Write-up for FriendZone

Hosted @ HackTheBox

Creator: Askar

Writeup: NO53LF

First Off, as always I'd like to thank The Folks at HackTheBox and the maker of this box, Askar without the people hosting and building these labs, we wouldn't be able to have a safe legal enviroment to test/build our skills.

This box was bit more challenging than I expected for a 20 point box, particularly the user flag but lets jump in.

First let's fire up kali in VirtualBox.

Once that's running we'll start off as always with an nmap scan.

(The target box can be found at 10.10.10.123.)

```
root@kali:~# nmap -sC -sV -T5 10.10.10.123
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-28 20:04 MDT
Nmap scan report for 10.10.10.123
Host is up (0.15s latency).
Not shown: 993 closed ports
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  https?
445/tcp open  netbios-ssn Samba smbd 4.7.6-Ubuntu (workgroup: WORKGROUP)
Service Info: Host: FRIENDZONE; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: -59m59s, deviation: 1h43m54s, median: 0s
|_nbstat: NetBIOS name: FRIENDZONE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| 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: 2019-06-29T05:05:01+03:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2019-06-28 20:05:00
|_  start_date: N/A

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
```

A slower scan(-T4) gave some more info on port 443:

```
File  Edit  View  Terminal  Tabs  Help
443/tcp open  ssl/http    Apache httpd 2.4.29
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: FriendZone escape software
| 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
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|   http/1.1
|   http/1.1
```

We'll add friendzone.red to our host file:

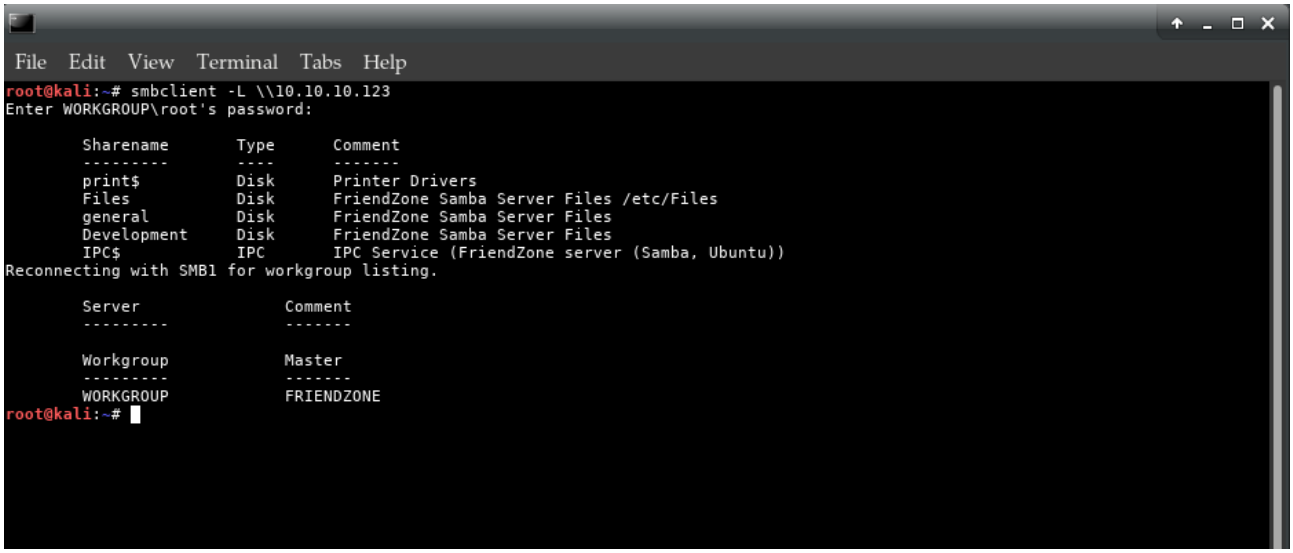
```
File  Edit  Search  Options  Help
1 127.0.0.1       localhost
2 127.0.1.1       kali
3 10.10.10.123    friendzone.red
```

Our scan returned some interesting results with ports:

- 21 FTP
- 22 SSH
- 53 Self hosted DNS
- 80 HTTP
- 139 & 445 (SMB related)
- 443 HTTPS

My first thought was to check FTP for anonymous login to see if any clues may kicking around but that wasn't the case and knowing it's hackthebox I opted not to attempt brute-force attack.

Moving on I decided to take a look at SMB service by enumerating shares listed using SMBClient.



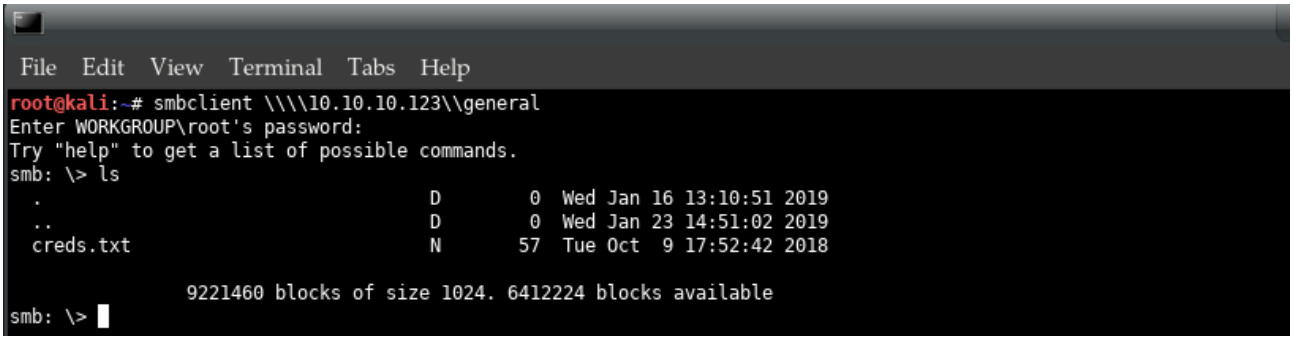
- Files shows us a path *"/etc/Files"*

- general
- Development (*looks interesting*)

I'll use SMBMap to checkout some permissions.

