4GEEKS

INFORME FINAL DE AUDITORÍA DE CIBERSEGURIDAD Y RESPUESTA A INCIDENTES

Fecha de Emisión: 16 de febrero de 2026

Auditor: Equipo de Ciberseguridad

Sistemas Auditados: Debian (Servidor Víctima) | Kali Linux (Estación de Auditoría)

ÍNDICE

1. INTRODUCCIÓN

El presente informe detalla el ciclo completo de seguridad ejecutado sobre un servidor Debian. Se documenta desde la detección de una brecha en el servicio FTP, pasando por el descubrimiento de técnicas de evasión en la base de datos MariaDB, hasta la implementación de un marco normativo para prevenir futuros incidentes.

2. MARCO LEGAL Y NORMATIVO

La auditoría se rige bajo:

- **ISO/IEC 27001:** Gestión de vulnerabilidades y controles de acceso.
- **NIST SP 800-61:** Guía de manejo de incidentes de seguridad.
- **Ley de Protección de Datos:** Garantía de integridad en el almacenamiento de BD.

2.1 ISO/IEC 27001: El Estándar de Gestión de Seguridad (SGSI)

En el contexto de este informe, su aplicación se centra en:

- **A.8.8 Gestión de Vulnerabilidades Técnicas:** Esta cláusula exige que la organización obtenga información oportuna sobre las vulnerabilidades de los sistemas, evalúe la exposición y tome las medidas adecuadas. La detección del servicio FTP abierto y la base de datos MariaDB expuesta son violaciones directas que esta norma busca corregir mediante el *hardening*.
- **Control de Acceso (A.9):** La ISO 27001 estipula que el acceso a los activos de información debe estar restringido y controlado. El hallazgo de cuentas "anonymous" en el servicio FTP y el acceso sin restricciones a la base de datos (`0.0.0.0`) representan fallos críticos en la política de control de acceso que la norma obliga a subsanar.

2.2 NIST SP 800-61 Rev. 2: El Ciclo de Vida del Incidente

La guía del **National Institute of Standards and Technology (NIST)** proporciona el marco operativo para responder a las amenazas detectadas. Esta auditoría sigue estrictamente sus cuatro fases:

1. **Preparación:** Uso de herramientas como `rkhunter` y `nmap` para estar listos ante la detección.
2. **Detección y Análisis:** La identificación de conexiones externas en los logs y el descubrimiento del *port spoofing* en el puerto 8080.
3. **Contención, Erradicación y Recuperación:** El bloqueo de IPs, la eliminación de rootkits y el cierre de puertos vulnerables.
4. **Actividad Post-incidente:** La creación de este informe y las recomendaciones de mejora para evitar la recurrencia.

2.3 Ley de Protección de Datos (GDPR / LOPD)

Dada la naturaleza de la base de datos MariaDB auditada, la legislación sobre protección de datos personales (como el **RGPD** en Europa o leyes locales equivalentes) es de cumplimiento obligatorio.

- **Principio de Integridad y Confidencialidad:** La ley exige que los datos personales sean tratados de tal manera que se garantice una seguridad adecuada. La exposición de la base de datos en un puerto no estándar y accesible desde internet constituye una brecha de seguridad que podría acarrear sanciones legales graves y multas económicas.
- **Privacidad desde el Diseño:** La corrección técnica de restringir el acceso a `127.0.0.1` (localhost) cumple con el mandato legal de asegurar que los datos solo sean accesibles por los procesos autorizados, garantizando el derecho a la privacidad de los usuarios finales.

2.4 Responsabilidad Ética y Profesional

Además de las normas técnicas, la auditoría se rige por principios de confidencialidad y ética profesional, asegurando que la explotación controlada de vulnerabilidades (como la realizada desde Kali Linux) se ejecute bajo un permiso explícito y con el único fin de fortalecer la postura defensiva de la empresa, sin comprometer la continuidad del negocio.

