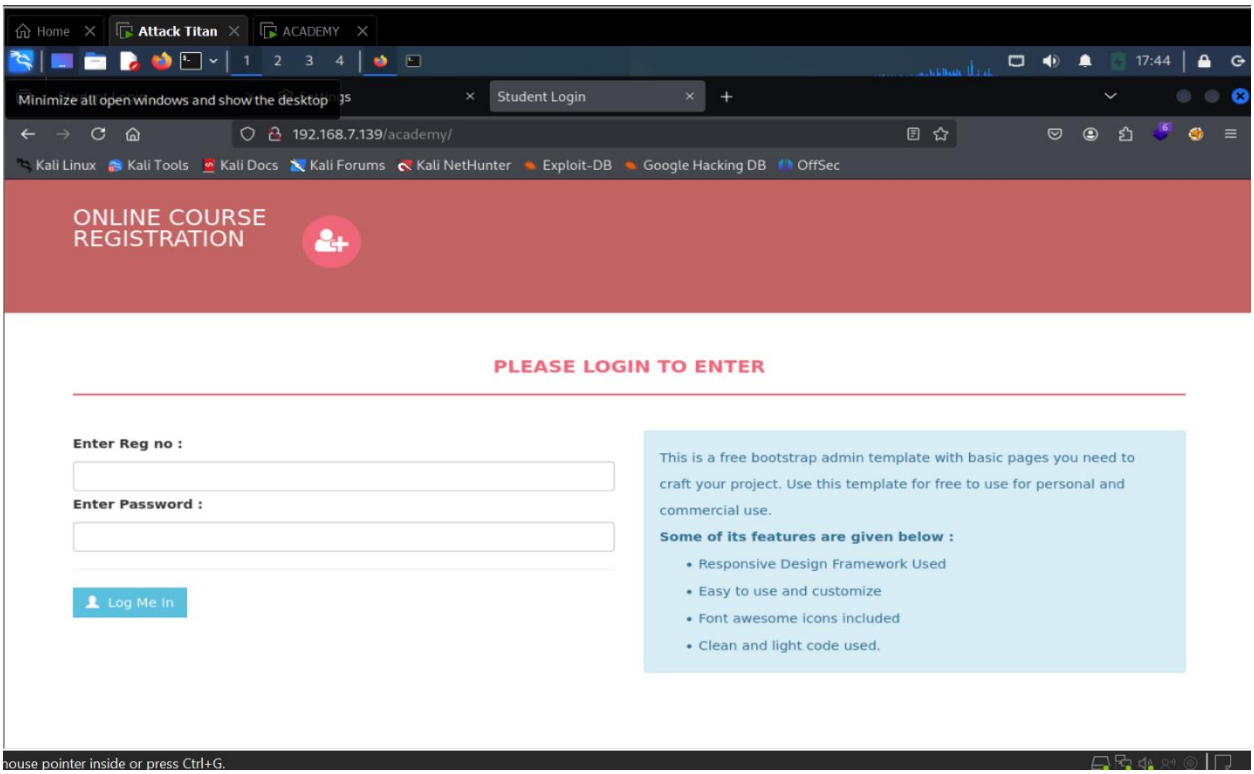
After a couple of seconds, the crack was successful and we have the password in plaintext (student) as highlighted. Now, we have to find this academy site in order to even use these credentials in the first place.



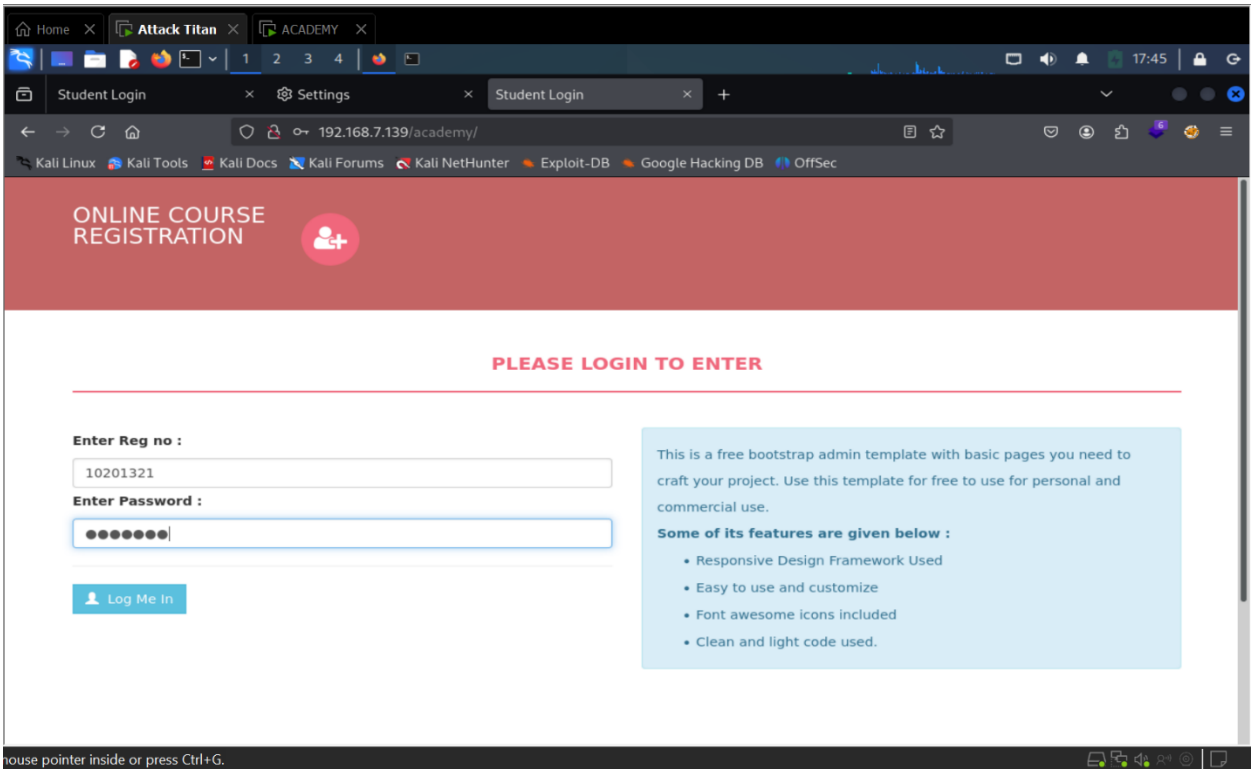
Here, we will dig out hidden directories within the website that could lead us there by fuzzing it. The **FFuF** (Fuzz Fast u Fool) tool is a great choice for this.



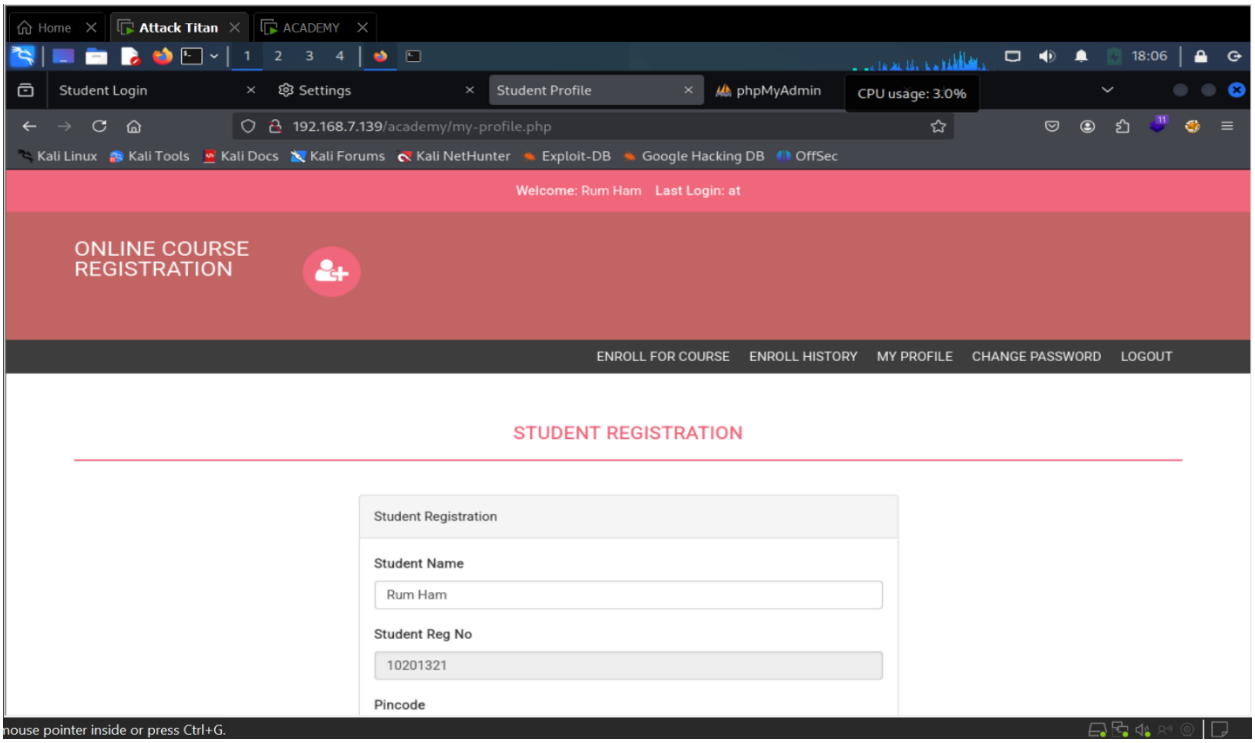
Eureka!, this time we found something more interesting. The directories that instantly draw our attention first is the academy as this is what we were hoping to see.



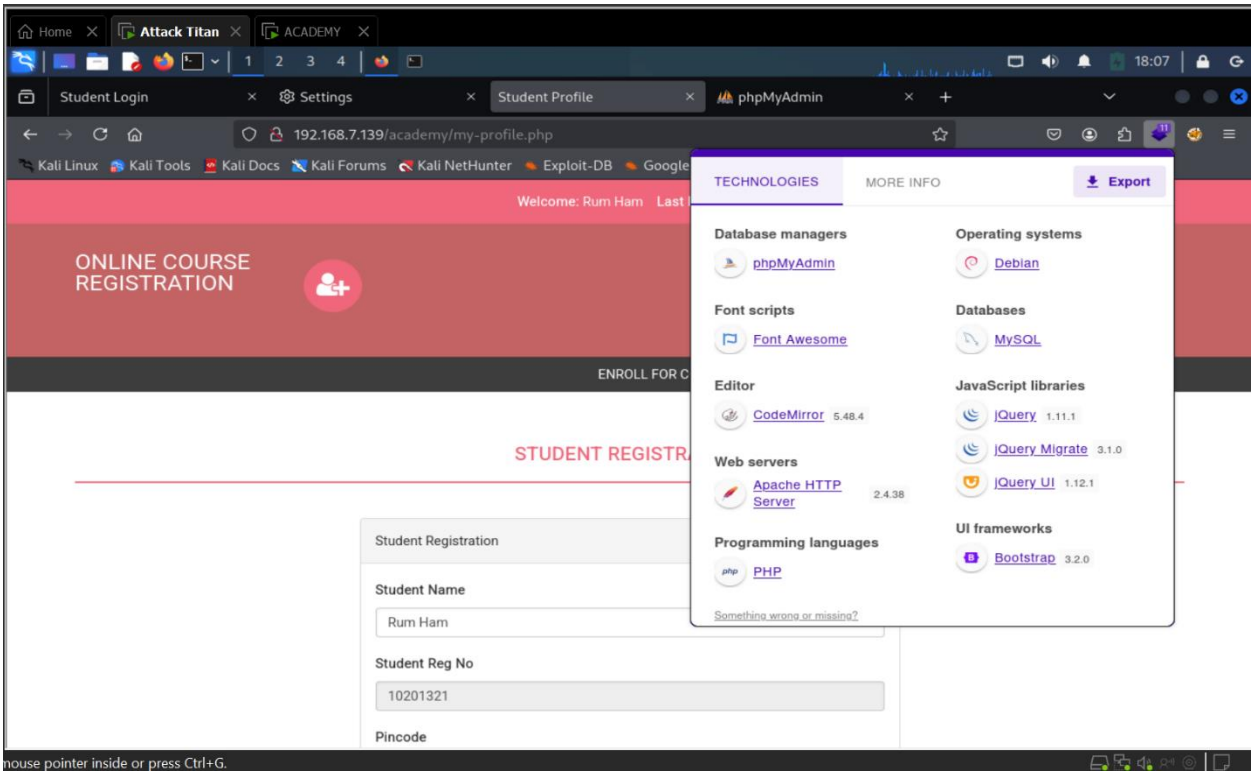
So we navigate to http://192.168.7.139/academy/ and we are brought to this interface.



We insert our credentials accordingly and click “Log Me In”.

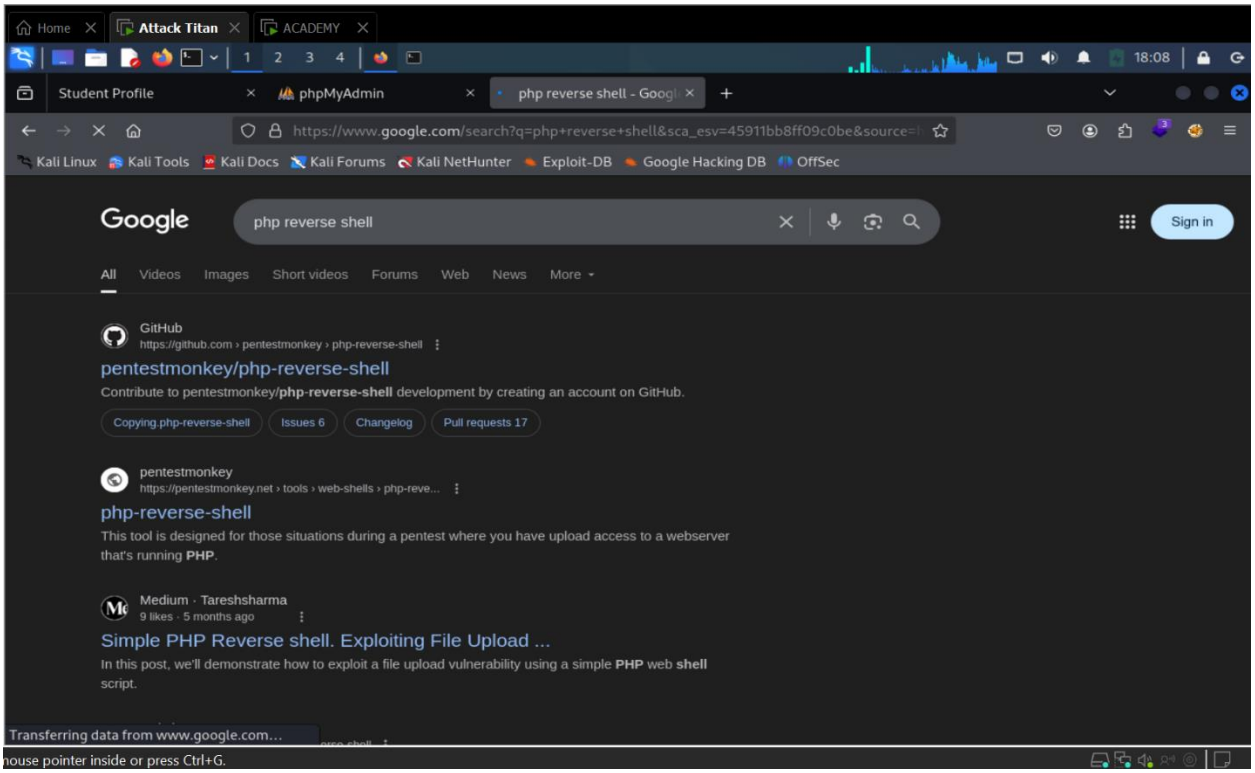


Voila, we are in the academy platform as we can see. Some exploring is done just to see how the platform has been setup and what features are in place.