Looks like we only have access to 2 shares, general & Development but the later has read and write permissions which is quite interesting!

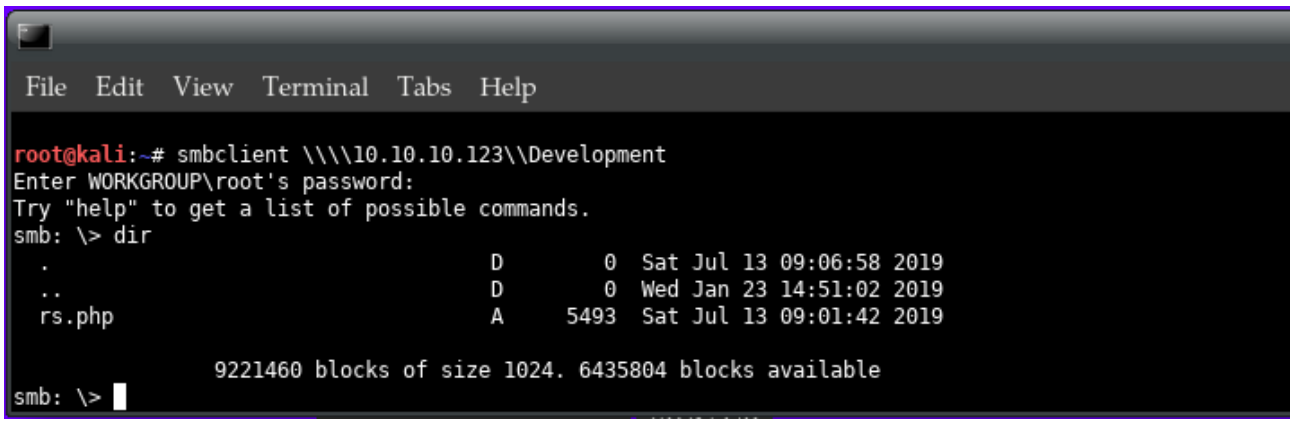
So going back to SMBClient we'll check these out:



As you can see there is a file named "*creds.txt*" which contained the string "*creds for the admin THING: "* and the credentials *" admin:WORKWORKHhallelujah@# "*

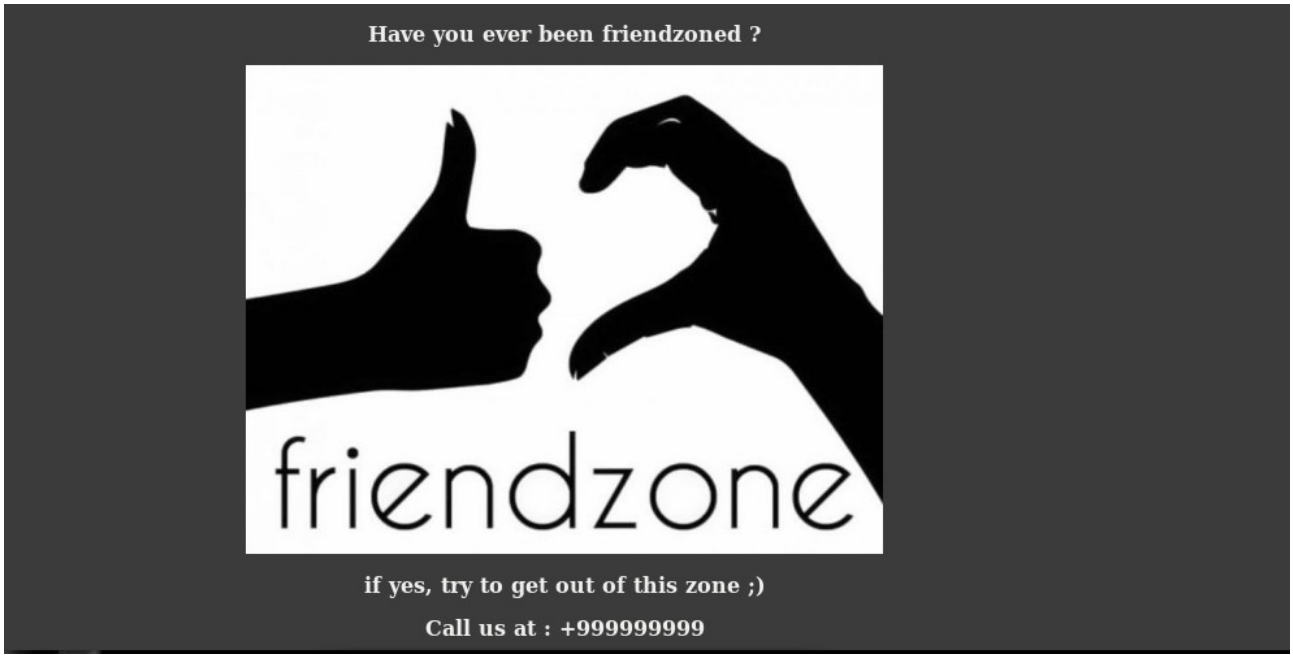
These will most likely come in handy later!
(I did test them on FTP & SSH but did not work)

So moving to the Development share I see a php file which is interesting. (probably left there by someone else)



I uploaded a file test.txt to confirm my ability to write which was successfull and made note of that as it could be very useful later on.