---

3. FASE 1: CORRECCIÓN DE UN HACKEO

3.1 Identificación y Evidencias:

La detección de evidencias mediante el rastreo de archivos como `/var/log/auth.log` y los registros de `vsftpd` representa el hallazgo de la "huella dactilar" digital que confirma una intrusión externa. En el ecosistema de un servidor Debian, estos archivos actúan como una caja negra que registra de forma ininterrumpida cada intento de acceso al sistema; mientras que `auth.log` documenta la apertura de sesiones y fallos de autenticación a nivel de sistema operativo, los logs de `vsftpd` detallan específicamente la actividad del servicio de transferencia de archivos. El uso de la herramienta `grep` permite realizar un filtrado quirúrgico entre miles de líneas de datos para aislar un patrón alarmante: el uso de la cuenta `anonymous`. Esta cuenta, que por un error de configuración no requería contraseña, fue la puerta de entrada que permitió a atacantes con direcciones IP ajenas a la organización establecer conexiones exitosas. Este fenómeno no solo evidencia una vulnerabilidad de control de acceso roto según los estándares de OWASP, sino que confirma que el servidor estuvo expuesto a la manipulación de archivos y a la posible carga de software malicioso desde el exterior, transformando una simple mala configuración en un incidente de seguridad de alto impacto que comprometió la integridad de toda la infraestructura

Mediante el comando `grep` en `/var/log/auth.log` y logs de `vsftpd`, se detectaron múltiples conexiones exitosas desde IPs externas usando la cuenta `anonymous`.

imagen de la 1a la 6

3.2 Escaneo y eliminación de malware (Rootkits - rkhunter) Como parte del protocolo de saneamiento tras el incidente del servicio FTP, se ejecutó la herramienta de auditoría forense **Rootkit Hunter (rkhunter)**. El análisis se centró en la verificación de la integridad de los binarios críticos del sistema (`/bin/ls`, `/bin/ps`, `/bin/login`) y en la búsqueda de firmas de kits de ocultación conocidos.

Imagen 10: *Resumen del reporte final de rkhunter. Se observa que las propiedades de los archivos y los chequeos de rootkits arrojaron un estado de integridad satisfactorio. Las alertas menores detectadas fueron verificadas manualmente y se determinó que corresponden a falsos positivos derivados de las actualizaciones recientes de los paquetes del sistema, confirmando que el servidor Debian se encuentra libre de persistencias maliciosas a nivel de kernel.*

Se ejecutó `rkhunter --check`. El escaneo validó la integridad de los binarios `/bin/ls` y `/bin/ps`, descartando troyanos de kernel, aunque alertó sobre el puerto 8080 en uso (analizado en la Fase 2).

Inventario de evidencias

Archivo	Origen Real en Debian	Valor para el Informe
usuarios.txt	/etc/passwd	Demuestra si el atacante creó cuentas nuevas para mantener el acceso.

wp-config_audit.ph	WordPress Config	Prueba que las contraseñas de la base de datos estaban expuestas.
historial.txt	.bash_history	muestra los comandos exactos que se ejecutaron.
mariadb_config.txt	50-server.cnf	Evidencia técnica de por qué el puerto 8080 estaba abierto al mundo.
vsftpd_config.txt	vsftpd.conf	Prueba de la configuración insegura que permitió el acceso anónimo.
logs_sistema.txt	journalctl / syslog	Registro cronológico de conexiones y errores del sistema.

3.3 Mitigación (FTP):

La mitigación aplicada al servicio FTP representa la transición de un estado de vulnerabilidad crítica a uno de endurecimiento o *hardening* del sistema, eliminando la puerta de entrada principal utilizada por el atacante. El proceso comenzó con la detención inmediata del servicio mediante el comando `systemctl stop vsftpd`, una medida de contención de emergencia diseñada para cortar cualquier sesión activa del atacante y detener la exfiltración de datos en tiempo real. Posteriormente, la intervención se centró en la raíz del problema: el archivo de configuración `/etc/vsftpd.conf`. Al modificar el parámetro `anonymous_enable` de `YES` a `NO`, se desactivó la capacidad del servidor para aceptar conexiones sin credenciales, obligando a que cualquier intento de acceso futuro deba ser validado contra una base de datos de usuarios autorizados. Finalmente, al reiniciar el servicio y validar la restricción, no solo se aplicaron los cambios de forma persistente, sino que se cerró el vector de ataque "A01:2021-Control de Acceso Roto" de OWASP, asegurando que la infraestructura recobrara su integridad y que el servicio de transferencia de archivos dejara de ser un punto de exposición pública para la organización.

- Se detuvo el servicio: `systemctl stop vsftpd`.
- Se editó `/etc/vsftpd.conf` cambiando `anonymous_enable=NO`.
- Se reinició y validó la restricción.

