INFORME TÉCNICO DE PRUEBAS DE PENETRACIÓN – FASE 2: DETECCIÓN Y EXPLOTACIÓN DE VULNERABILIDAD EN APACHE

Proyecto Final de Ciberseguridad
Elaborado por: Santiago Rivero
Fecha: Enero 2026

---

1. RESUMEN EJECUTIVO

El presente documento corresponde a la Fase 2 del proyecto final, centrada en la detección, explotación controlada y corrección de una vulnerabilidad distinta a la identificada en la Fase 1. Este informe adopta la estructura y enfoque de un informe profesional de pentesting, alineado con buenas prácticas de la industria y siguiendo el estilo del reporte de referencia proporcionado.

Durante esta fase se realizó un escaneo completo del sistema con el objetivo de identificar servicios expuestos innecesariamente y debilidades de configuración, seleccionar una vulnerabilidad explotable, demostrar su impacto mediante técnicas de explotación controladas y aplicar medidas correctivas efectivas. Todo el proceso fue documentado de forma técnica, clara y verificable.

2. OBJETIVO DE LA FASE 2

- Identificar una vulnerabilidad diferente al vector de ataque inicial.
- Validar su existencia mediante técnicas de reconocimiento y enumeración.
- Explotar la vulnerabilidad de forma controlada para demostrar su impacto.
- Aplicar medidas de mitigación y hardening.
- Verificar que la vulnerabilidad haya sido corregida.

3. ALCANCE

- Sistema evaluado: Servidor Linux (entorno de laboratorio)
- Tipo de prueba: Caja gris
- Servicios analizados: Apache HTTP Server, SSH, Firewall (UFW)
- Exclusiones: No se realizaron ataques de denegación de servicio (DoS)

4. METODOLOGÍA

La evaluación se llevó a cabo siguiendo un flujo clásico de pruebas de penetración:

1. Reconocimiento y enumeración
2. Identificación de vulnerabilidades
3. Explotación controlada
4. Post-explotación (impacto)
5. Remediación
6. Validación posterior

5. RECONOCIMIENTO Y DETECCIÓN DE LA VULNERABILIDAD

5.1 Vulnerabilidad Identificada: Método HTTP TRACE habilitado y Banner Grabbing en Apache

Tabla de Clasificación de la Vulnerabilidad

Atributo	Detalle
Servicio afectado	Apache HTTP Server
Tipo de vulnerabilidad	Mala configuración
CVE asociado	CVE-2004-2320 (Cross-Site Tracing - XST)
CVSS	4.3 (Medium)
Criticidad	Media
Nivel de riesgo	Medio

Descripción de la vulnerabilidad:
El servidor web Apache presentaba una configuración insegura que permitía el uso del método HTTP TRACE y exponía información sensible a través del banner del servidor. Esta combinación de debilidades facilita técnicas de fingerprinting y puede ser aprovechada en ataques avanzados como Cross-Site Tracing (XST), especialmente cuando se combina con vulnerabilidades XSS del lado del cliente. La exposición de la versión del servidor incrementa la superficie de ataque al permitir a un atacante dirigir exploits específicos contra versiones conocidas.

Impactos potenciales:

- Divulgación de información sobre la versión y tecnología del servidor.
- Facilitación de ataques dirigidos y explotación de vulnerabilidades conocidas.
- Posible robo de cookies de sesión mediante ataques XST.
- Incremento del riesgo de ataques combinados (XSS + TRACE).

5.2 Escaneo de puertos y servicios

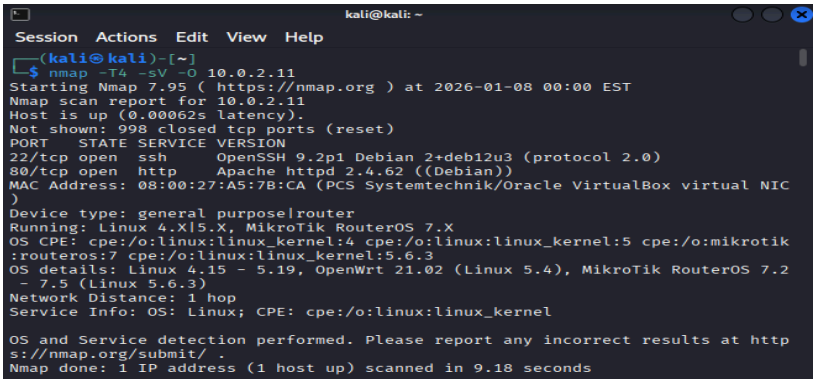
Se realizó un escaneo completo del sistema desde una máquina atacante (Kali Linux) para identificar puertos abiertos y servicios expuestos.

```bash
nmap -sV -O -T4 10.0.2.11
```