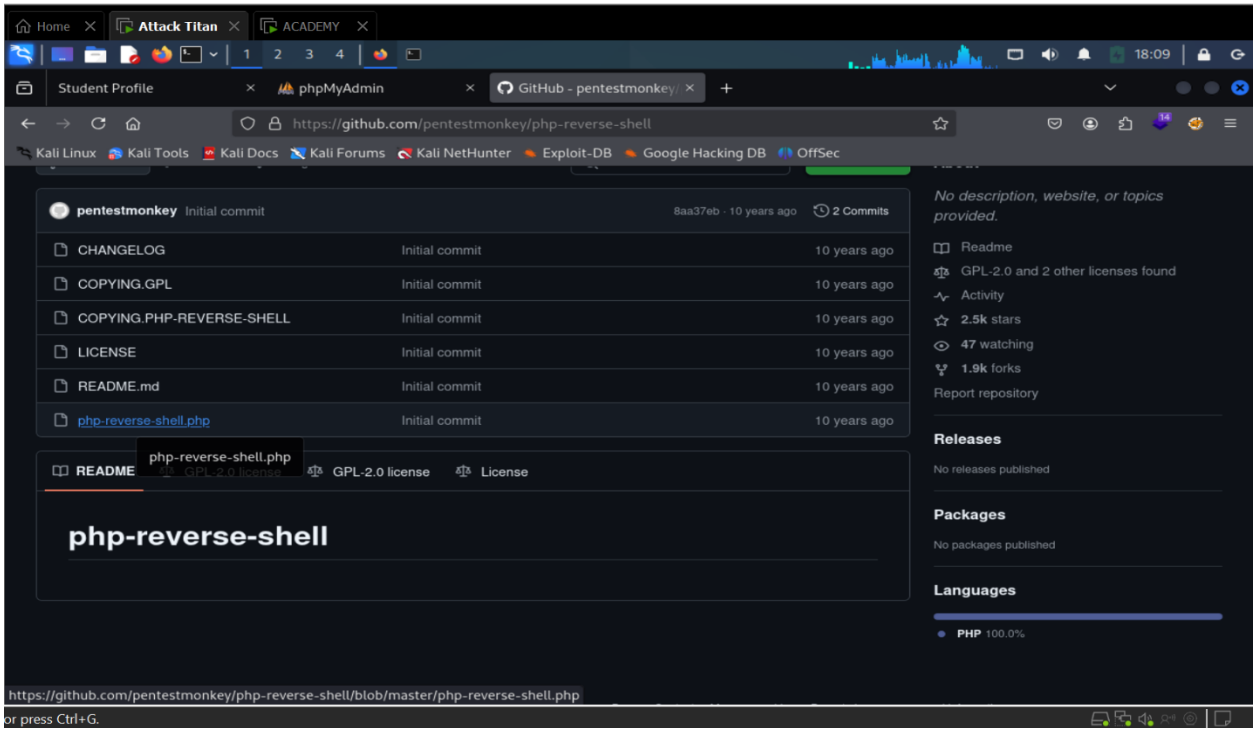


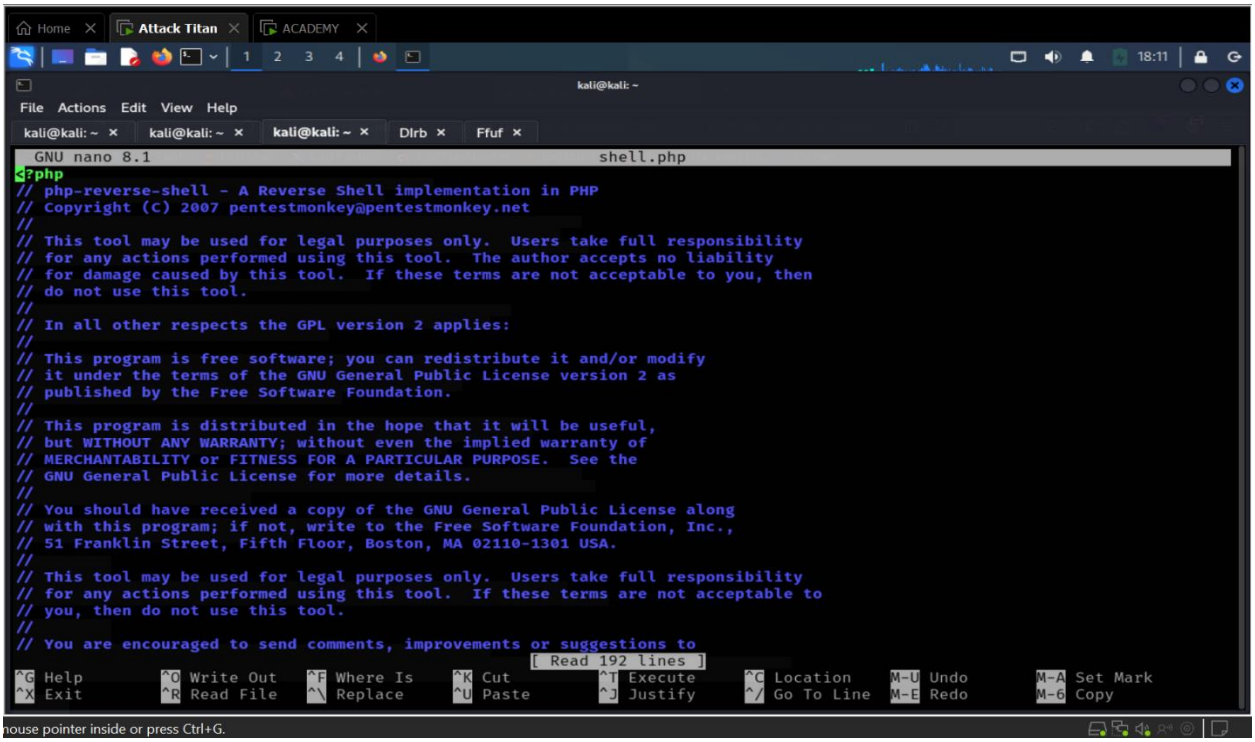
With the help of our **Wappalyzer** extension, we can also see what's running under the hood which can be used to our advantage. While exploring, we noticed that the platform allows users

to upload pictures for their profile picture. As long as the input hasn't been sanitized, then we can upload a malicious file that can spawn a shell or grant us access.

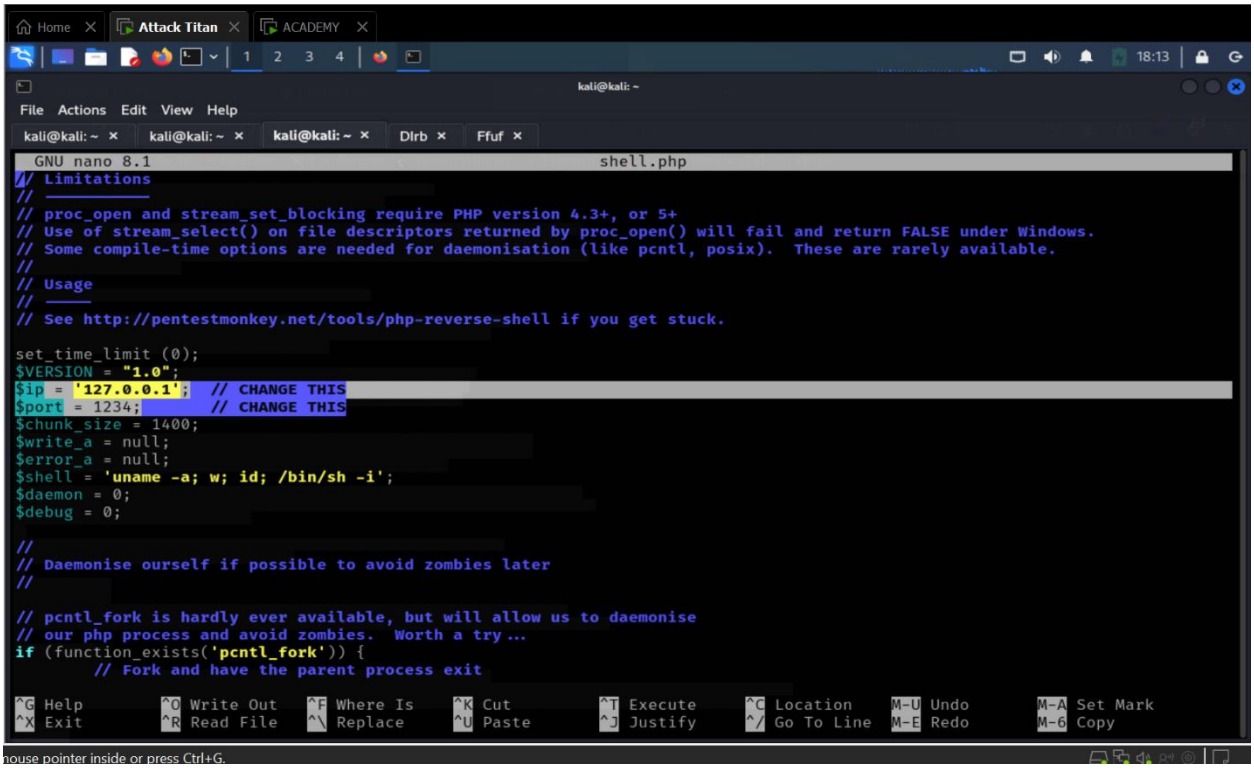


Since the platform was built with the PHP language, we can utilize a reverse shell written in PHP.

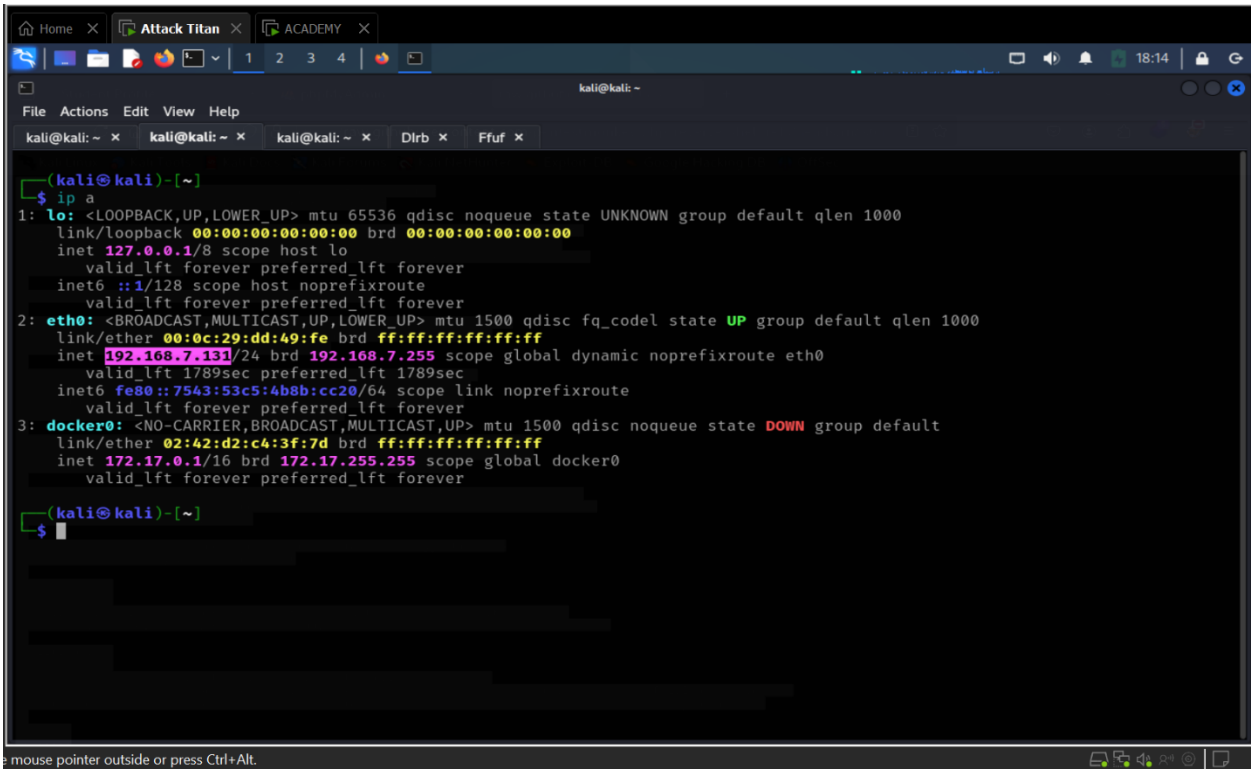




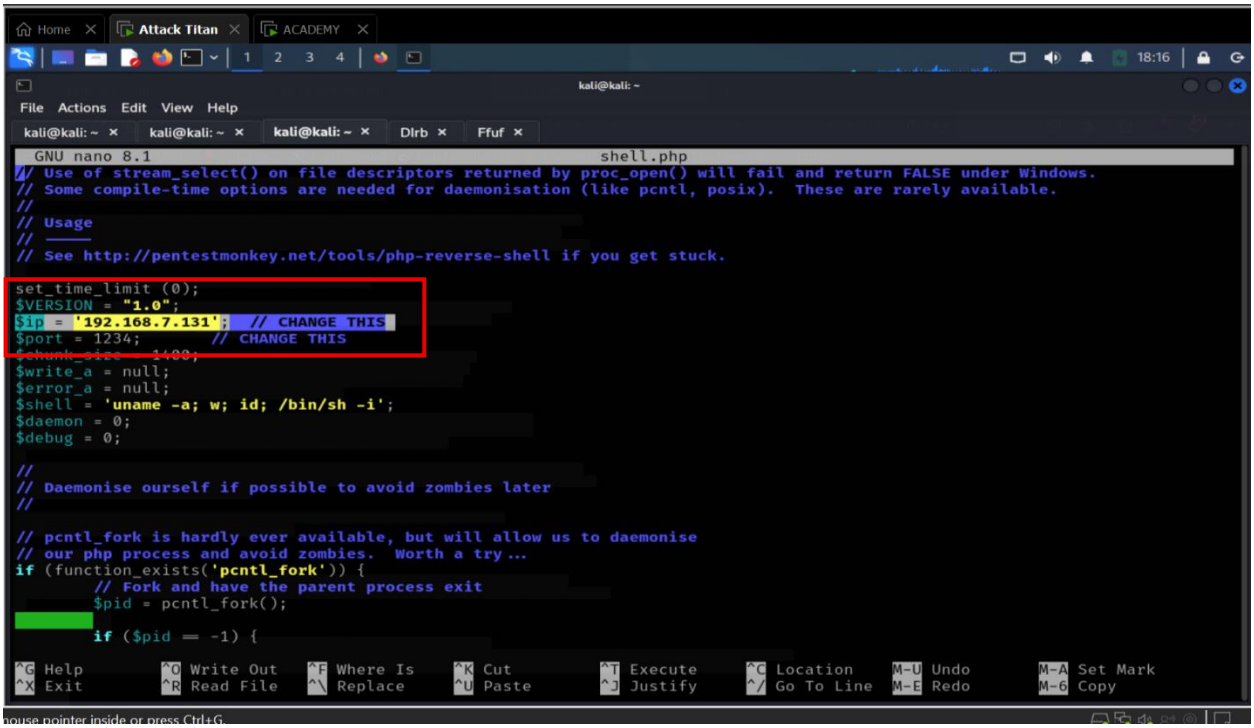
After downloading the file, this is what it looks like.



