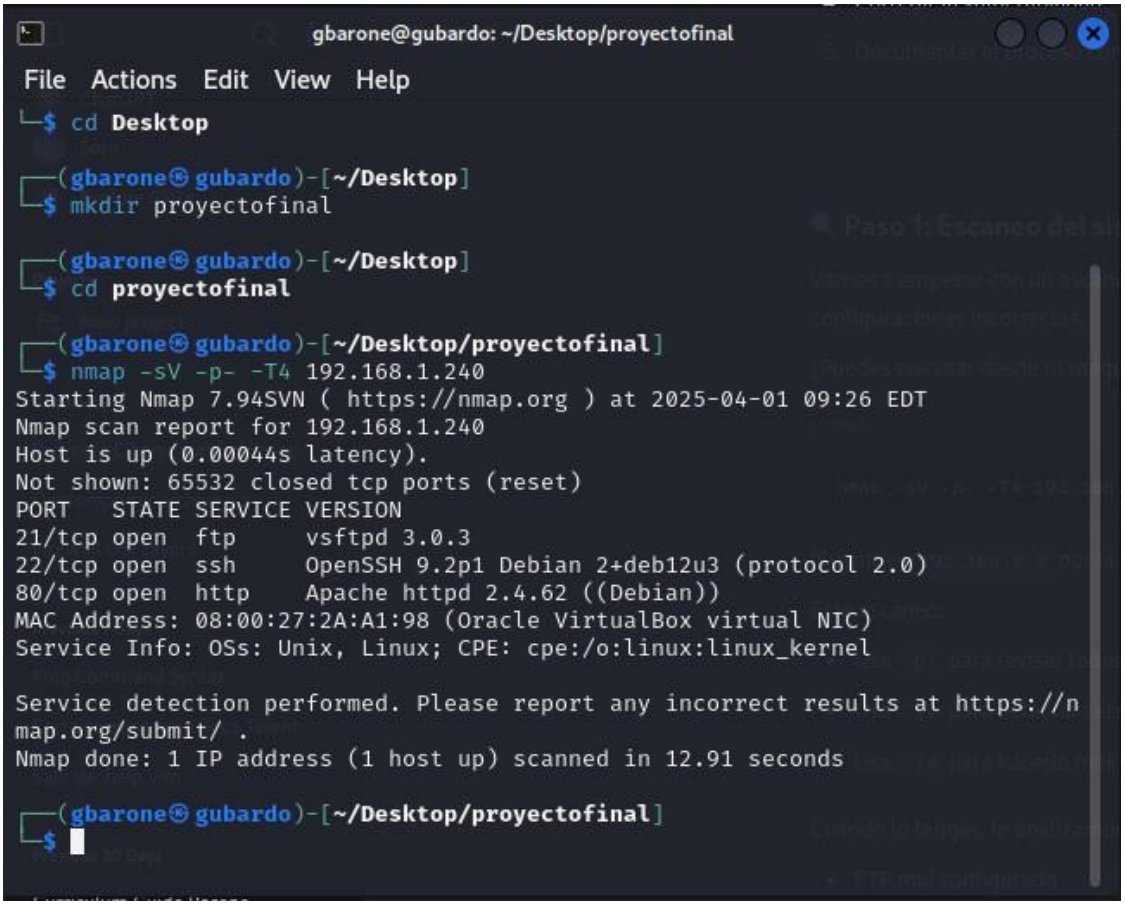
INFORME PENTESTING

Índice

1. Escaneo Completo del Sistema

Se realizó un escaneo exhaustivo de la máquina objetivo utilizando **Nmap** para identificar servicios expuestos, puertos abiertos y configuraciones vulnerables. El siguiente comando fue utilizado para obtener información sobre los servicios disponibles en la máquina:

```
nmap -sV -p- -T4 192.168.1.240
```



1.1. Resultados del Escaneo

El escaneo reveló los siguientes puertos abiertos y servicios activos:

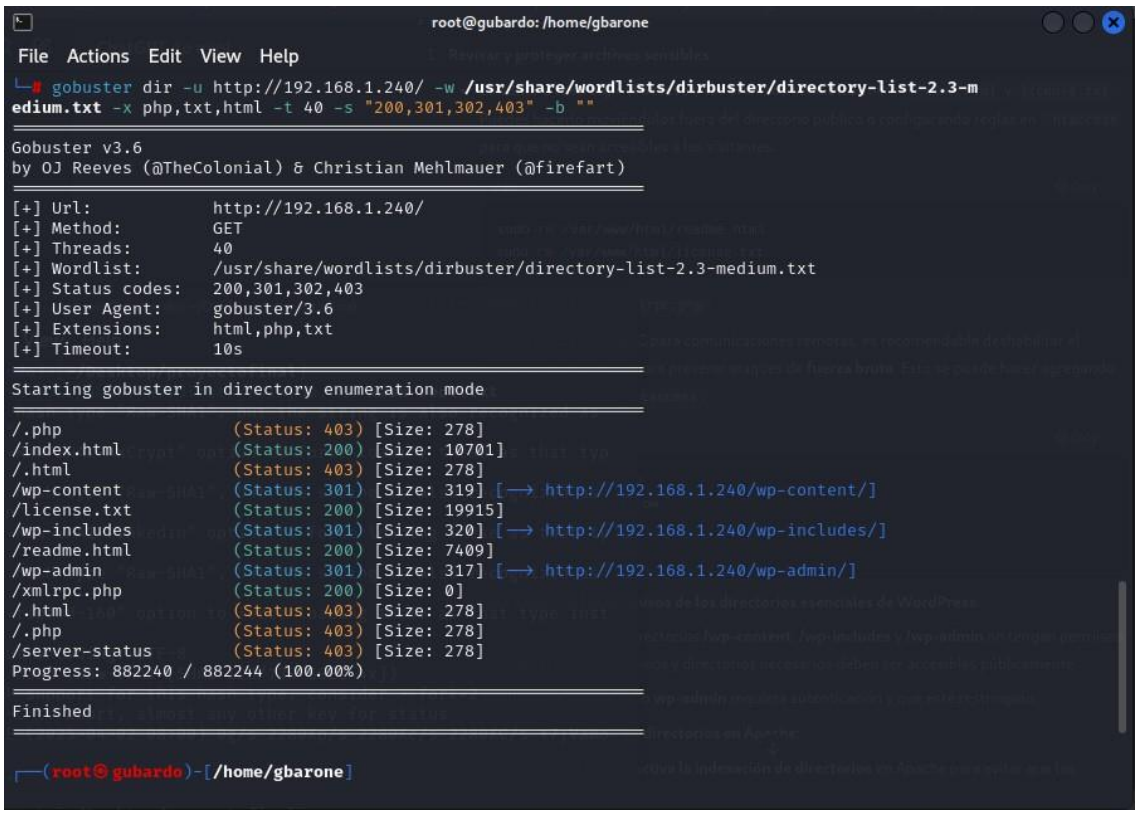
- **Puerto 21 (FTP):**
  - Servicio **vsftpd 3.0.3**, una versión conocida por su vulnerabilidad de "backdoor" en la versión 2.3.4, que se explota para ejecutar comandos arbitrarios en el servidor.
- **Puerto 22 (SSH):**
  - Servicio **OpenSSH 9.2p1** en Debian, sin embargo, no se pudo detectar una vulnerabilidad específica en esta versión durante el escaneo.
- **Puerto 80 (HTTP):**
  - Servidor **Apache httpd 2.4.62 (Debian)**, con un posible problema de configuración de indexación de directorios.

2. Identificación y Explotación de Vulnerabilidad en Apache

Tras realizar el escaneo, se identificó una vulnerabilidad potencial relacionada con la configuración de **Apache**. El servidor estaba configurado para permitir la **indexación de directorios**, lo que significa que los atacantes podían ver el contenido de directorios sin archivo index (como index.html o index.php). Este tipo de configuración puede exponer archivos sensibles y aumentar la superficie de ataque.

2.1. Proceso de Explotación

- **Acceso a los Directorios Expuestos:** Durante la revisión manual y el escaneo con **Gobuster**, se descubrió que el directorio **wp-includes** estaba accesible sin restricciones, lo que permitió a un atacante ver los archivos en este directorio sin necesidad de autenticación.



- **Revisión de Archivos Expuestos:** Al acceder al directorio wp-includes, se pudieron ver archivos críticos que podrían dar información sobre la instalación de WordPress y sus configuraciones. Esto podría ayudar a un atacante a obtener más información sobre la infraestructura de la web y a identificar vulnerabilidades adicionales para explotar.

2.2. Confirmación de la Vulnerabilidad

Acceder a la siguiente URL mostró la lista de archivos en el directorio:

http://192.168.1.240/wp-includes/

Esto confirma que **Apache** tenía habilitada la opción **Indexes** en la configuración, lo que permite la visualización no autorizada de los archivos del servidor.