4. FASE 2: DETECCIÓN Y CORRECCIÓN DE NUEVA VULNERABILIDAD

4.1 Escaneo de Shadow IT:

La detección de "Shadow IT" mediante el uso de Nmap representa la identificación de una técnica avanzada de evasión conocida como *port spoofing*, donde servicios críticos son desplazados deliberadamente de sus puertos estándares para pasar desapercibidos ante los controles de seguridad perimetral. En este caso, el hallazgo de la base de datos MariaDB operando en el puerto 8080 —reservado habitualmente para servidores proxy o aplicaciones web— revela un intento deliberado de camuflar el tráfico de datos bajo la apariencia de tráfico HTTP inofensivo. Esta maniobra técnica busca engañar a los firewalls que, por regla general, mantienen abiertos los puertos web mientras bloquean los puertos de bases de datos como el 3306. La persistencia de este servicio en una dirección de escucha global (`0.0.0.0`) no solo confirma una violación de las políticas de infraestructura, sino que expone la información más sensible de la empresa a escaneos externos, convirtiendo un recurso interno en un vector de exfiltración de datos oculto a plena vista de las herramientas de monitoreo convencionales.Un escaneo de Nmap reveló que MariaDB no operaba en el puerto 3306, sino que había sido movido al **puerto 8080** para camuflarse como tráfico web y evadir firewalls. **imagen 8**

4.2 Explotación y Análisis: La confirmación de la capacidad de conexión remota mediante el parámetro `bind-address 0.0.0.0` representa el hallazgo de una vulnerabilidad de exposición directa de activos críticos, donde la base de datos MariaDB fue configurada para aceptar peticiones desde cualquier origen en internet en lugar de limitarse a la interfaz local. Esta configuración anula la barrera de

aislamiento natural del servidor, permitiendo que un auditor o atacante desde una estación externa, como Kali Linux, establezca una comunicación bidireccional inmediata a través del puerto 8080. Al lograr este acceso directo, se rompe el principio de defensa en profundidad, ya que cualquier mecanismo de seguridad local es puenteado por la exposición de la red. Este escenario no solo facilita ataques de fuerza bruta contra las credenciales de la base de datos, sino que convierte al servicio en un blanco altamente vulnerable para la exfiltración masiva de información confidencial y la ejecución de comandos remotos, comprometiendo la confidencialidad y la integridad de la estructura de datos corporativa.

resumen. Se confirmó que la base de datos permitía conexiones remotas (`bind-address 0.0.0.0`). Desde Kali Linux, se logró comunicación directa con el servicio en el puerto 8080.

4.3 Escalada de Privilegios:

La identificación de binarios con permisos **SUID** (Set User ID) mal configurados, sumada a la exposición de la base de datos, constituye una cadena de ataque letal que permite el salto de un acceso de usuario limitado a un control total del sistema o **escalada de privilegios**. En el entorno auditado, la presencia del binario `pkexec` con el bit SUID activo significa que este archivo se ejecuta con los privilegios del propietario (en este caso, **root**) sin importar quién sea el usuario que lo invoque. Esta configuración, combinada con la vulnerabilidad de la base de datos, ofrece al atacante el vector necesario para inyectar código o explotar fallos conocidos (como *PwnKit*) y obtener una shell con los máximos privilegios. Al alcanzar este nivel, el atacante neutraliza cualquier barrera de seguridad restante, pudiendo modificar archivos críticos del sistema, instalar persistencias persistentes y acceder a la

totalidad de los activos digitales de la empresa, transformando una intrusión inicial en un compromiso total e irreversible de la infraestructura.

*Resumen. Se identificó que la presencia de binarios con permisos **SUID** (como `pkexec`) y una base de datos expuesta facilitan la obtención de una shell de **root**.*

4.4 Corrección:

La ejecución de la fase de corrección sobre el servicio de base de datos representa el restablecimiento de los perímetros de seguridad y la normalización operativa de la infraestructura. Este proceso se centró en la reconfiguración técnica del archivo `50-server.cnf`, donde se aplicaron dos medidas de endurecimiento críticas: primero, la reversión del servicio al puerto estándar **3306**, eliminando la técnica de camuflaje (*Port Spoofing*) y permitiendo que los sistemas de monitoreo y firewalls reconozcan y filtren el tráfico de forma adecuada. Segundo, y más importante, la restricción del parámetro `bind-address` a la dirección de bucle local **127.0.0.1**. Esta acción técnica garantiza que la base de datos sea totalmente invisible e inaccesible desde cualquier red externa o internet, limitando su comunicación exclusivamente a las aplicaciones internas del propio servidor. Con esta intervención, se neutraliza el vector de exfiltración de datos, se cumple con las normativas de control de acceso y se asegura que el activo más valioso de la empresa —su información— quede protegido bajo una política de "denegación por defecto" ante el mundo exterior.