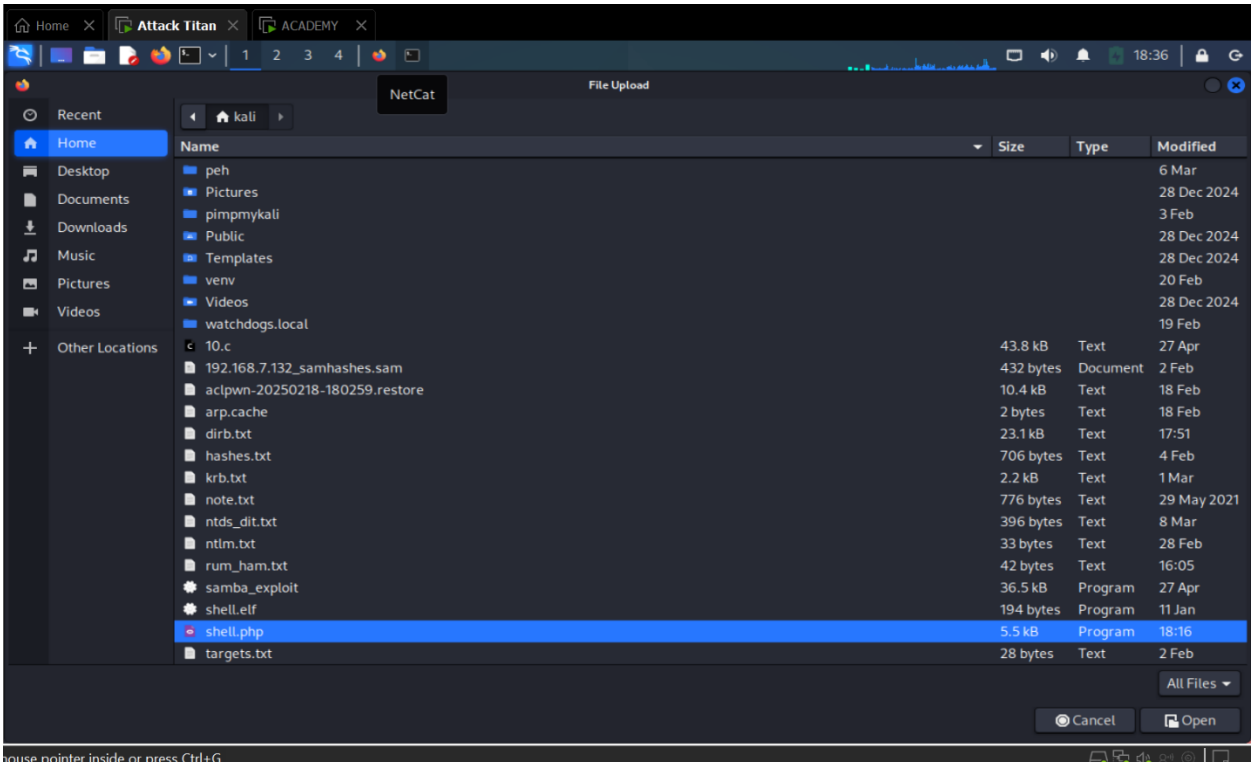
When we scroll through the lines of codes, there are some comments suggesting that there are some changes that should made such as inserting our attack machine IP address and port number in order for this to work.



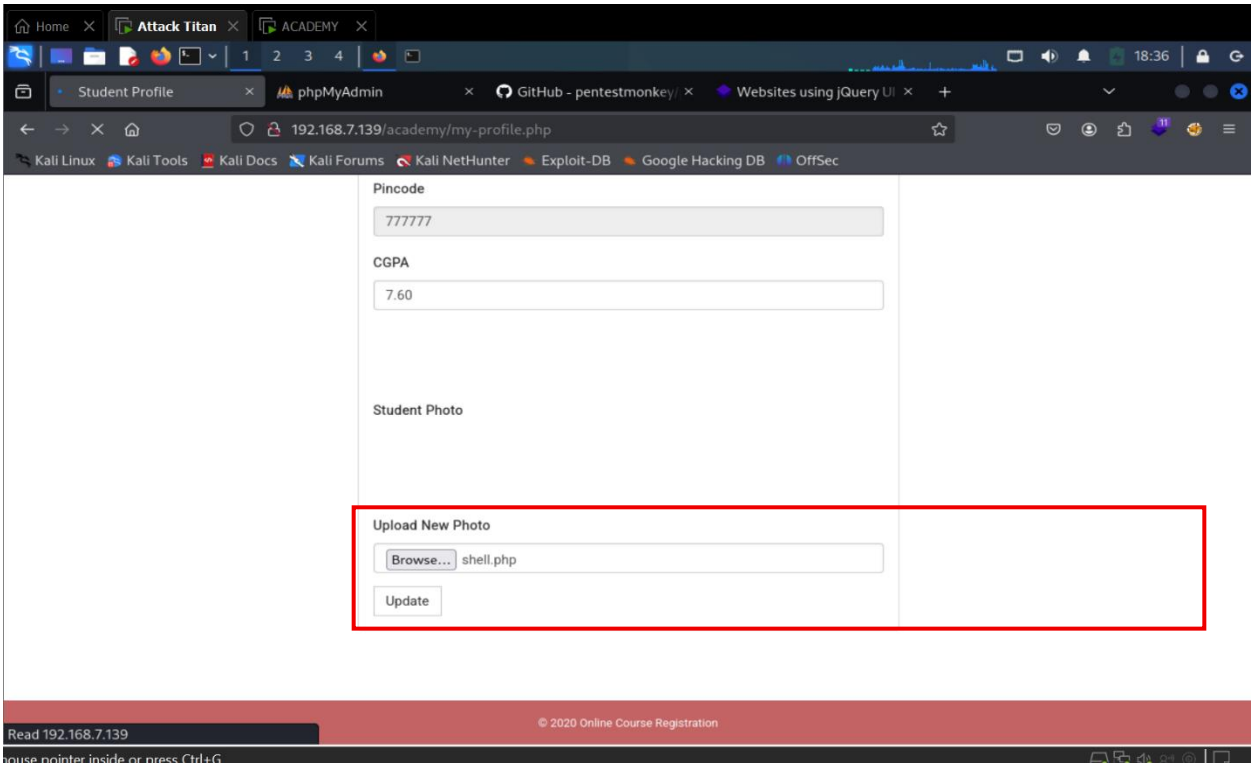
Just for confirmation, we verify our attack machine's IP address with "ip a". The highlighted IP address is ours.

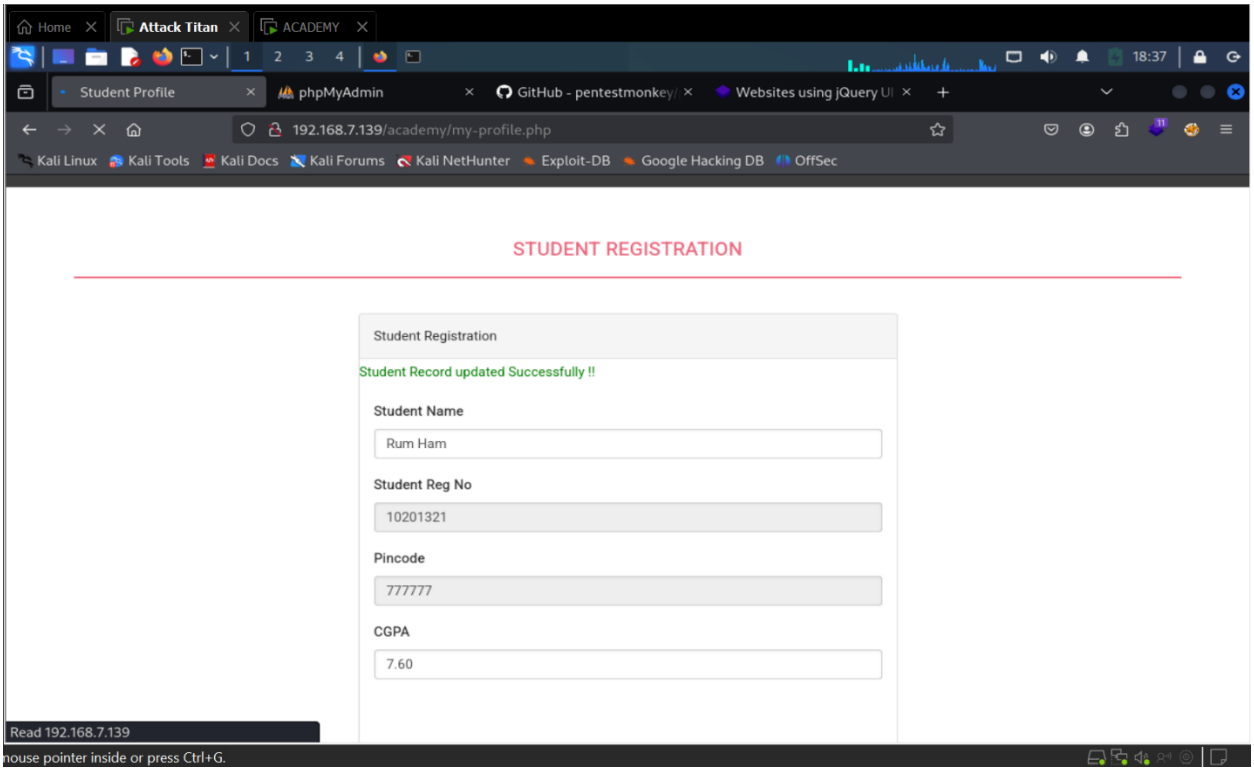


We insert our IP address and left the port number as its default (1234) and saved the file as "shell.php"

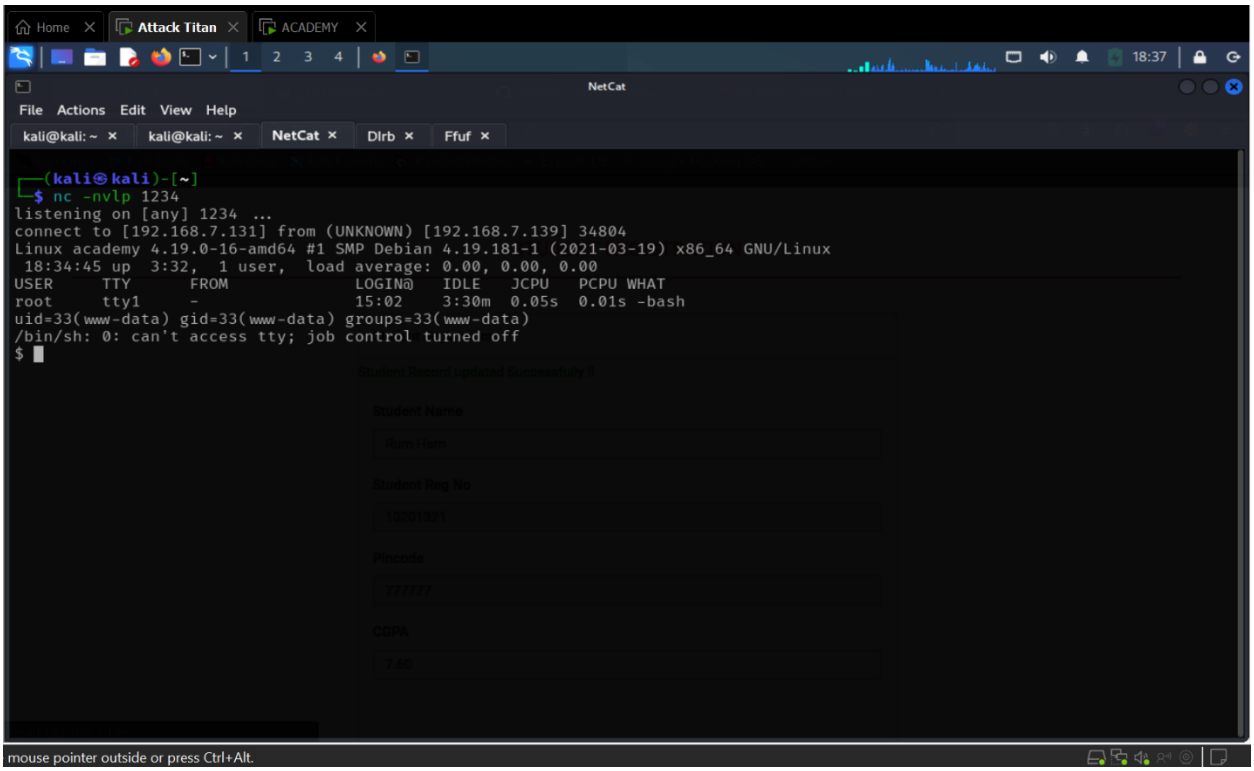


Now we upload the “shell.php” file onto the academy platform and see if there are any resistance.