Moving on to the web I browsed to 10.10.10.123:80 where I got this landing page:



And this landing page for friendzone.red with ssl:

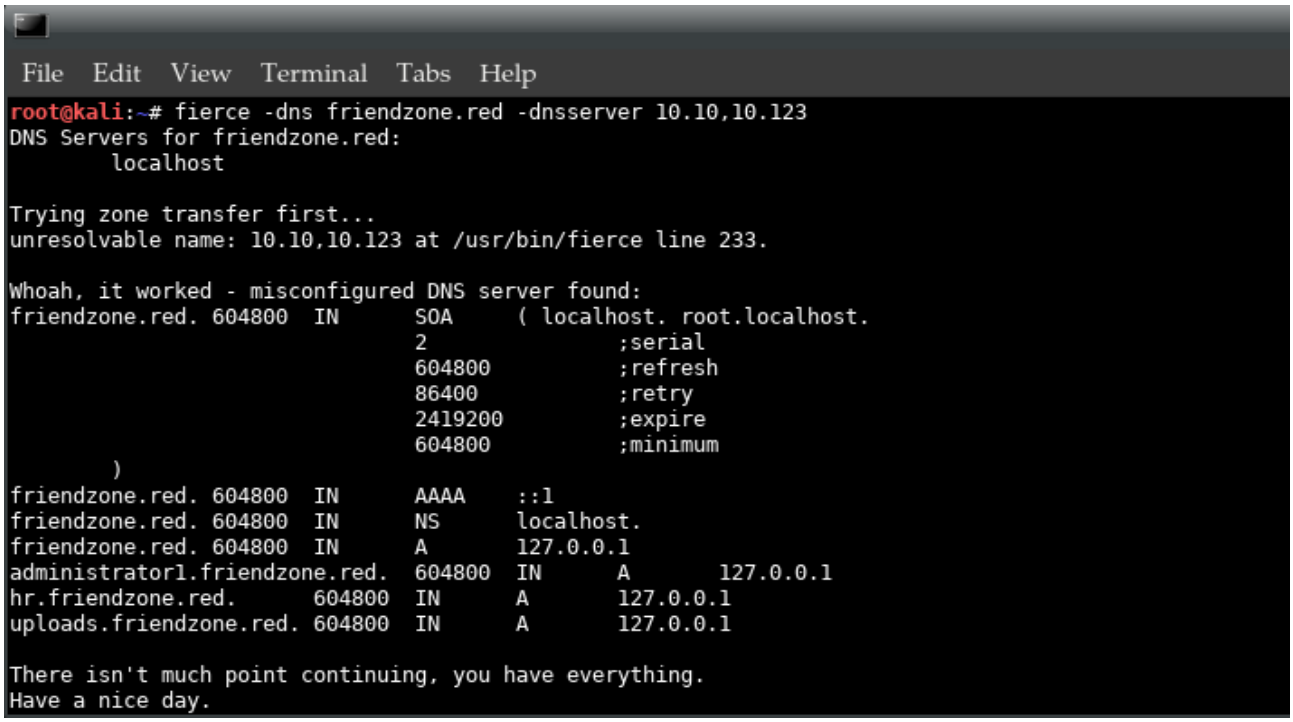
The landing page on port 80 had another domain listed for a contact email listed at info@ friendzoneportal.red but that was a dead end.

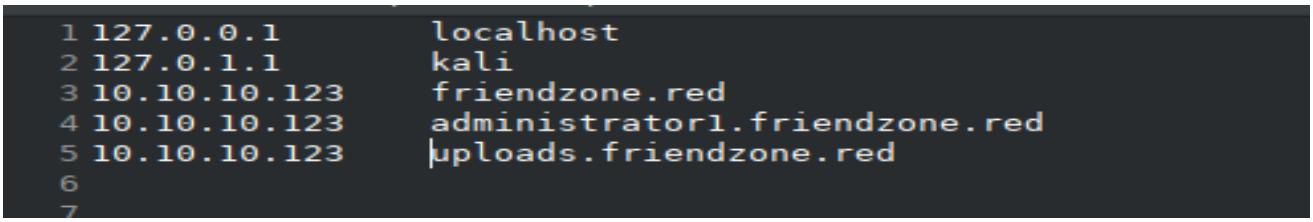
I also tried dirbust to brute any other files or directories that might be hidden that may contain some information or a login page for the credentials I found earlier but those were unsuccessful as well

Looking back at nmap results the light bulb went off.... Port 53 running dns,box name “Friend **ZONE**”...