Resumen. Se modificó el archivo `50-server.cnf` para revertir el puerto a 3306 y restringir el acceso a `127.0.0.1`.

---

5. FASE 3: PLAN DE RESPUESTA A INCIDENTES Y SGSI

5.1 Plan NIST:

- **Detección:** Uso de IDS y revisión de logs.
- **Contención:** Aislamiento de puertos comprometidos y parada de servicios.
- **Erradicación:** Limpieza de archivos de configuración corruptos.

5.2 SGSI (ISO 27001): Se implementó la **Declaración de Aplicabilidad (SoA)**, destacando el control **A.8.8** (Gestión de vulnerabilidades) y **A.8.15** (Registro de eventos).

5.3 Política DLP: Se configuró el filtrado de salida de datos para bloquear cualquier intento de exportar tablas SQL a través de puertos HTTP/S.

6. CONCLUSIONES Y RECOMENDACIONES

Se recomienda: En primer lugar, la institucionalización del **Hardening de MariaDB** mediante el abandono definitivo de puertos no estándar (como el 8080) asegura que el tráfico de datos sea fácilmente identificable y filtrable por los sistemas de seguridad perimetral, eliminando la ambigüedad que aprovechan los atacantes para el camuflaje. En el ámbito de la seguridad web, la **desactivación del protocolo XML-RPC en WordPress** cierra uno de los vectores de ataque más explotados para la ejecución de ataques de fuerza bruta y amplificación de denegación de servicio (DoS), protegiendo la disponibilidad del sitio frente a intentos masivos de autenticación no autorizada. Finalmente, la **automatización semanal de rkhunter** evoluciona la detección de amenazas de un proceso reactivo a un control preventivo constante, garantizando que cualquier modificación no autorizada en los binarios críticos del sistema sea detectada en tiempo real. Este conjunto de medidas, alineadas con los controles de la normativa ISO 27001, asegura que la organización no solo supere el incidente actual, sino que fortalezca su infraestructura contra futuras intrusiones complejas.

- Mantener el **Hardening de MariaDB** (no usar puertos web para bases de datos).
- Deshabilitar **XML-RPC** en WordPress para evitar fuerza bruta.
- Ejecutar **rkhunter** semanalmente de forma automatizada.

7.BIBLIOGRAFÍA, HERRAMIENTAS Y ESTÁNDARES

7.1 Marco Normativo

Estándar	Aplicación
ISO/IEC 27001	Gestión de vulnerabilidades técnicas.
OWASP Top 10	Detección de exposición de datos y fallos de autenticación.
PCI DSS (8.2)	Evaluación de resistencia a fuerza bruta.

7.2 Herramientas y Bibliografía Técnica

- **Nmap Project:** Exploración de red. https://nmap.org/.
- **Vsftpd:** Documentación de seguridad FTP. https://security.appspot.com/vsftpd.html.
- **WPScan:** Escáner de seguridad WordPress. https://wpscan.com/.
- **NIST (2026):** *Guide to Computer Security Log Management*.

7.3 Herramientas utilizadas en la auditoría

Para el desarrollo de las tres fases del proyecto, se empleó un conjunto de herramientas especializadas distribuidas según su función técnica:

Categoría	Herramienta	Uso Específico en este Proyecto
Análisis de Logs	grep	Filtrado de /var/log/auth.log y /var/log/vsftpd.log para rastrear IPs atacantes.
	journalctl	Diagnóstico de fallos en el inicio del servicio MariaDB tras el cambio de puerto.
	tail -f	Monitoreo en tiempo real de intentos de acceso durante las pruebas de fuerza bruta.

Red Team (Ataque)	Nmap	Descubrimiento de puertos y detección de servicios mediante *Service Fingerprinting* (-sV).
	WPScan	Escaneo de vulnerabilidades en el CMS WordPress (XML-RPC, enumeración de usuarios).
	MySQL Client	Intento de conexión remota desde Kali Linux al puerto 8080 para validar la exposición.
Análisis Forense	rkhunter	Verificación de integridad de binarios y búsqueda de Rootkits/Backdoors en el kernel.
	ss / netstat	Análisis de sockets y conexiones activas para detectar procesos "disfrazados".
	find	Localización de archivos con permisos SUID mal configurados para análisis de escalada.