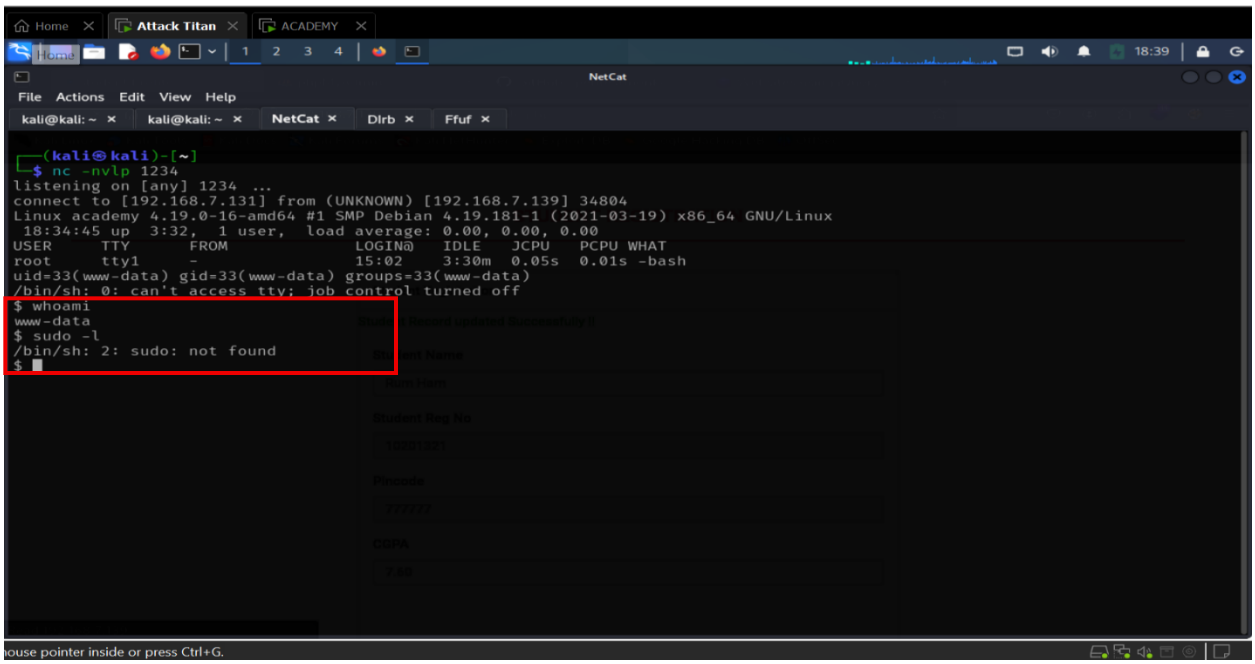




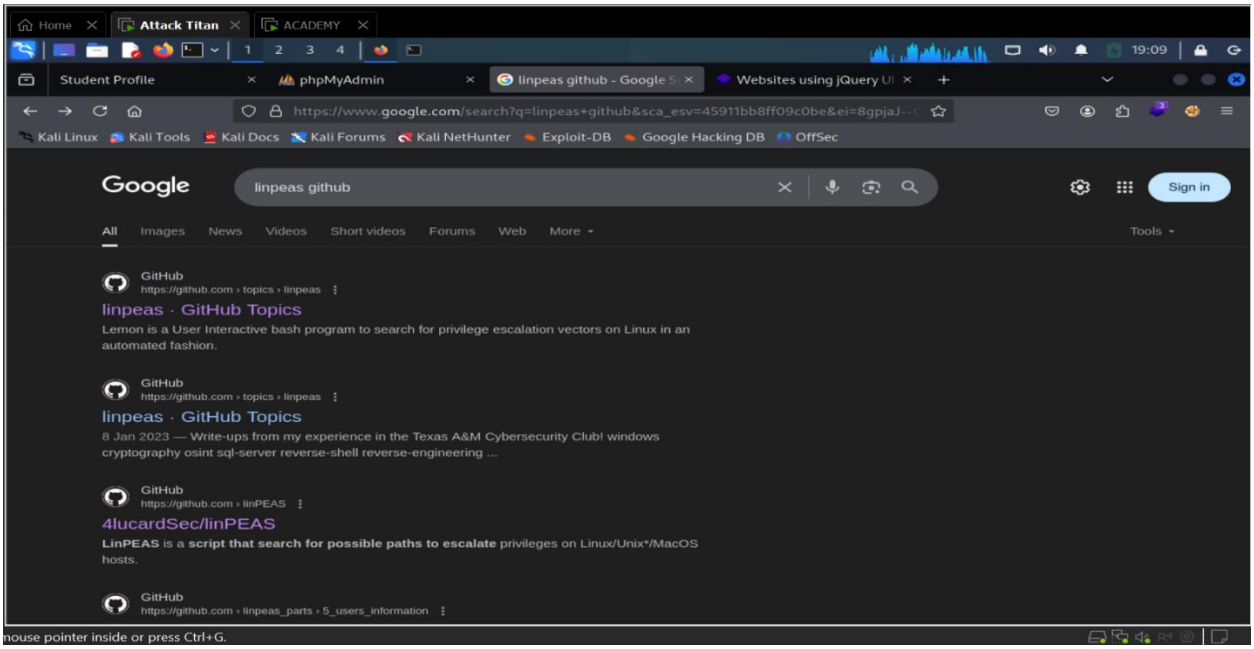
And our upload was a success with no troubles. Now we can setup a listener using the **Netcat** tool using port 1234 if we can recall.



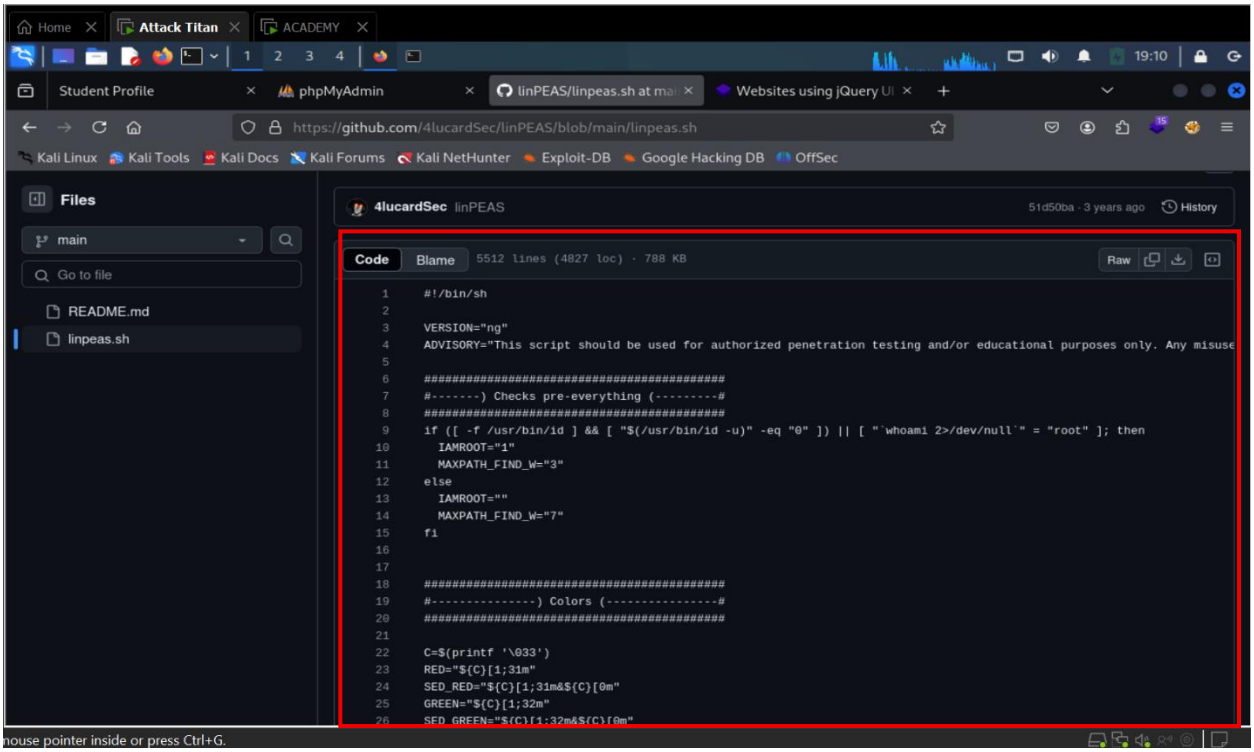
So,124 our listener is setup and after a few seconds, we were able to spawn a shell.



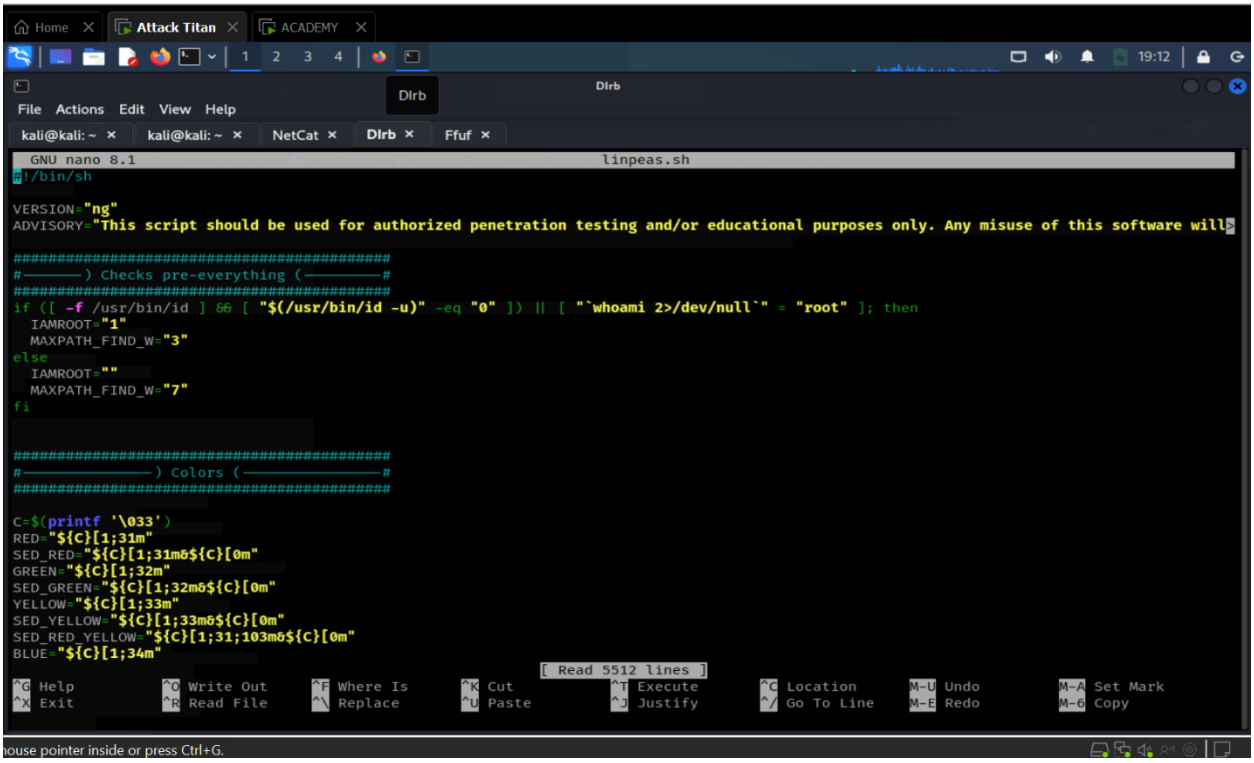
Now this is where we identify if we got root access or not with the “whoami” command. The response from this tells us we are anything but root, another dead end. At this point, we will need to escalate our privileges somehow. To do this, we can abuse the features already available in our terminal. Using “sudo -l” allows us to list binaries we can run with “sudo” to gain root. Looks like that isn’t available too.



This is where the **LinPEAS** (Linux Privilege Escalation Awesome Script) tool comes in very handy as it finds potential privilege escalation vulnerabilities on Linux/Unix systems.

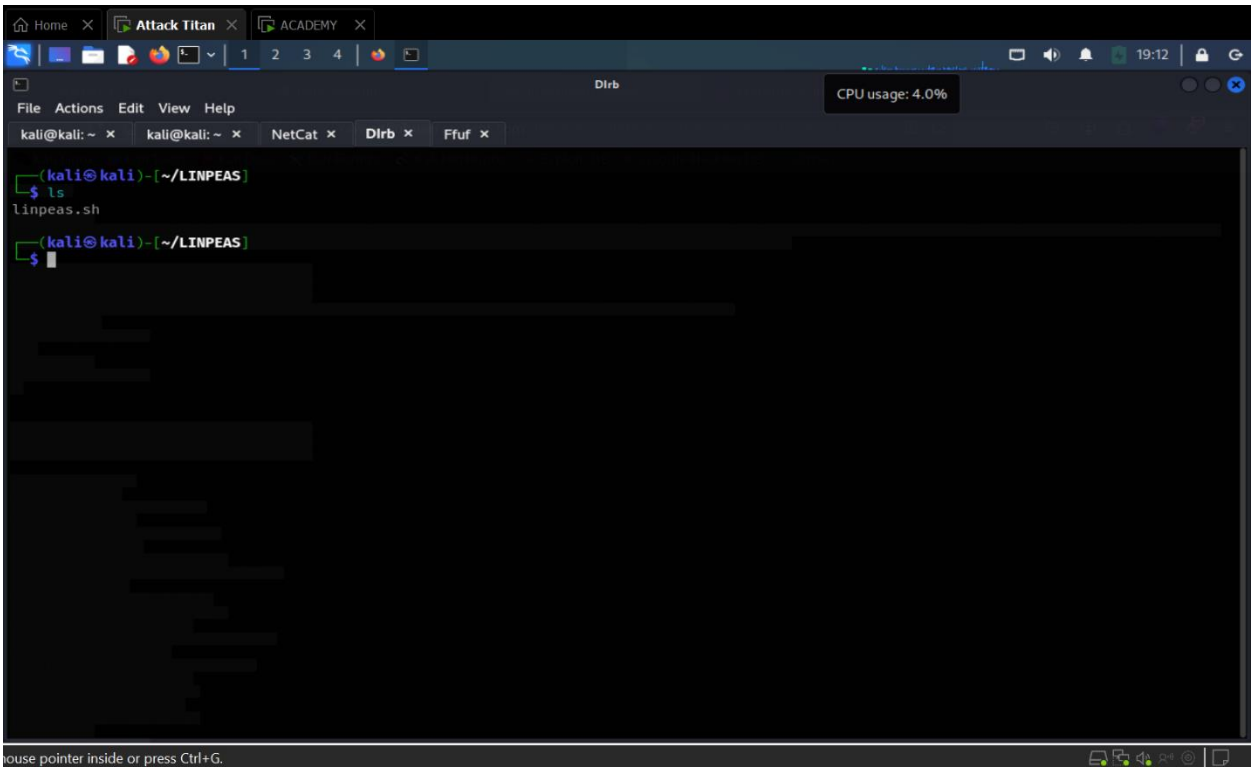


LinPEAS in raw text saved as “linpeas.sh”.

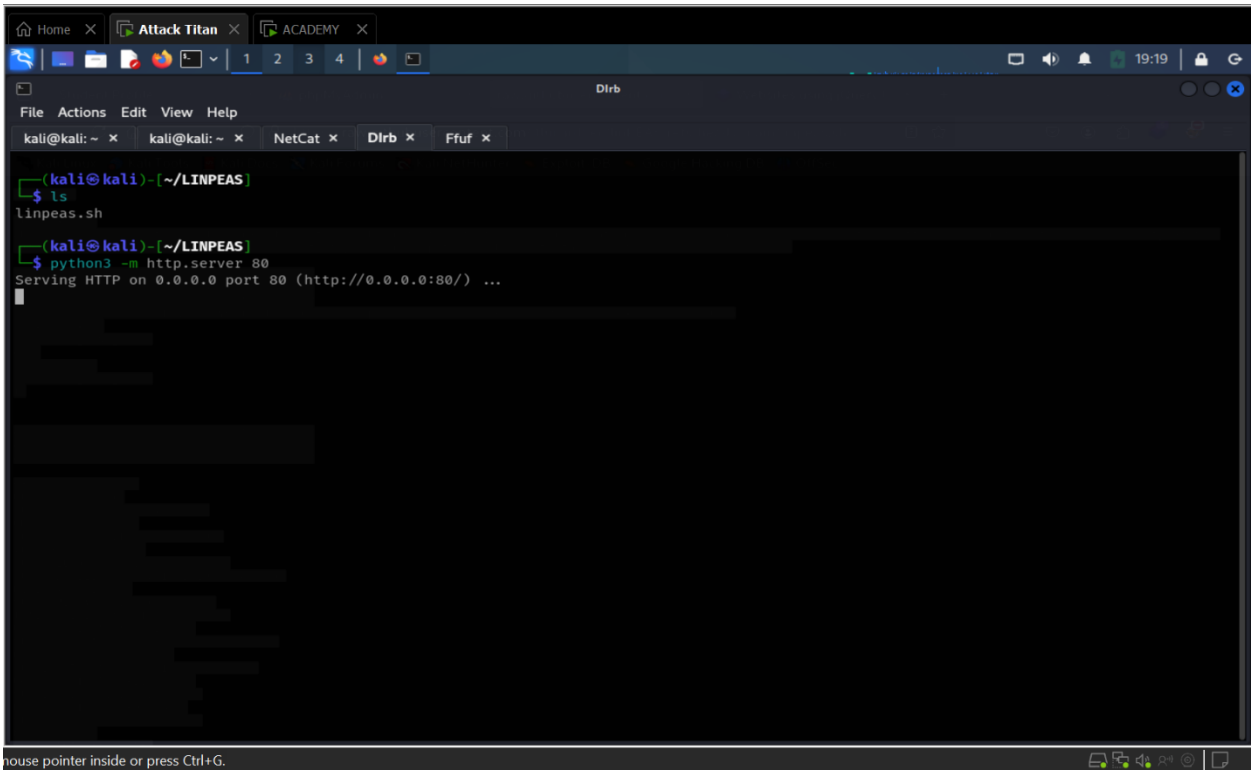


So far, we will not need to make any changes as we will run this directly on the target system.