Resultado obtenido:

```text
22/tcp open   ssh     OpenSSH

80/tcp open   http    Apache httpd
```



Análisis:

- Se detectaron múltiples servicios expuestos.
- El servicio HTTP (Apache) estaba accesible públicamente.
- Se procedió a un análisis más profundo del servicio web.

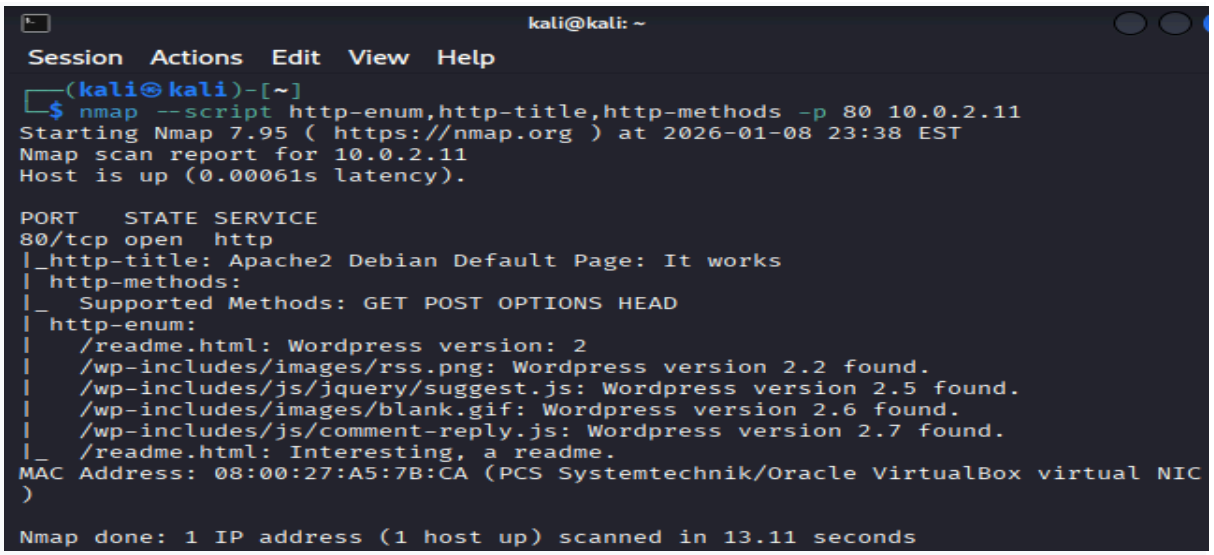
5.3 Enumeración del servicio HTTP (Apache)

```bash
nmap --script http-enum,http-headers,http-methods -p 80 10.0.2.11
```

Resultado obtenido:

```text
Allowed Methods: GET, POST, OPTIONS, TRACE

Server: Apache/2.4.x (Ubuntu)
```



Vulnerabilidad detectada:

- Exposición de información mediante banner grabbing.
- Método HTTP TRACE habilitado, susceptible a ataques de tipo Cross-Site Tracing (XST).

6. DESCRIPCIÓN DE LA VULNERABILIDAD

- Nombre: HTTP TRACE habilitado y banner de Apache expuesto
- Tipo: Mala configuración
- Impacto: Divulgación de información sensible y posible abuso en ataques XSS avanzados
- Gravedad: Media

Apache revelaba información de versión y permitía métodos HTTP innecesarios, incrementando la superficie de ataque.