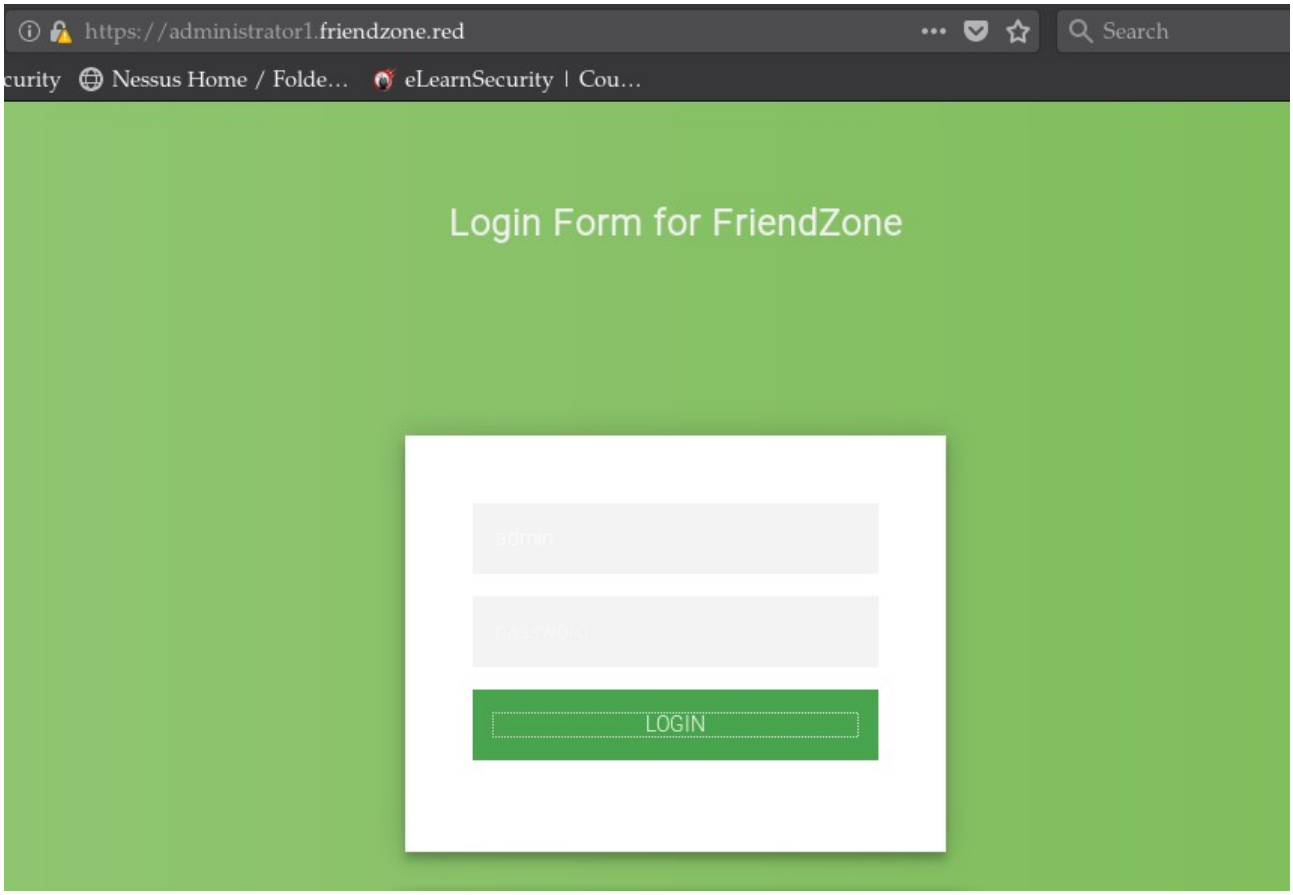
Zone Transfer!

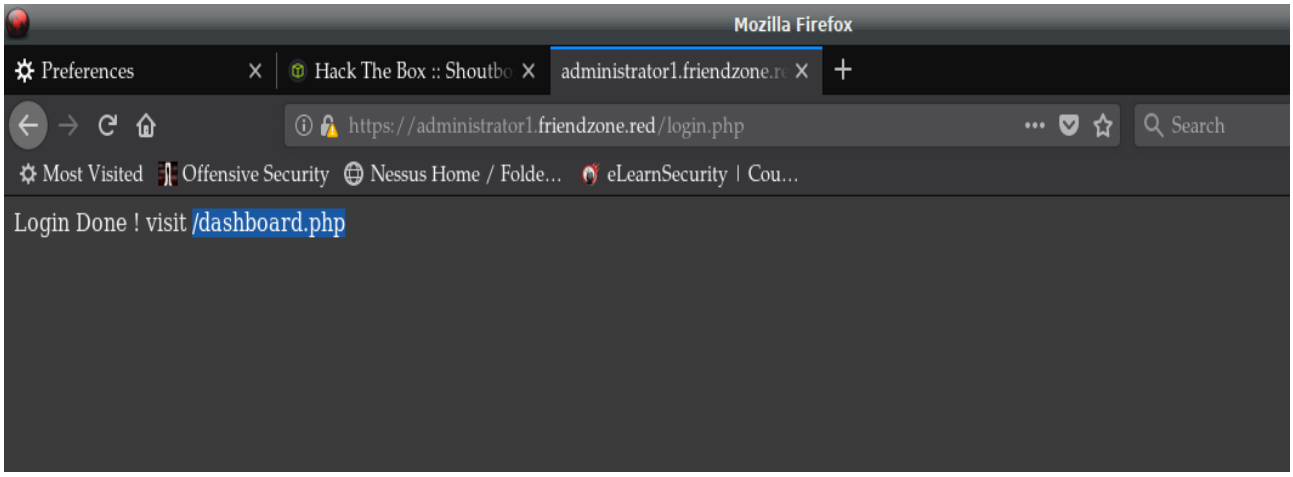
Using fierce I was able to find some subdomains and edit my hostsfile accordingly.