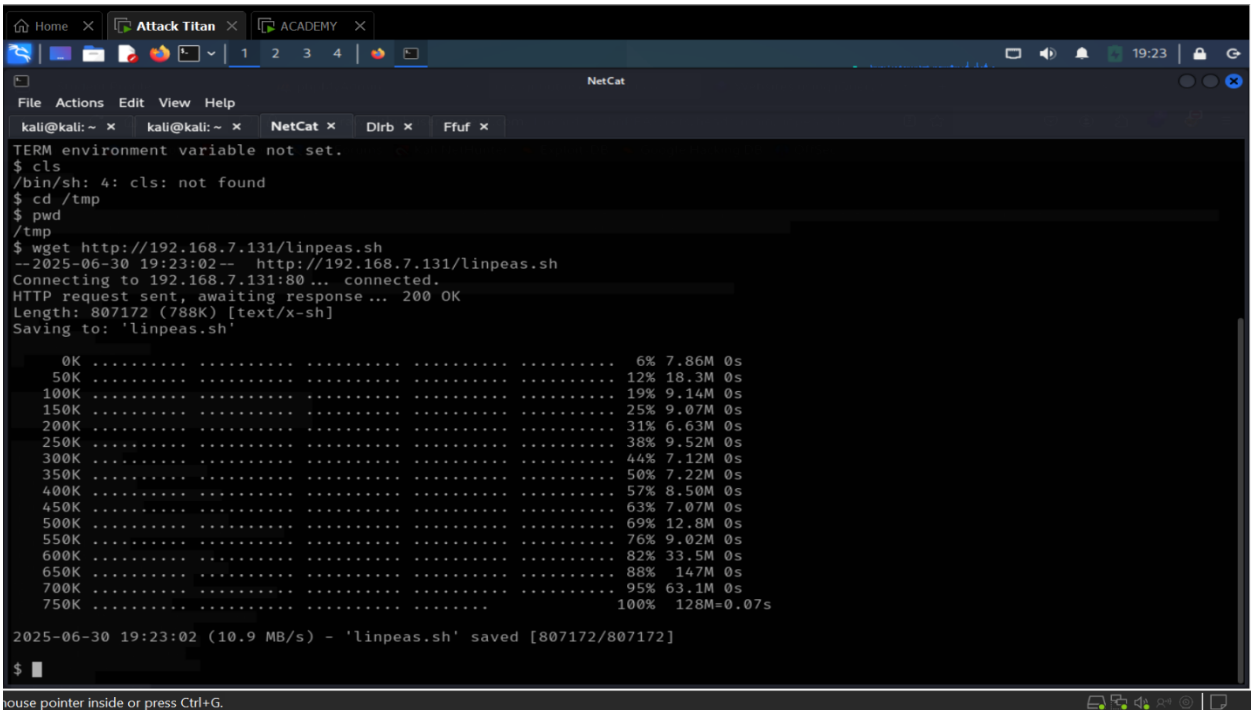
But before we do that, we first have to transfer the file to the target system.



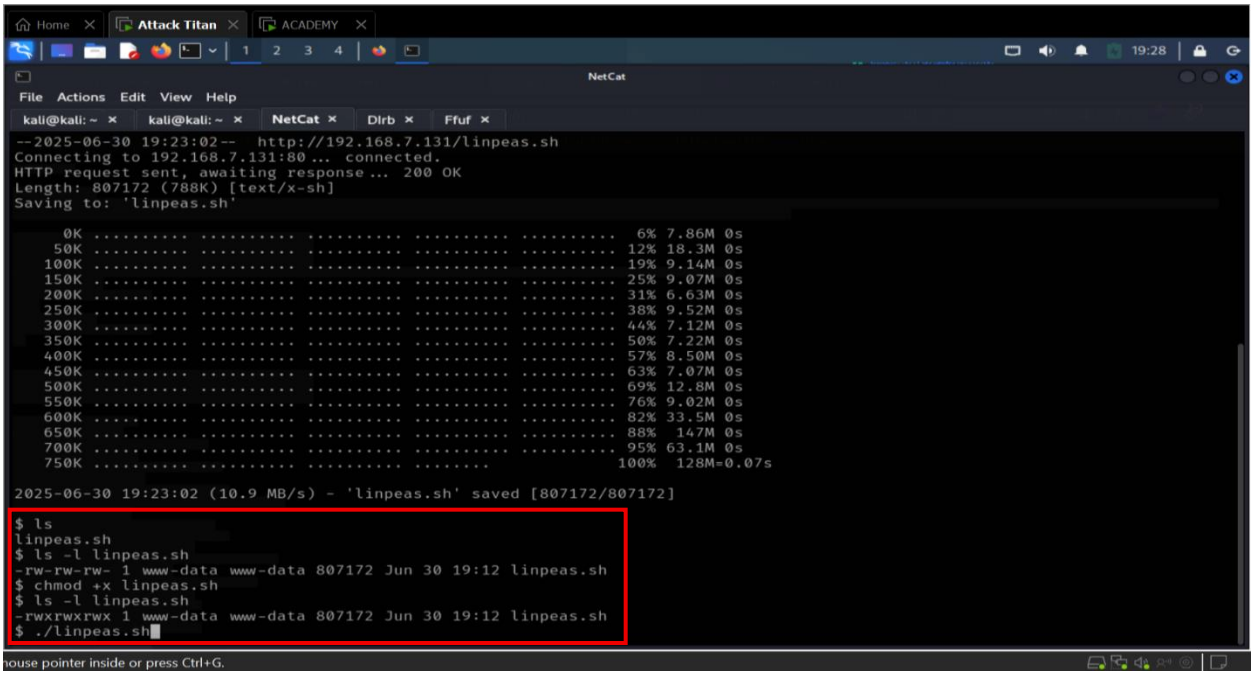
The “linpeas.sh” file was kept in the “LINPEAS” directory, this is where we setup a web server in order to pull out the file on the target system with ease.



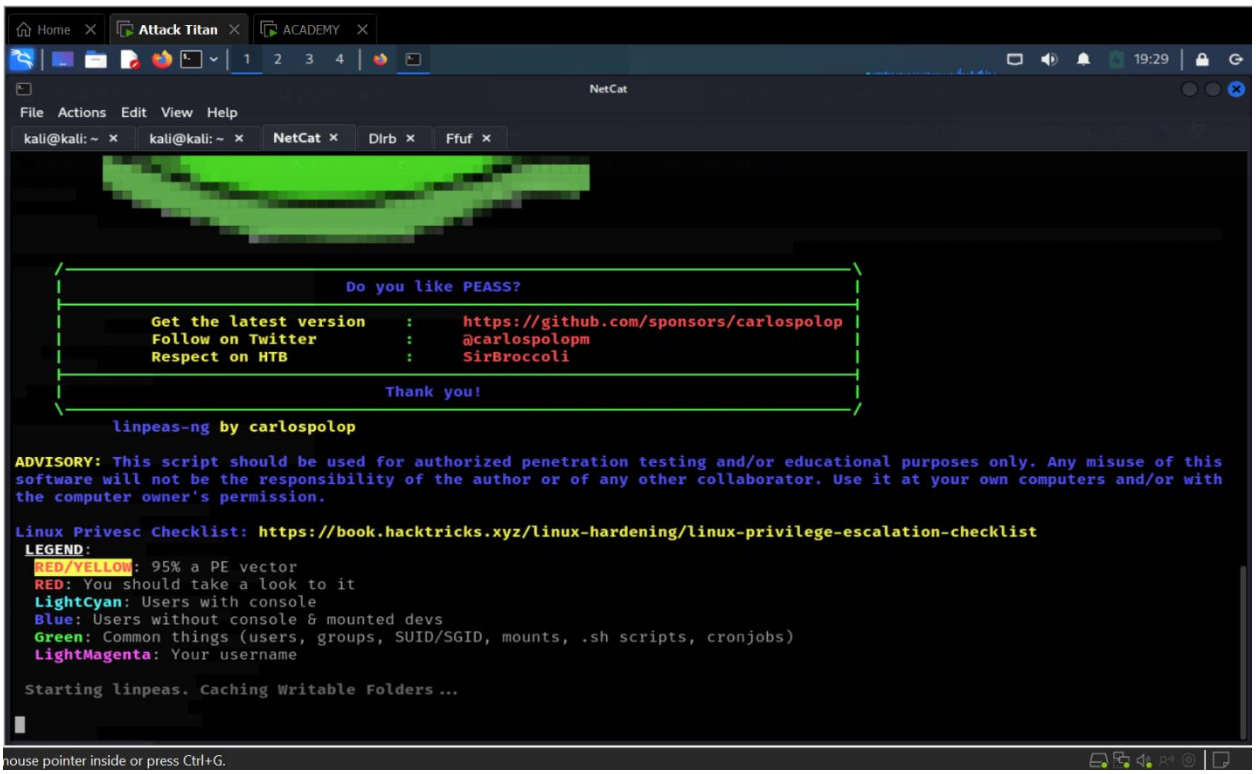
Our web server is active and running.



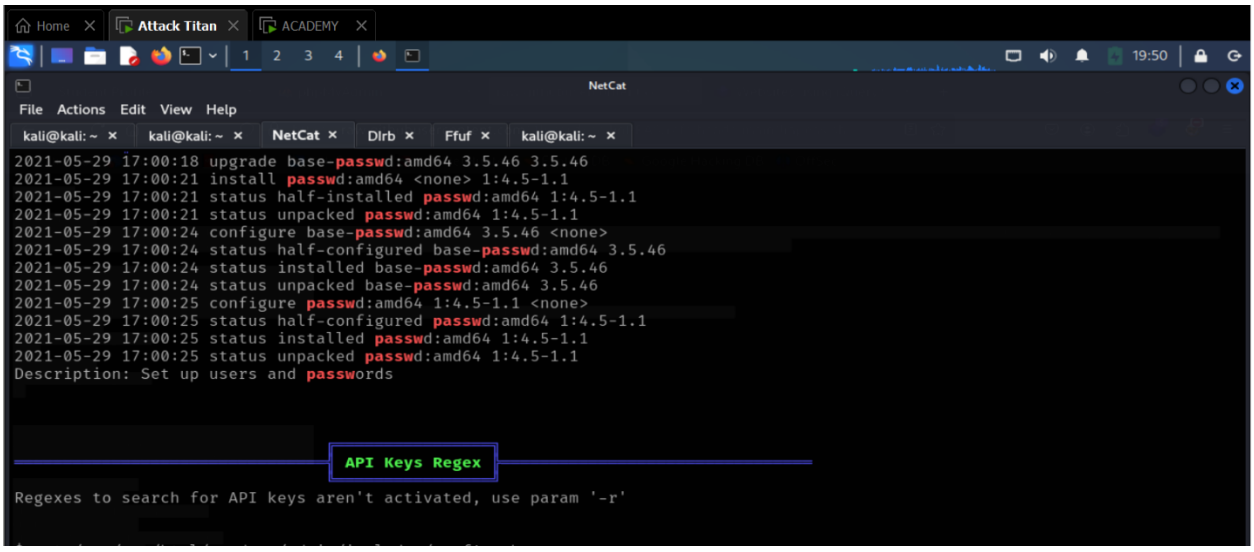
On the target system, we navigated to the /tmp directory in order to download the file. We use the “wget” command to achieve this. Though we finally transferred the file, the file would never run because most likely, it doesn’t have execute permission set to it. So, we first confirm this



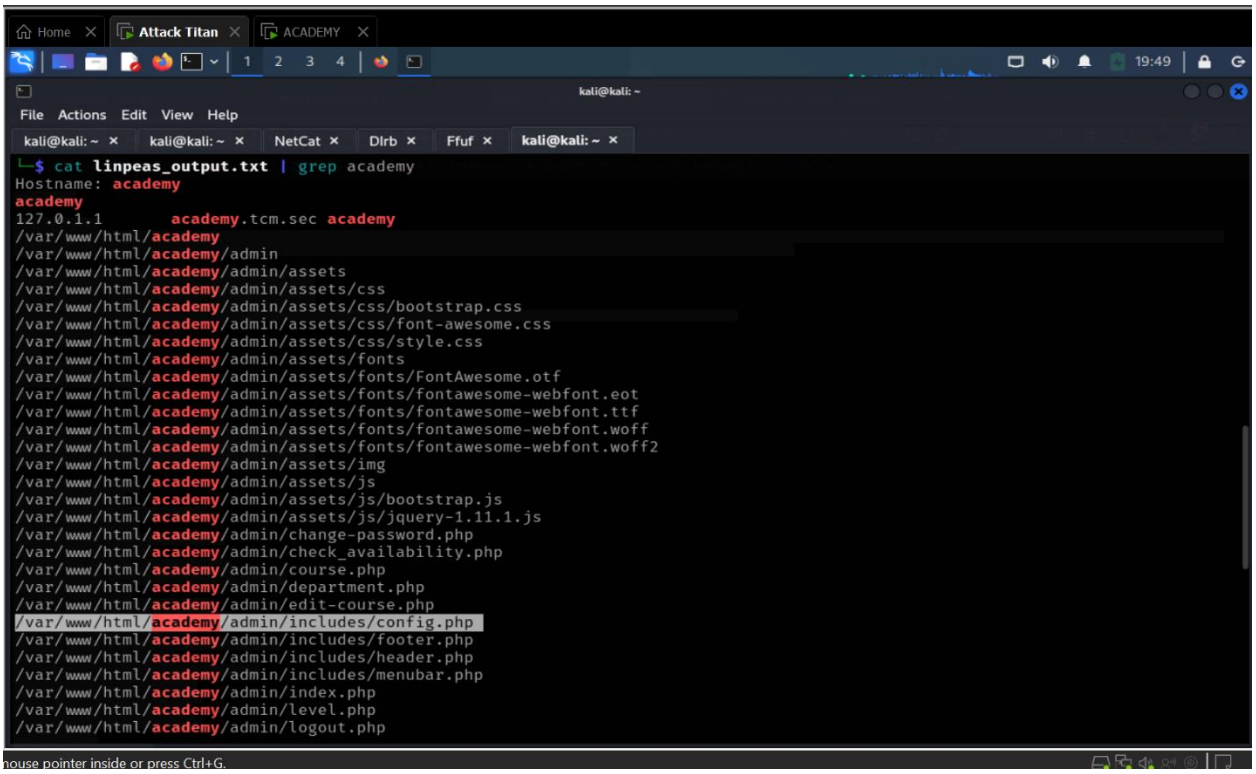
using “ls -l file_name” then we procede to add the execute permission with the “chmod” with “x” representing execute and “+” as the add function. Once this is all done, we can now execute the script by typing “./linpeas.sh”.