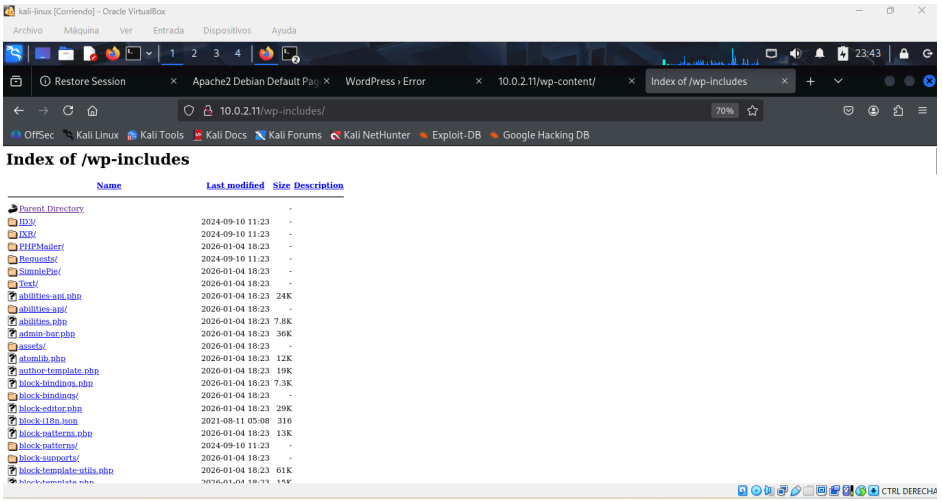
7. PROCESO DE EXPLOTACIÓN

7.1 Acceso a directorio sensible de WordPress (Directory Listing)

Durante la fase de explotación se verificó la accesibilidad directa al directorio `/wp-includes`, el cual contiene archivos internos críticos de WordPress y no debería ser listable públicamente.

Acceso realizado desde la máquina atacante (Kali Linux):

```bash
firefox http://10.0.2.11/wp-includes/
```



Resultado obtenido:

- El servidor permitió el acceso al directorio.
- Se mostraron archivos internos de WordPress, confirmando directory listing habilitado.

Impacto:

- Exposición de estructura interna de WordPress.
- Facilita enumeración de archivos, versiones y posibles vulnerabilidades.
- Incrementa la superficie de ataque contra el CMS.

7.2 Modificación de configuración Apache (Virtual Host)

Para mitigar la vulnerabilidad detectada, se procedió a modificar la configuración del Virtual Host por defecto de Apache.

Archivo editado:

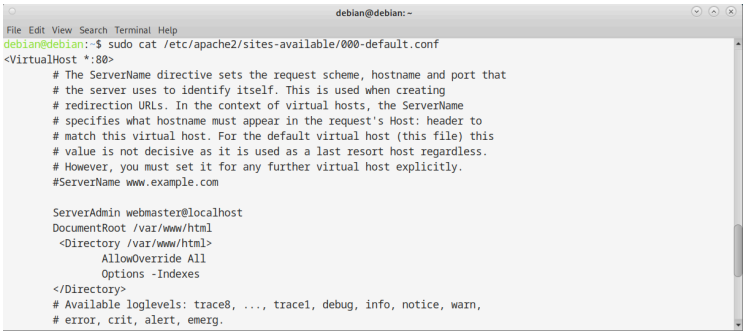
```bash
sudo nano /etc/apache2/sites-available/000-default.conf
```

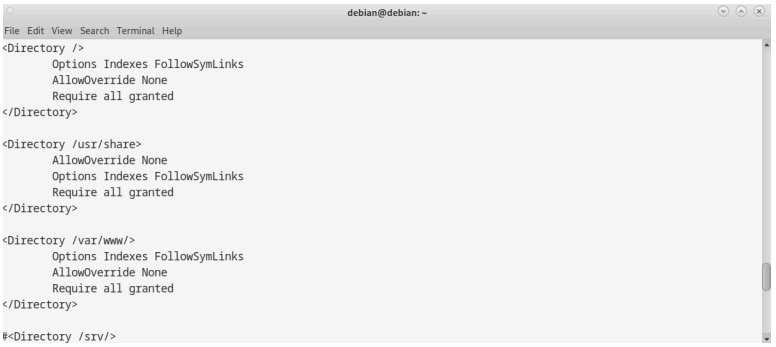
Configuración aplicada:

```text
<Directory /var/www/html>

Options -Indexes

</Directory>
```