Index of /wp-includes

Name	Last modified	Size	Description
Parent Directory		-	
ID3/	2024-09-10 11:23	-	
IXR/	2024-09-10 11:23	-	
PHPMailer/	2024-09-10 11:23	-	
Requests/	2024-09-10 11:23	-	
SimplePie/	2025-03-18 11:41	-	
Text/	2025-03-18 11:41	-	
admin-bar.php	2024-09-03 16:45	36K	
assets/	2025-03-18 11:41	-	
atomlib.php	2025-03-18 11:41	12K	
author-template.php	2023-05-14 13:58	19K	
block-bindings.php	2024-06-12 08:44	5.5K	
block-bindings/	2024-09-10 11:23	-	
block-editor.php	2025-03-18 11:41	28K	
block-i18n.json	2021-08-11 05:08	316	
block-patterns.php	2025-03-18 11:41	13K	
block-patterns/	2024-09-10 11:23	-	
block-supports/	2024-09-10 11:23	-	
block-template-utils.php	2025-03-18 11:41	59K	
block-template.php	2025-03-18 11:41	14K	
blocks.php	2025-03-18 11:41	102K	

3. Medidas Correctivas Implementadas

Para mitigar esta vulnerabilidad, se aplicaron las siguientes correcciones en la configuración de **Apache**:

3.1. Desactivación de la Indexación de Directorios

Se modificó el archivo de configuración de **Apache** para deshabilitar la opción **Indexes**, lo que previene que los directorios sean listados si no hay un archivo index disponible. Esto se logró editando el archivo de configuración de **Apache**:

```
sudo nano /etc/apache2/apache2.conf
```

En este archivo, se cambió:

```
Options +Indexes
```

a:

```
Options -Indexes
```

3.2. Reinicio del Servicio de Apache

Para que los cambios surtan efecto, se reinició el servicio de **Apache** con el siguiente comando:

```
sudo systemctl restart apache2
```

AGREGAMOS UN FIREWALL (IPTABLES)
Puerto 80 (HTTP) - Servidor Apache

Se configuró el firewall para permitir solo el tráfico saliente en el puerto 80 (para que el servidor pueda acceder a recursos externos si es necesario), pero se bloqueó todo el tráfico entrante en ese puerto para evitar accesos no autorizados a través de HTTP.

- **Comando aplicado:**

```
sudo iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 80 -j DROP
```

- **Explicación:**
  - sudo iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT: Esta regla permite que el tráfico saliente hacia el puerto 80 (HTTP) sea aceptado. Esto es necesario para que el servidor pueda realizar solicitudes salientes, como la descarga de actualizaciones o la consulta a otros servidores web.
  - sudo iptables -A INPUT -p tcp --dport 80 -j DROP: Esta regla bloquea todo el tráfico entrante en el puerto 80 (HTTP), impidiendo que los atacantes puedan acceder al servidor a través de este puerto. Esto es una medida de seguridad para proteger el servidor de posibles intentos de explotación de vulnerabilidades en el servicio web.

3.3. Verificación de los Resultados

Después de aplicar los cambios, se volvió a intentar acceder a los directorios sensibles a través del navegador y se verificó que la lista de archivos ya no estaba disponible.

4. Informe Detallado de las Medidas Aplicadas

4.1. Vulnerabilidad Detectada

- **Vulnerabilidad:** Exposición de directorios en **Apache**.
- **Descripción: Apache** estaba configurado para permitir la indexación de directorios (**Options +Indexes**), lo que permite a los usuarios no autorizados ver los archivos contenidos en directorios sin archivo index.
- **Impacto:** Permite a un atacante obtener información sobre la infraestructura del servidor web, incluyendo la versión de los componentes y la ubicación de archivos sensibles.

4.2. Proceso de Explotación

- Escaneo de puertos con **Nmap** para identificar servicios activos.
- Detección de la vulnerabilidad de indexación de directorios en **Apache**.
- Acceso a directorios sensibles como **wp-includes** y exploración de los archivos disponibles.

4.3. Medidas Correctivas

- Desactivación de la indexación de directorios en **Apache** mediante la configuración de **Options -Indexes**.
- Reinicio de **Apache** para aplicar la nueva configuración.
- Verificación de que la exposición de directorios ya no está disponible.

4.4. Recomendaciones

- **Revisión regular de configuraciones:** Asegurarse de que la opción **Indexes** esté desactivada en todas las configuraciones de **Apache**.
- **Uso de reglas de firewall:** Limitar el acceso a directorios sensibles mediante reglas de firewall.
- **Auditorías de seguridad periódicas:** Realizar auditorías regulares para detectar configuraciones incorrectas y vulnerabilidades.

5. Conclusiones y Continuación

En esta fase, se identificó y explotó una vulnerabilidad de exposición de directorios en **Apache**. Las medidas correctivas se implementaron con éxito para bloquear esta vulnerabilidad, y se verificó que ya no era posible acceder a los directorios sensibles.