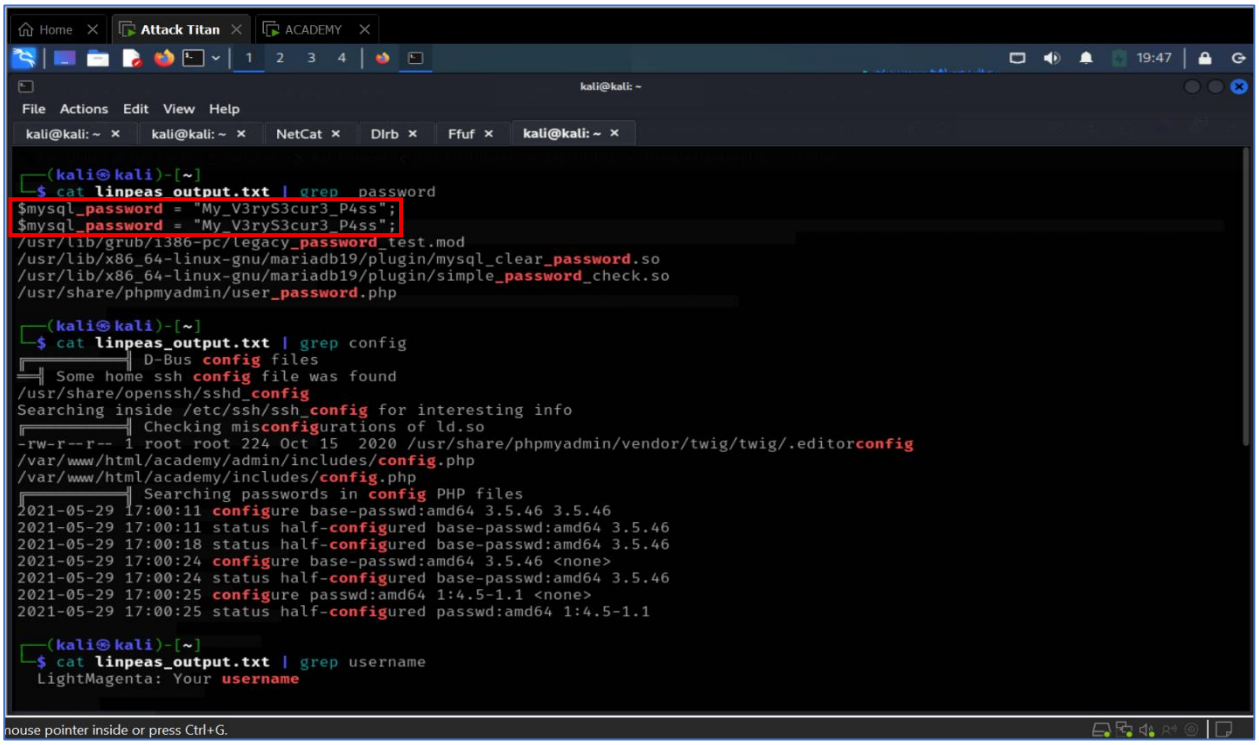
The LinPEAS script starts to run.



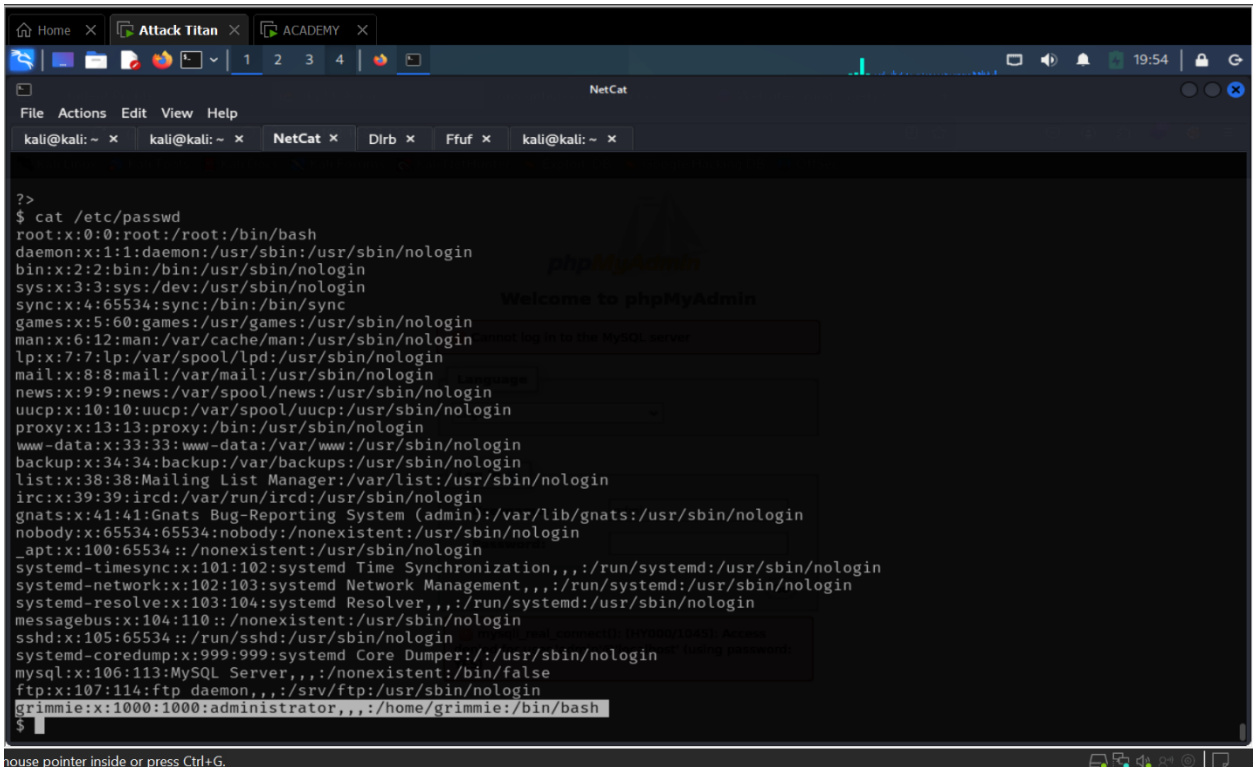
The execution of the script is now complete. Now running this powerful tool produced a ton of information we might have to take our time to analyze. But we can make this process a bit more convenient by copying the output into a file and utilize the “grep” command to fetch key words.



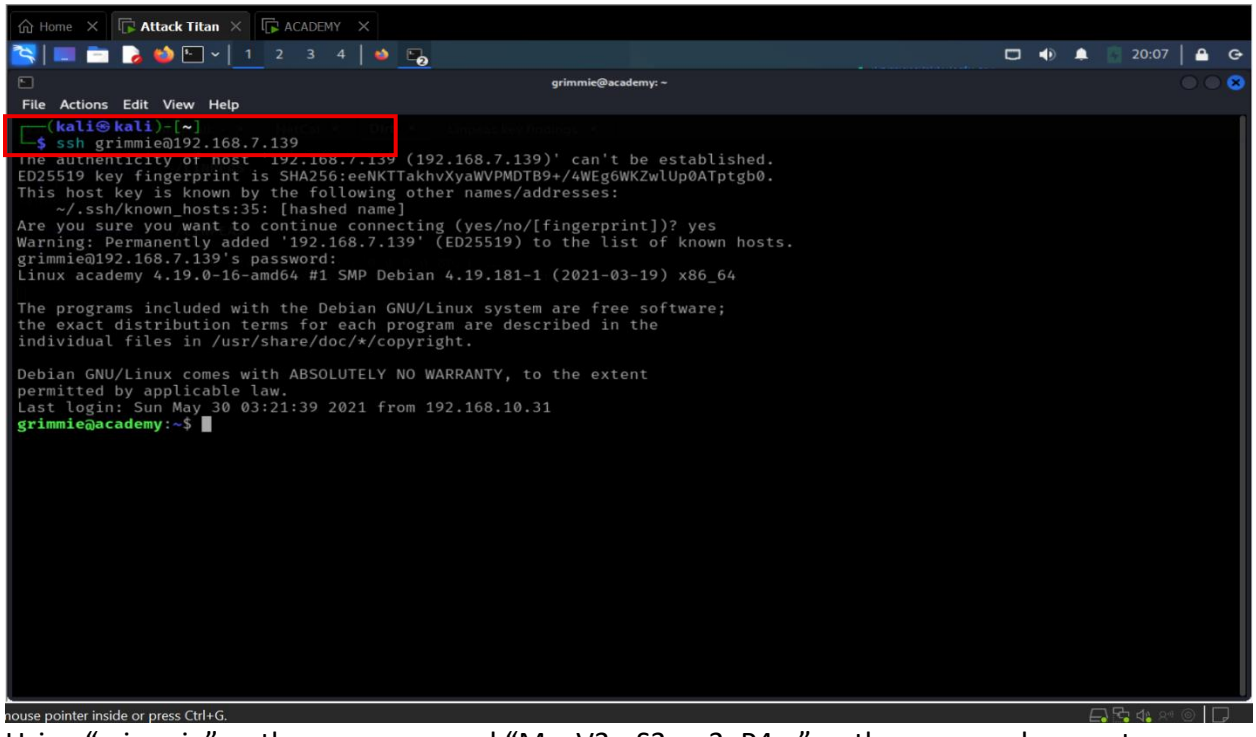
As demonstrated, we used keywords such as “password”, “config”, “academy”, “sql”, etc and anything that could provide some insights.



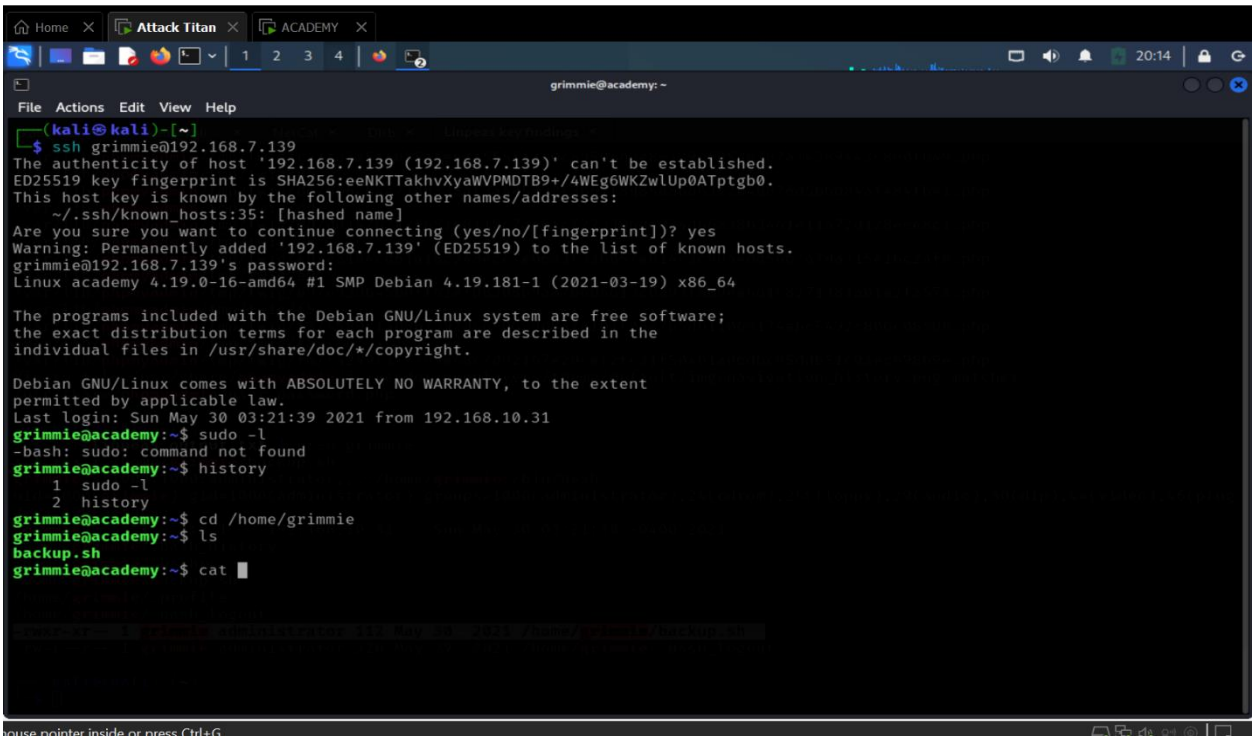
As a result, we got a password which could belong to a user with higher privileges. So, our next move is to know what users are on the system and probably use this to get access.



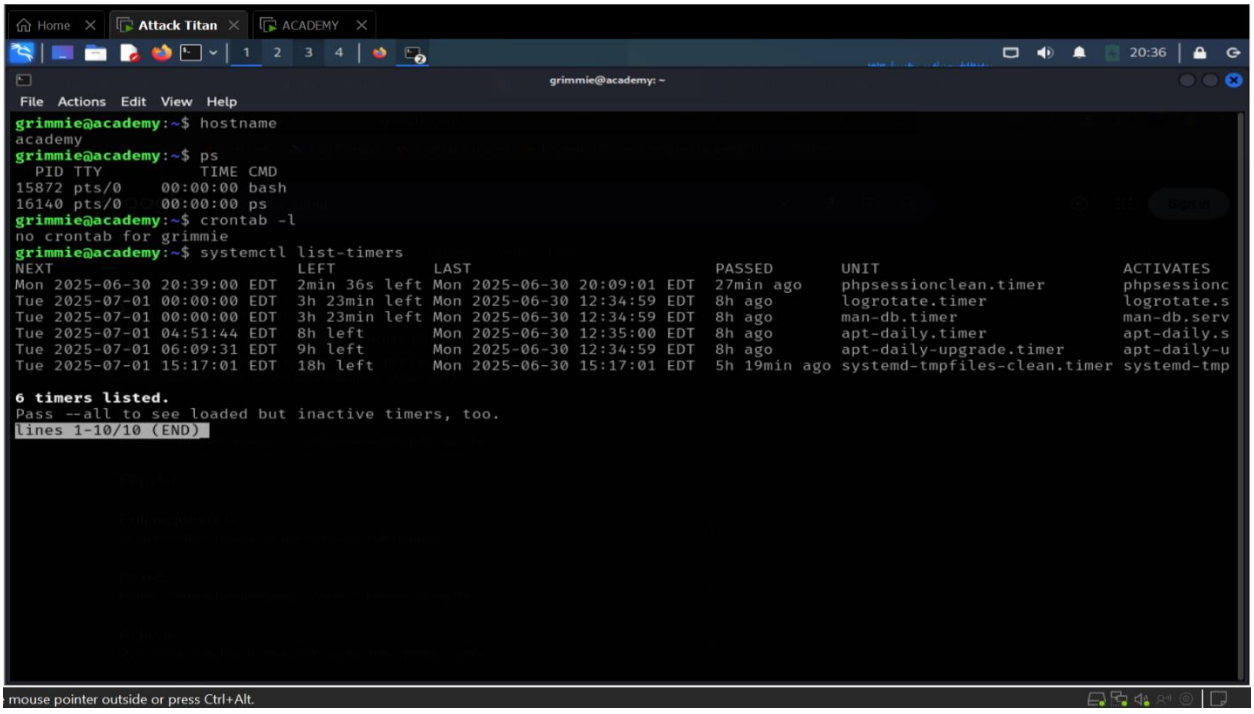
We run the “cat /etc/passwd” to list the users and look what we found, an administrator by the name “grimmie” (the name from the note.txt file). At this point, we have exhausted our chances with port 21 and port 80, now we are left with port 22 which is SSH (Secure SHell). It should be obvious at this point that we might want to log in through SSH on our attack machine.