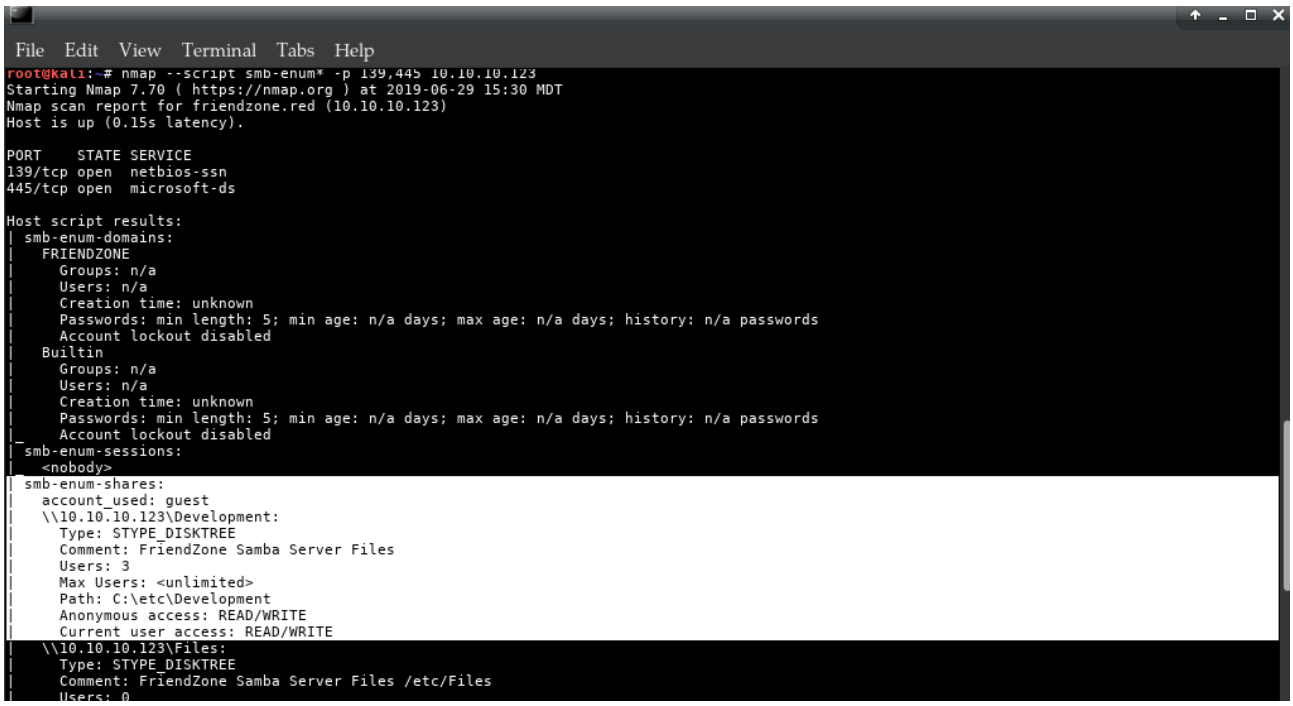
Going over to the administrator1 sub domain I was greated with a login page and able to authenticate using the credentials from the creds.txt file found earlier and follow the instructions to the dashboard.



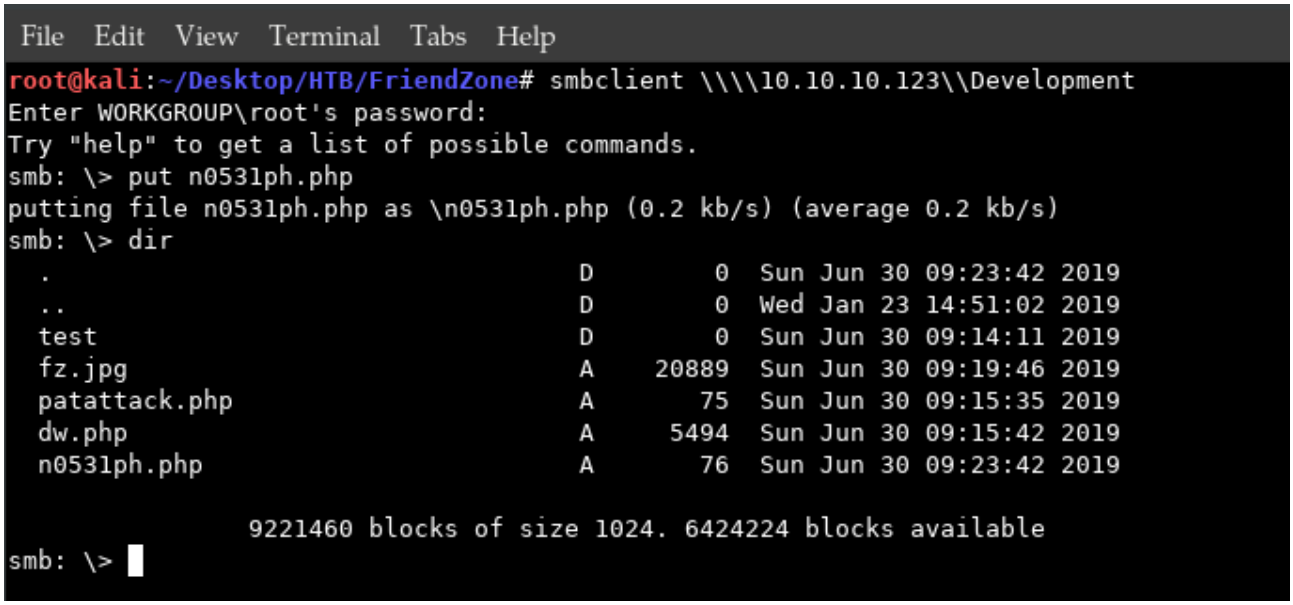


The /dashboard.php page gave us some big hints with regards to how to view your uplaoded files!