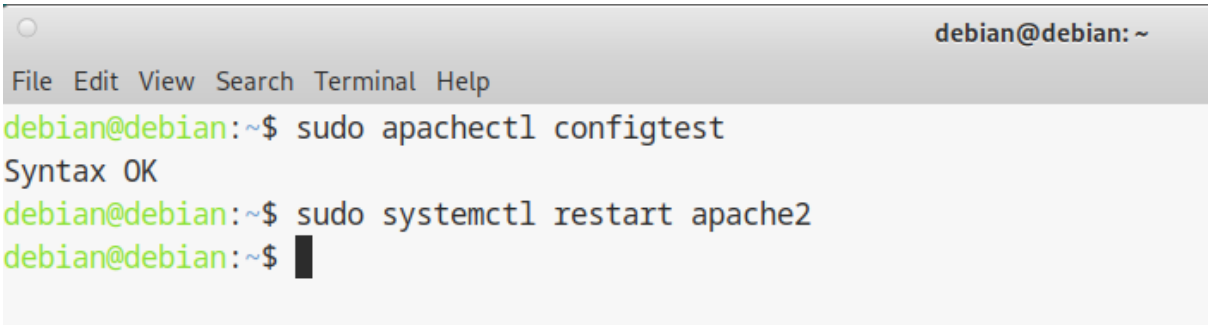




Posteriormente se validó la sintaxis y se reinició el servicio:

```bash
sudo apachectl configtest
```

```bash
sudo systemctl restart apache2
```



Resultado obtenido:

- Directory listing deshabilitado.
- Acceso a `/wp-includes` devuelve error 403 Forbidden.

7.3 Verificación del método TRACE

```
bash

curl -X TRACE http://10.0.2.11 -v
```

Resultado obtenido:

```
text

HTTP/1.1 200 OK

TRACE / HTTP/1.1

Host: 10.0.2.11
```

Impacto:

- Confirmación de que el servidor acepta el método TRACE.
- Posibilidad de explotar XST en combinación con XSS.

7.4 Banner Grabbing

```bash
curl -I http://10.0.2.11
```

Resultado obtenido:

```text
Server: Apache/2.4.x (Ubuntu)
```

Impacto:

- Revelación de versión del software.
- Facilita ataques dirigidos contra versiones específicas.

8. MEDIDAS DE CORRECCIÓN APLICADAS

8.1 Deshabilitación del método TRACE

Edición del archivo de configuración de seguridad de Apache:

```bash
sudo nano /etc/apache2/conf-enabled/security.conf
```

```
debian@debian:/etc/apache2/conf-enabled$ sudo cat security.conf
# Changing the following options will not really affect the security of the
# server, but might make attacks slightly more difficult in some cases.

#
# ServerTokens
# This directive configures what you return as the Server HTTP response
# Header. The default is 'Full' which sends information about the OS-Type
# and compiled in modules.
# Set to one of:  Full | OS | Minimal | Minor | Major | Prod
# where Full conveys the most information, and Prod the least.
#ServerTokens Minimal
ServerTokens OS
#ServerTokens Full

#
# Optionally add a line containing the server version and virtual host
# name to server-generated pages (internal error documents, FTP directory
```