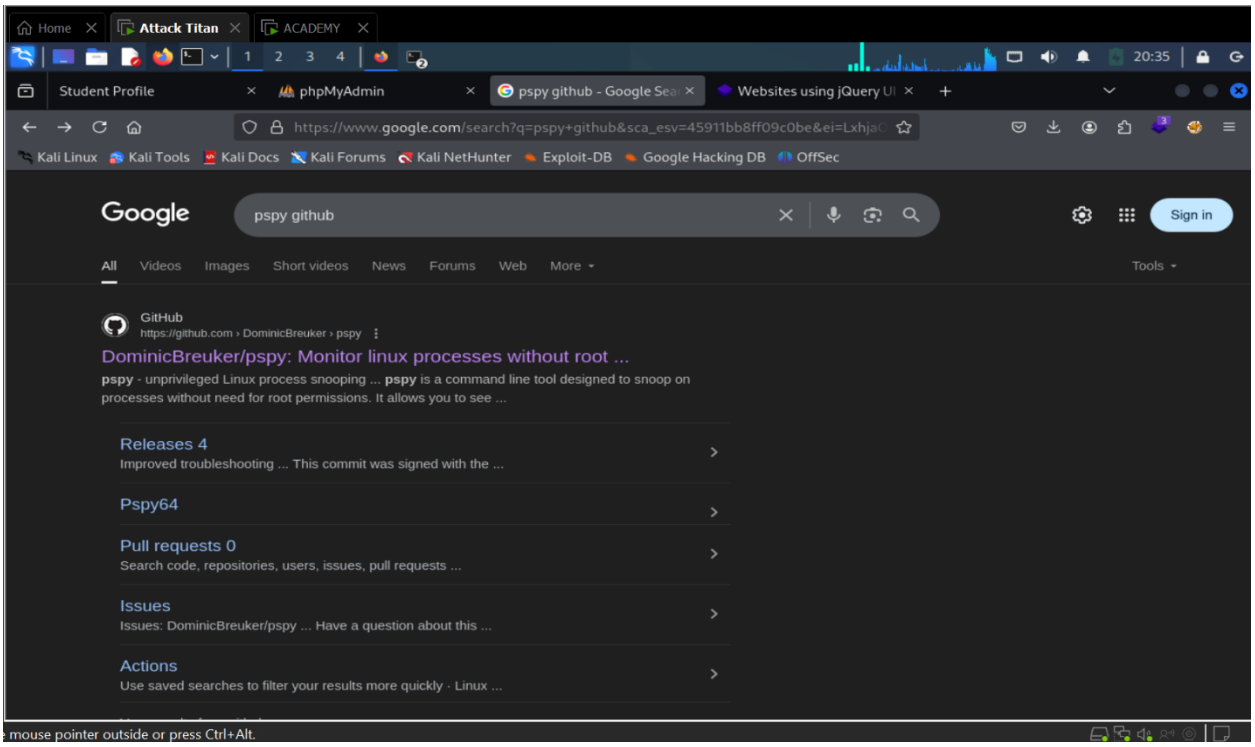
Using “grimmie” as the username and “My_V3ryS3cur3_P4ss” as the password, we got access.



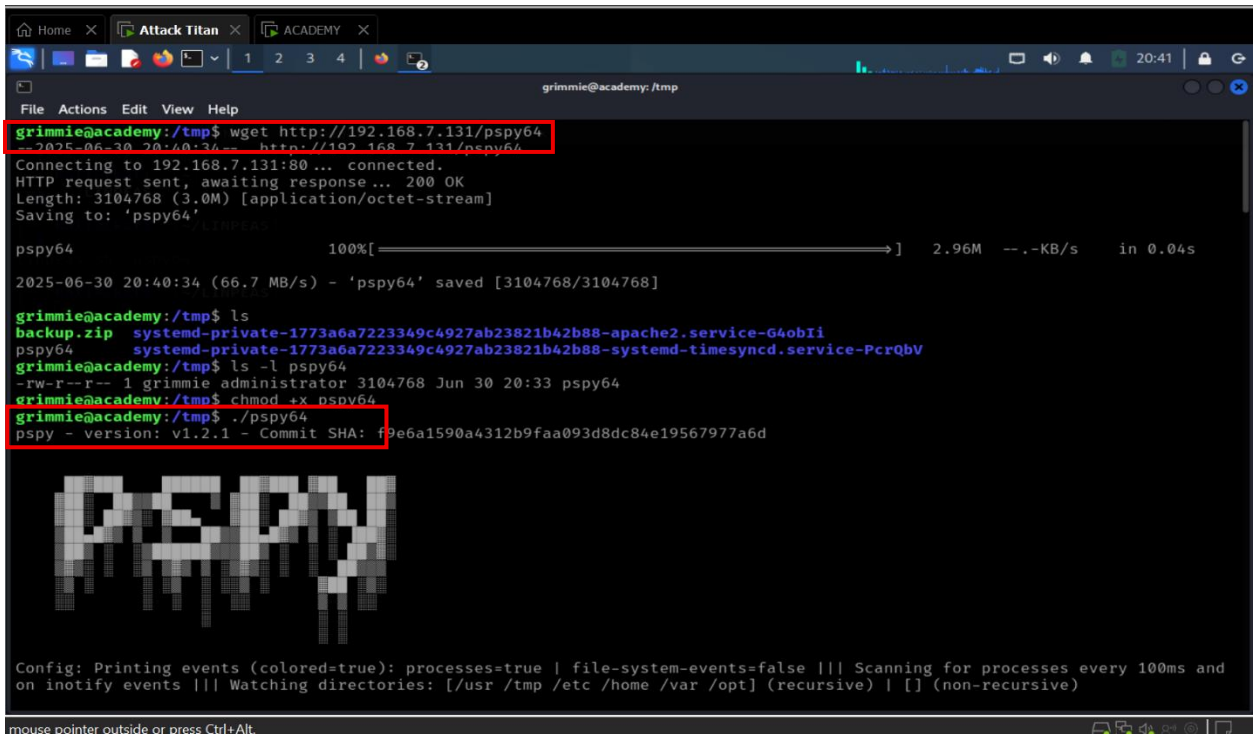
Once again, there isn't much here and so we will have to make use of what I have. The "backup.sh" appears to be a process that runs at a certain time interval in the background.



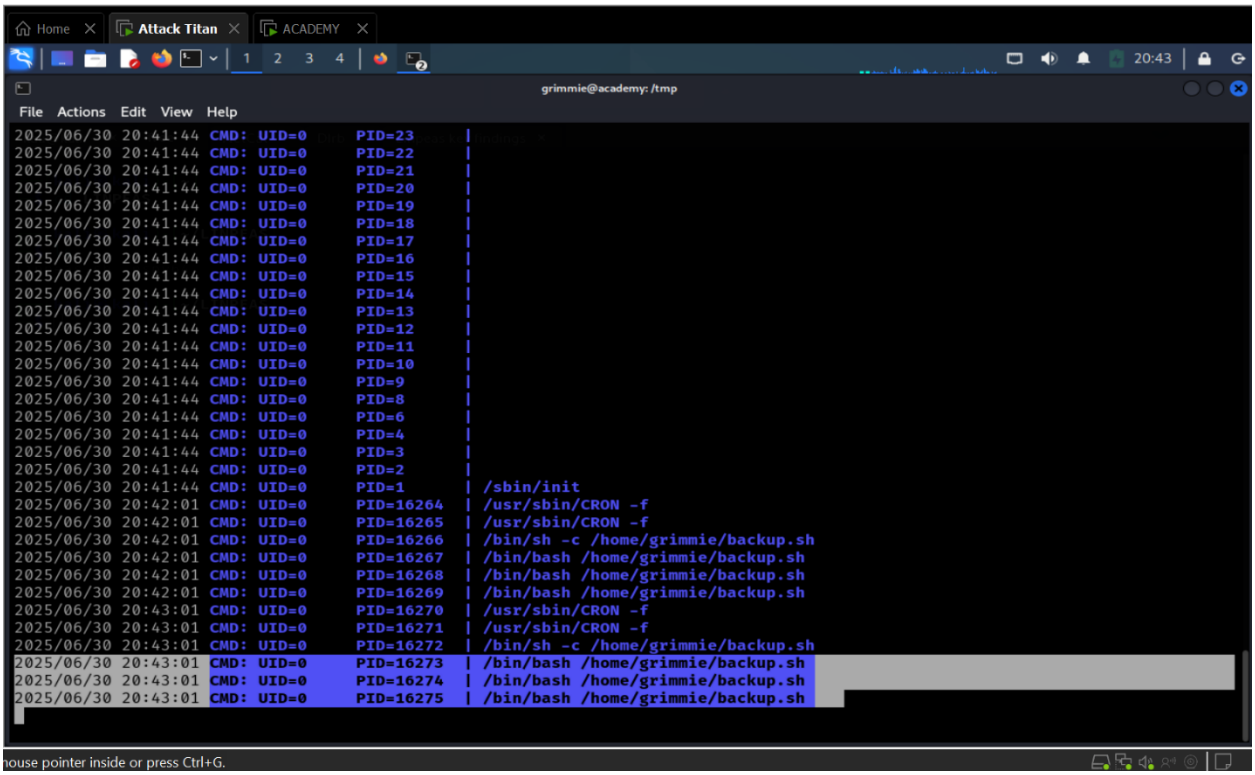
But when we look for this process, it doesn't seem to appear on the process list. This is probably occurring because we don't have root privileges to do so. A tool that can help us achieve this is the "**PSPY**" tool.



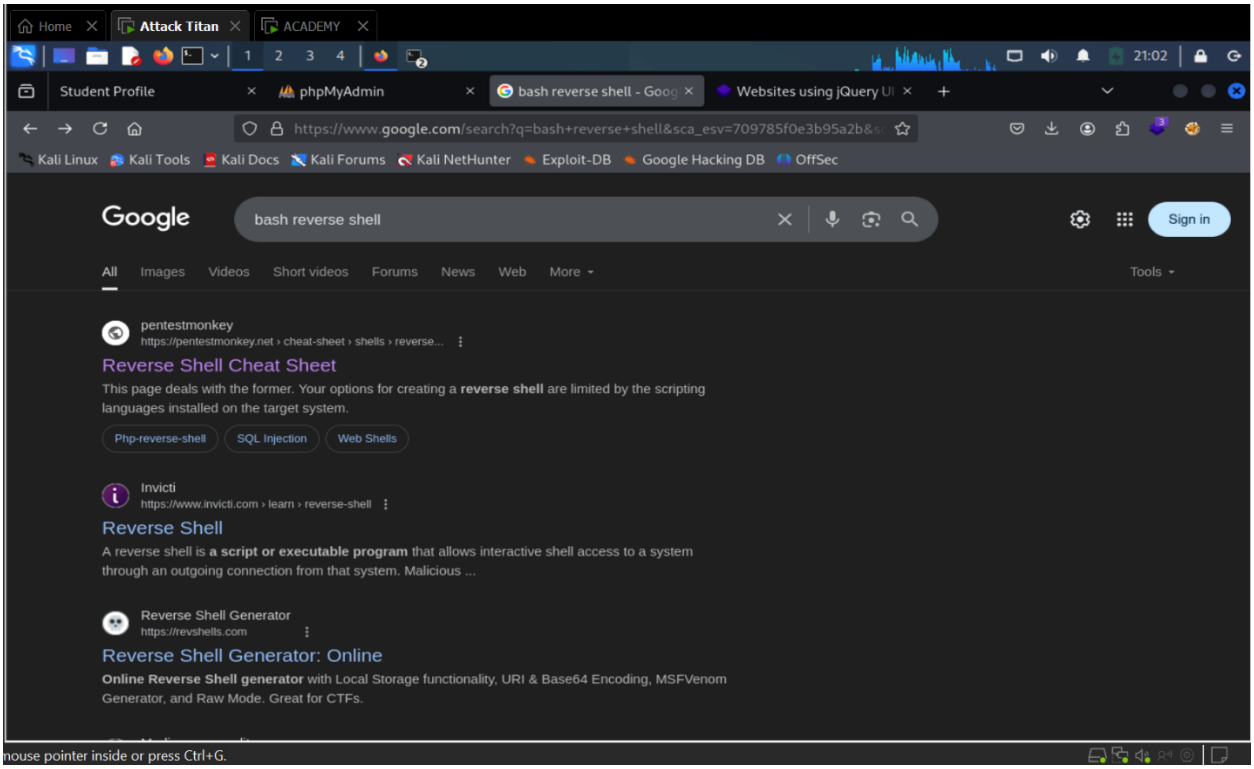
This tool allows us to view all processes without root privileges. We ensured the tool was downloaded in the LinPEAS directory where our web server was setup and still running.



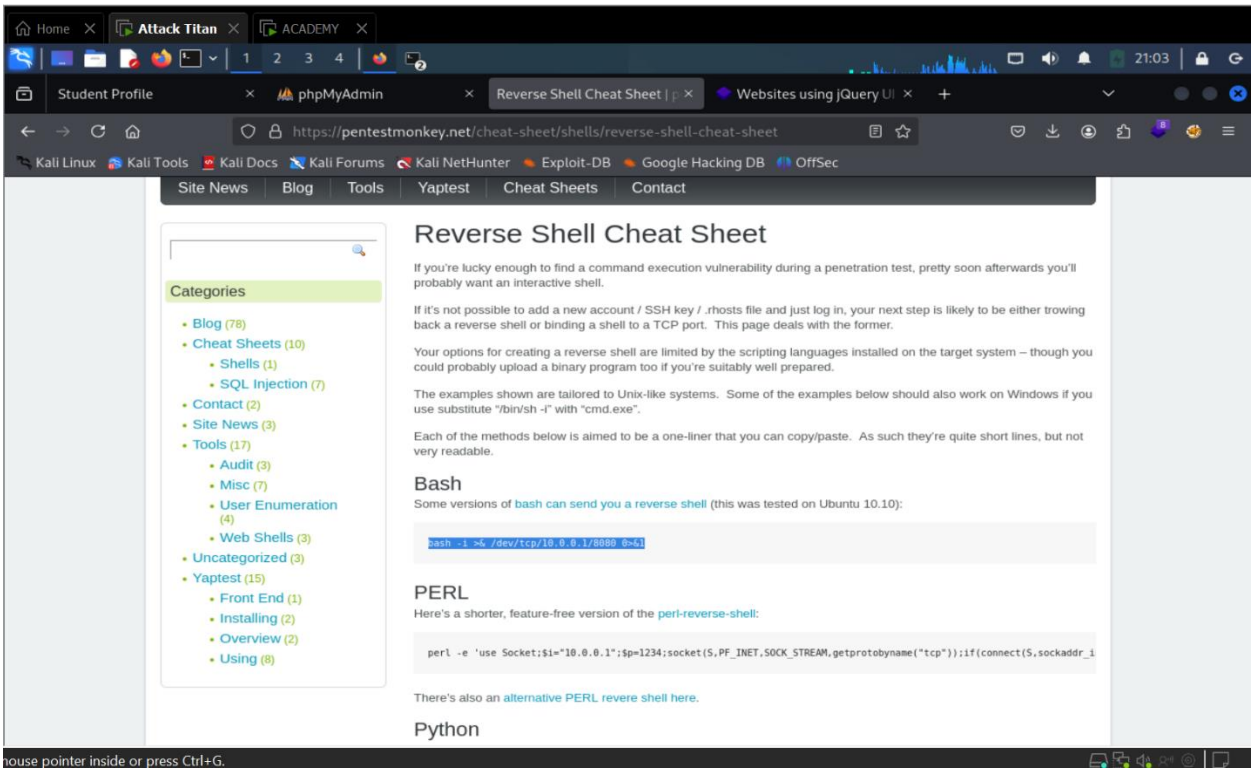
With that said, we simply locate to our preferred directory (/tmp) and download the file from the server with the “wget” command. Then we make the script executable with the “chmod” command and run it.