I tinkered around for a while trying to upload malicious files and fuzzing around for LFI/RCE when I remembered the writable SMB share “Development” that also already contained a php file in it (unintended big hint) but I wasn’t sure of the path to which files were being written so a bit more recon on that service is required. Using nmap’s smb-enum script I find it’s writing to “/etc/Development”

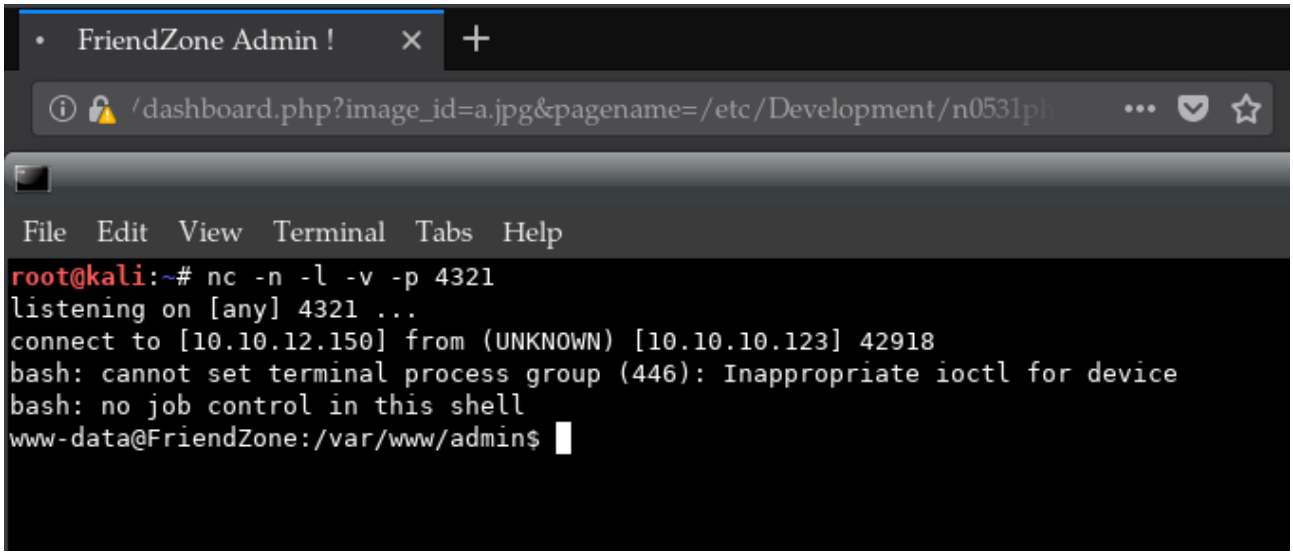


Armed with this new found info I fired up smbclient again and uploaded a very simple php reverse shell.



Going back to the webpage and fuzzing a bit more I was able to get a reverse shell through LFI by navigating to:

“dashboard.php?image_id=a.jpg&pagename=/etc/Development/n0531ph.php”



Navigating to the /home/friend directory I grabbed the user.txt flag:

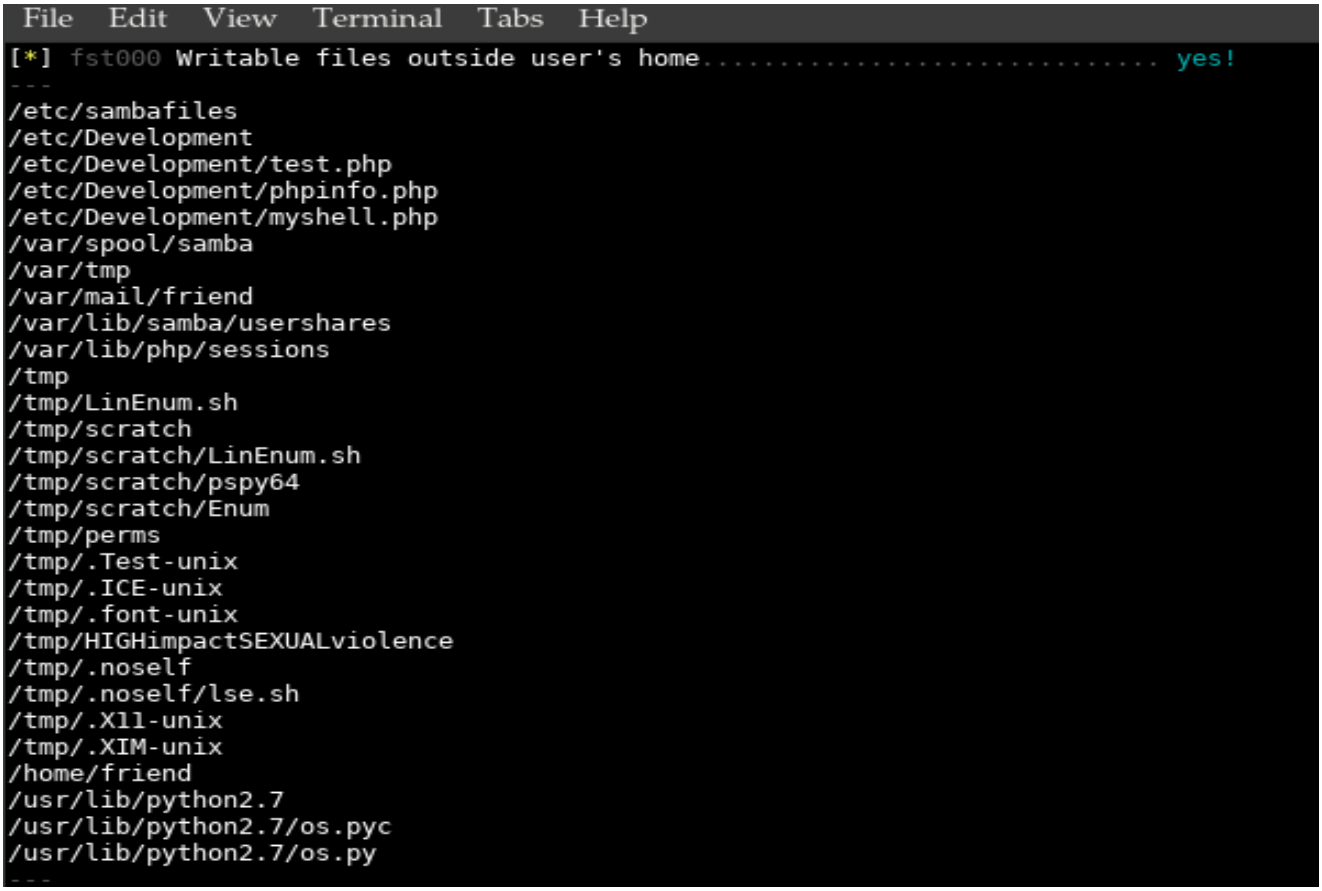
```
www-data@FriendZone:/home/friend$ ls -la
ls -la
total 36
drwxr-xr-x 5 friend friend 4096 Jan 24 00:59 .
drwxr-xr-x 3 root   root   4096 Oct  5  2018 ..
lrwxrwxrwx 1 root   root      9 Jan 24 00:59 .bash_history -> /dev/null
-rw-r--r-- 1 friend friend  220 Oct  5  2018 .bash_logout
-rw-r--r-- 1 friend friend 3771 Oct  5  2018 .bashrc
drwx------ 2 friend friend 4096 Oct  5  2018 .cache
drwx------ 3 friend friend 4096 Oct  6  2018 .gnupg
drwxrwxr-x 3 friend friend 4096 Oct  6  2018 .local
-rw-r--r-- 1 friend friend  807 Oct  5  2018 .profile
-rw-r--r-- 1 friend friend    0 Oct  5  2018 .sudo_as_admin_successful
-r--r--r-- 1 root   root     33 Oct  6  2018 user.txt
www-data@FriendZone:/home/friend$ cat user.txt
cat user.txt
a9ed20acecd6c5b6b52f474e15ae9a11
```