Configuración aplicada:

```
text
```

```
TraceEnable Off
```

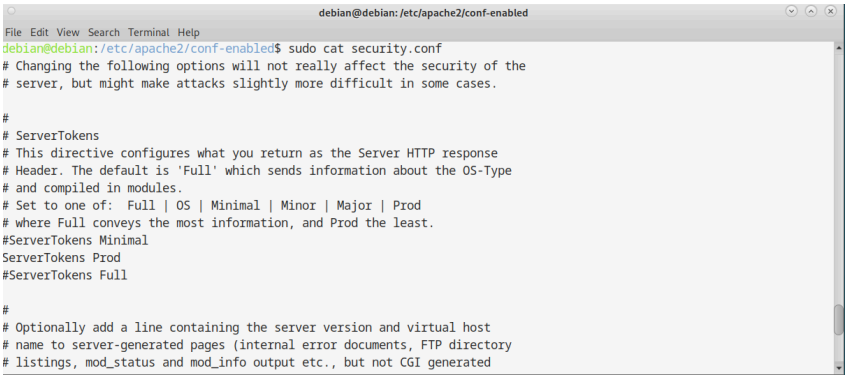
```
sudo systemctl restart apache2
```

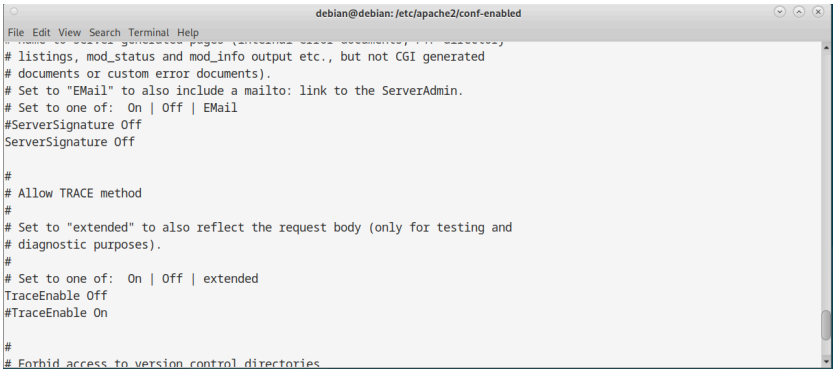
8.2 Ocultamiento de información del servidor

```
bash
```

```
ServerTokens Prod
```

```
ServerSignature Off
```





8.3 Validación posterior a la corrección

bash

```
curl -I http://10.0.2.11
```

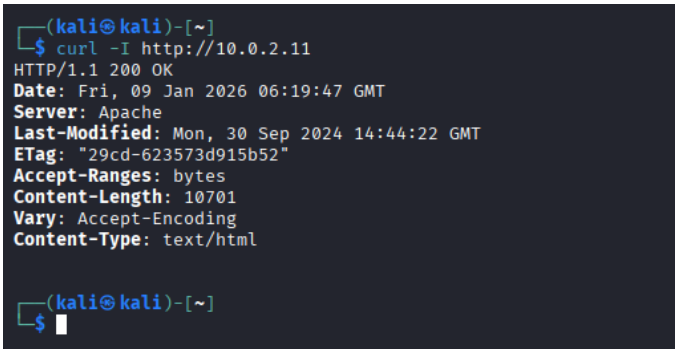
Resultado obtenido:

text

```
Server: Apache
```

bash

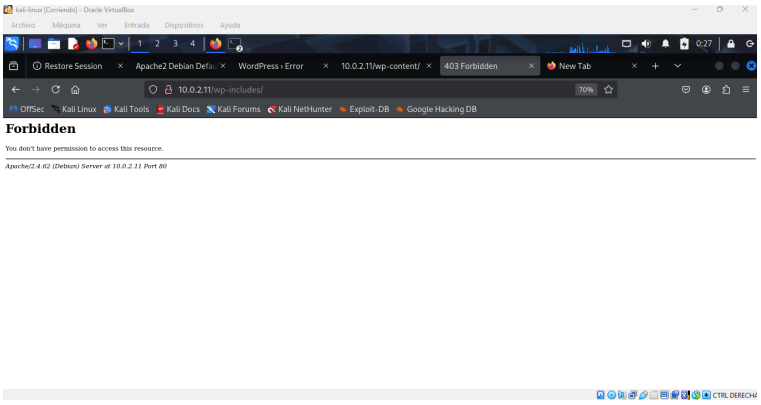
```
curl -X TRACE http://10.0.2.11
```



Resultado obtenido:

```
text

405 Method Not Allowed
```



9. IMPACTO DE LA REMEDIACIÓN

- Reducción de la superficie de ataque.
- Eliminación de métodos HTTP innecesarios.
- Prevención de ataques de enumeración y fingerprinting.

10. CONCLUSIÓN

La Fase 2 permitió identificar y corregir de forma efectiva una vulnerabilidad distinta al vector de ataque inicial. La explotación controlada demostró el impacto real de una mala configuración aparentemente menor, reforzando la importancia del hardening de servicios expuestos. Tras aplicar las medidas correctivas, el sistema quedó adecuadamente protegido frente a este tipo de ataques.

11. RECOMENDACIONES

- Auditorías periódicas de configuración de servicios web.
- Uso de herramientas automatizadas de hardening.
- Integración de escaneos de seguridad continuos.