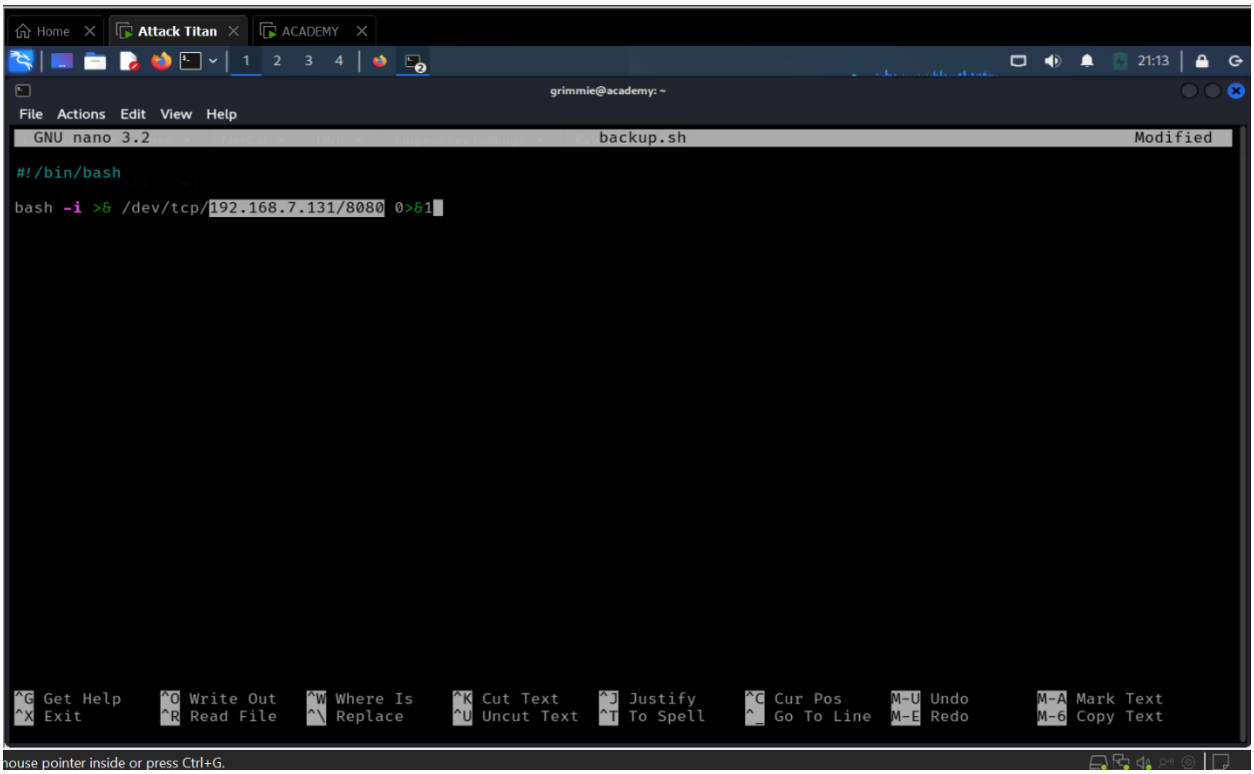
Finally, we were able to expose the hidden process as highlighted. We can once again abuse the scripting language used by the process (bash) to pop a shell.



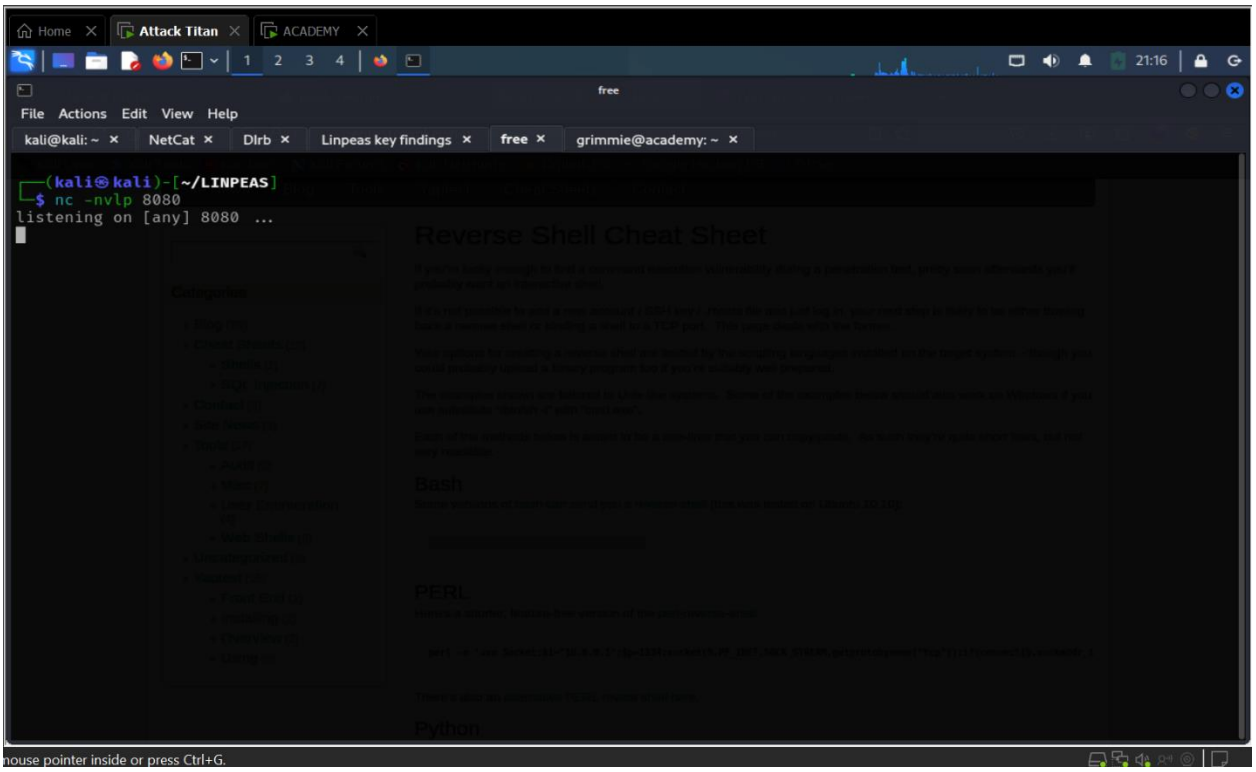
Just like we did with the PHP, we can also spawn a shell in bash.



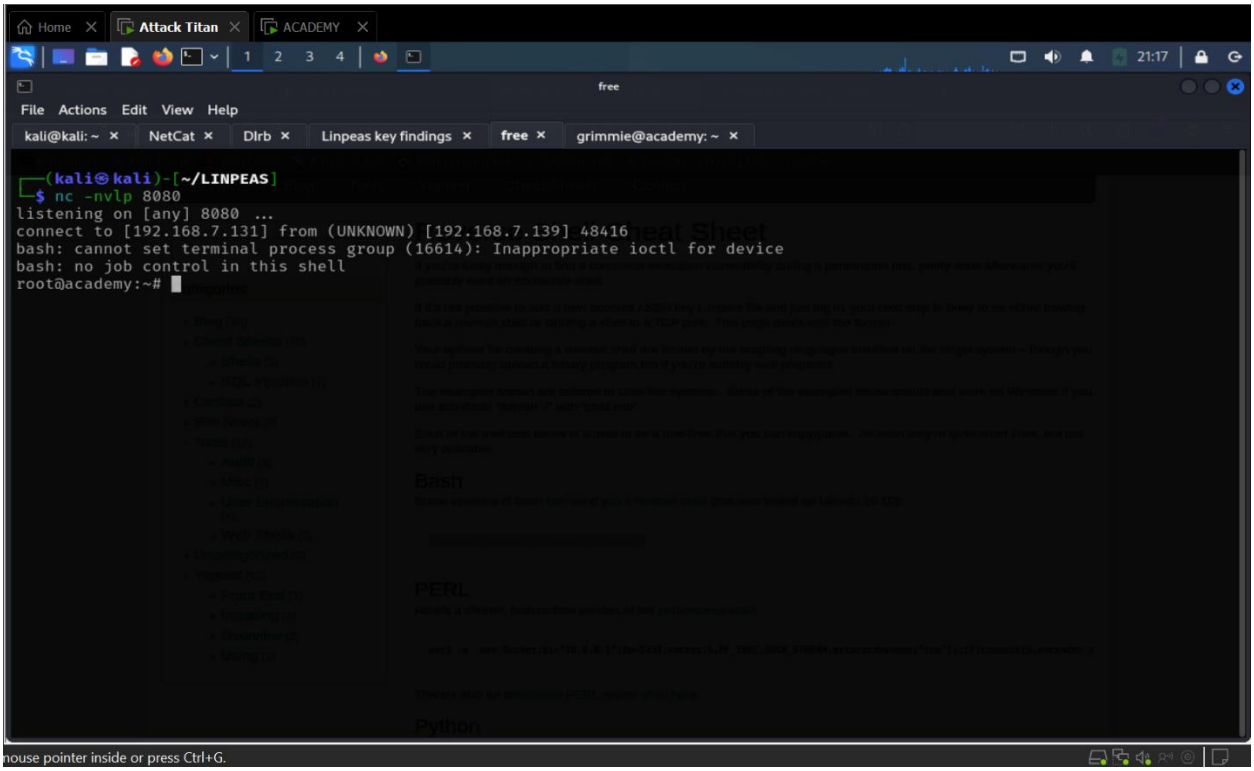
This is a one-line code we can copy and paste into the process file



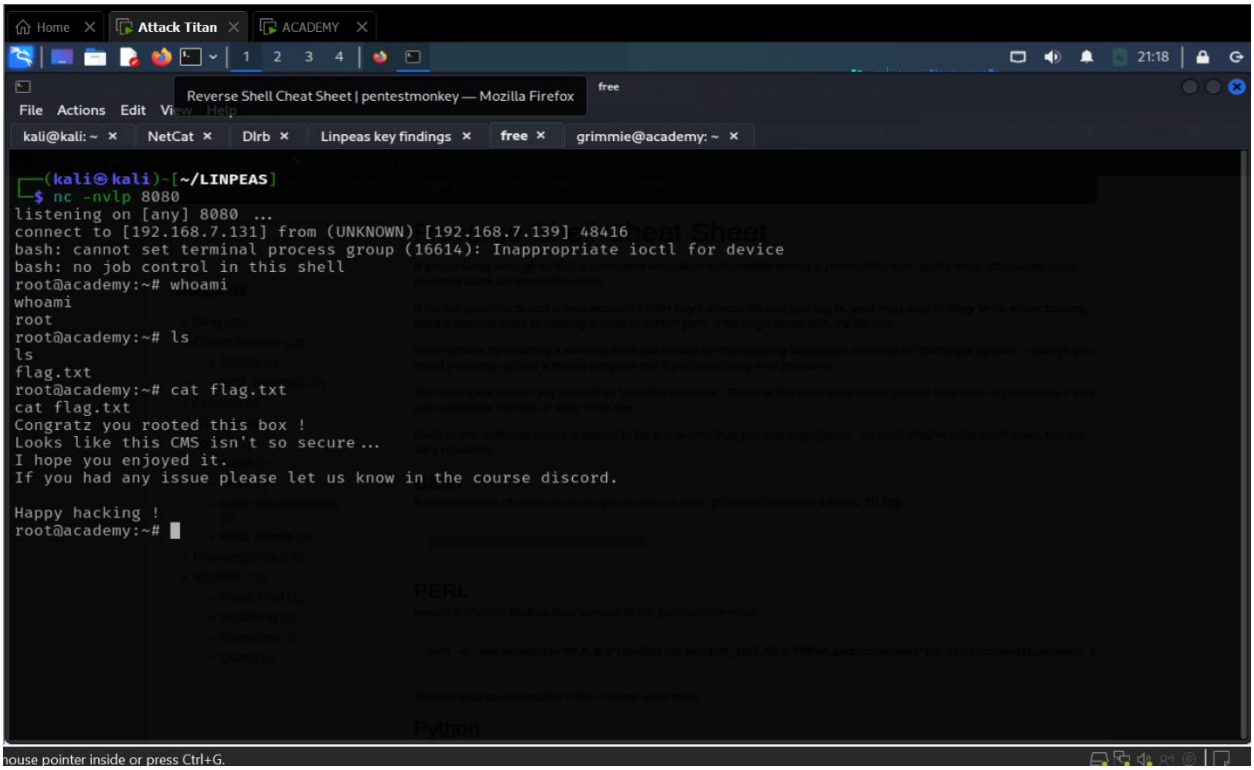
The line of code requires that we insert our attack machine's IP address along with a port number and save it.



Now we open a listener with **Netcat** and listen through the set port number (8080)



At long last, we finally got root access to the machine.


Home
Attack Titan
ACADEMY
1 2 3 4
21:18
Reverse Shell Cheat Sheet | pentestmonkey — Mozilla Firefox
free
File Actions Edit View Help
kali@kali: ~
NetCat
Dirb
Linpeas key findings
free
grimmie@academy: ~
┌──(kali㉿kali)-[~/LINPEAS]
└─$ nc -nvlp 8080
listening on [any] 8080 ...
connect to [192.168.7.131] from (UNKNOWN) [192.168.7.139] 48416
bash: cannot set terminal process group (16614): Inappropriate ioctl for device
bash: no job control in this shell
root@academy:~# whoami
whoami
root
root@academy:~# ls
ls
flag.txt
root@academy:~# cat flag.txt
cat flag.txt
Congratz you rooted this box !
Looks like this CMS isn't so secure...
I hope you enjoyed it.
If you had any issue please let us know in the course discord.
Happy hacking !
root@academy:~#
house pointer inside or press Ctrl+G.