Trying some basic enumeration I was getting a bit annoyed with this shell but came across some some credentials for mysql I was able to reuse to login to the ssh service that was running.

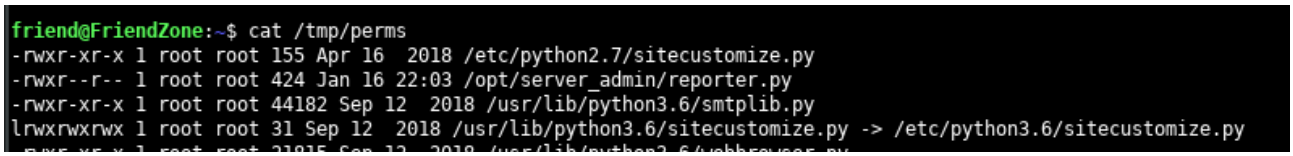
```
File  Edit  View  Terminal  Tabs  Help
www-data@FriendZone:/var/www$ ls -la
ls -la
total 36
drwxr-xr-x  8 root root 4096 Oct  6  2018 .
drwxr-xr-x 12 root root 4096 Oct  6  2018 ..
drwxr-xr-x  3 root root 4096 Jan 16 22:13 admin
drwxr-xr-x  4 root root 4096 Oct  6  2018 friendzone
drwxr-xr-x  2 root root 4096 Oct  6  2018 friendzoneportal
drwxr-xr-x  2 root root 4096 Jan 15 21:08 friendzoneportaladmin
drwxr-xr-x  3 root root 4096 Oct  6  2018 html
-rw-r--r--  1 root root  116 Oct  6  2018 mysql_data.conf
drwxr-xr-x  3 root root 4096 Oct  6  2018 uploads
www-data@FriendZone:/var/www$ cat mysql_data.conf
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
www-data@FriendZone:/var/www$
```

Once logged in with a propper shell I was able to upload Linux-Smart-Enumeration (lse.sh) which is available [from this github repo](#).

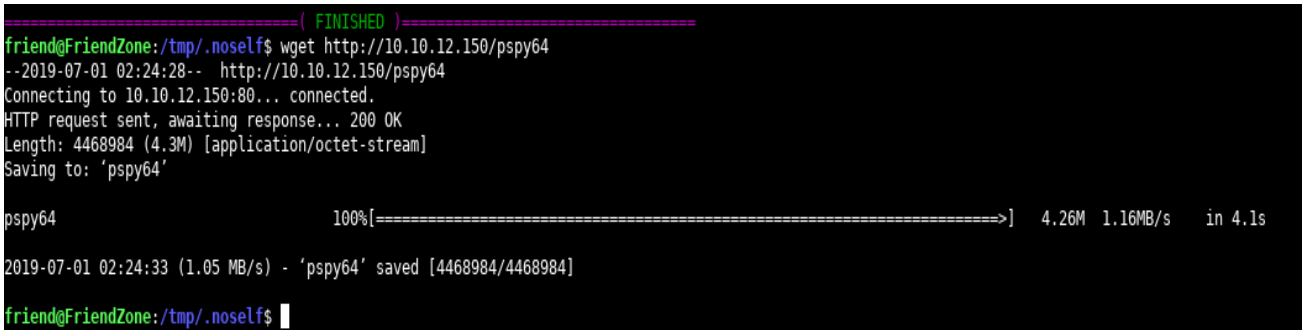
Further down I found some files that were writable in particualr some python files that caught my eye that seemed out of place for this user to have write access to.



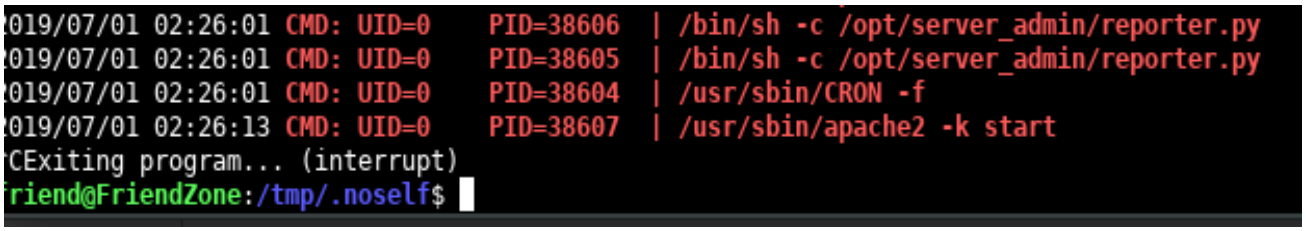
I decided to see what Python files outside of the usual might root be possibly executing?



Now that we know that root might be executing a python file I'll upload pspy which is [available here.]

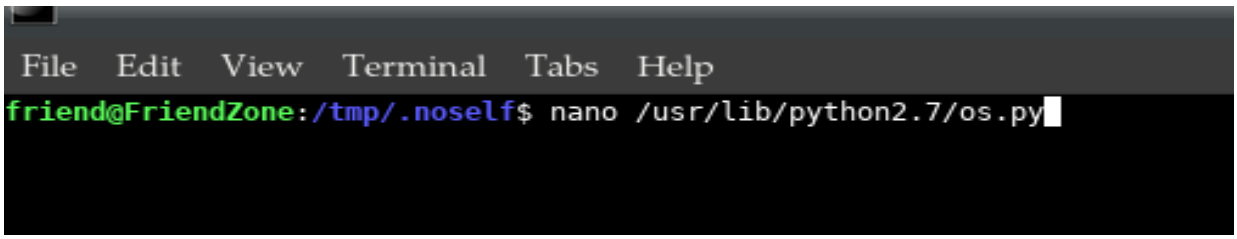


The output from pspy64 matches one of the files from earlier enumeration. Reporter.py is being executed by root!

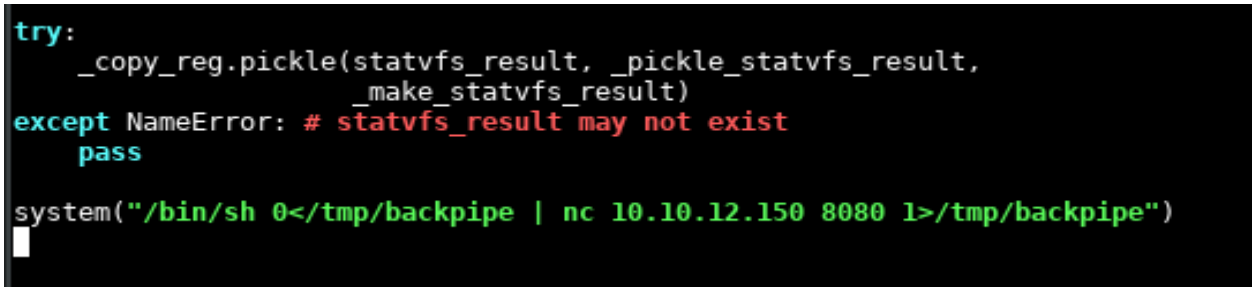


I opened up reporter.py in nano and it was a script that has yet to be completed and was importing OS module that if you recall from running lse.sh earlier was able to be written to by the current user.

Using nano again I opened up os.py file

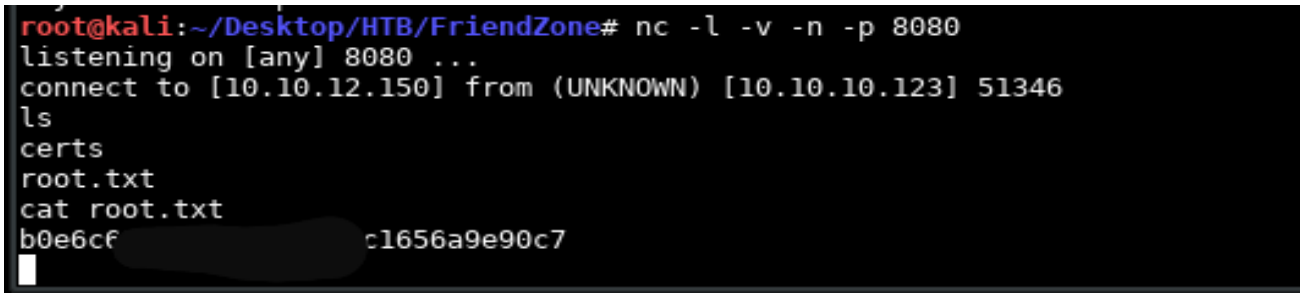


And added the following line:



By adding the line above to the os.py file it mimics calling os.system()
ie os.system(“bash_command_you_want_to_run”)

So earlier I noticesd that netcat was running, no -e flag available but were able to pipe bash using this common technique and get a root shell.



That’s all Folks