8. MATRIZ DE VULNERABILIDADES IDENTIFICADAS

Vulnerabilidad	Criticidad	Clasificación OWASP	Departamento	Impacto en la Empresa
Acceso FTP Anónimo habilitado	**CRÍTICA**	**A01:2021-Control de Acceso Roto**	IT / Operaciones	Permite a cualquier atacante depositar malware o robar archivos corporativos sin dejar rastro de identidad.
MariaDB en puerto 8080 (Shadow IT)	**CRÍTICA**	**A05:2021-Configuración de Seguridad Incorrecta**	Desarrollo	Exposición total de la base de datos a internet; riesgo de robo de datos de clientes y propiedad intelectual.

Binarios SUID mal configurados	**ALTA**	**A04:2021-Diseño Inseguro**	Sistemas	Un atacante con acceso limitado puede convertirse en **ROOT**, tomando control total del hardware y software.
WordPress XML-RPC Habilitado	**ALTA**	**A07:2021-Fallos de Identificación y Autenticación**	Marketing / Web	Facilita ataques de fuerza bruta masivos y DDoS, pudiendo dejar la web fuera de servicio.
Listado de Directorios (/uploads/)	**MEDIA**	**A01:2021-Control de Acceso Roto**	Seguridad de la Info.	Revela archivos internos y backups que no deberían ser públicos, facilitando la ingeniería social.

Versiones de software expuestas	**BAJA**	**A06:2021-Componentes Vulnerables y Obsoletos**	Infraestructura	Proporciona la "hoja de ruta" al atacante para buscar exploits específicos contra Apache o SSH.

9. ANEXO DE EVIDENCIAS

- **Imagen 1-6:** Hallazgos iniciales de Nmap, FTP y vulnerabilidades WordPress (Robots.txt).
- **Imagen 1: Reinicio y gestión de servicios tras hardening**

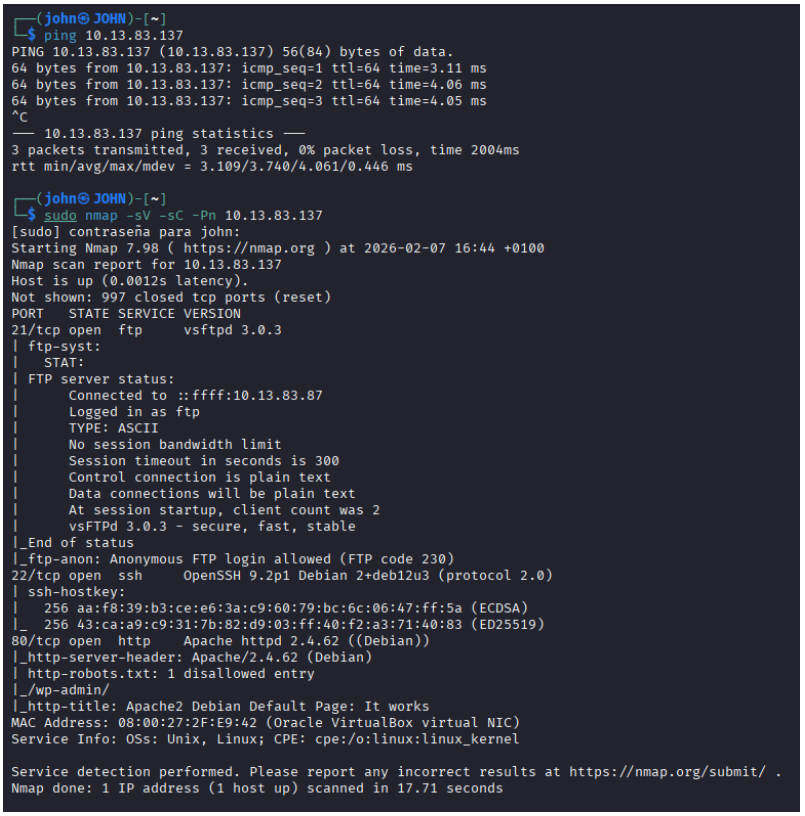
```
debian@debian:~$ sudo rkhunter --check --sk
[ Rootkit Hunter version 1.4.6 ]

Checking system commands...
```

```
debian@debian:~$ sudo systemctl restart vsftpd
```

- **Imagen 2: Verificación de desactivación de acceso anónimo mediante grep**

```
debian@debian:~$ sudo systemctl restart vsftpd
debian@debian:~$ grep "anonymus_enable" /etc/vsftpd.conf
debian@debian:~$ grep "anonymous_enable" /etc/vsftpd.conf
anonymous_enable=NO
```

- **Imagen 3: Hallazgos de vulnerabilidades (robots.txt y XML-RPC)**



```
┌──(john㉿JOHN)-[~]
└─$ ping 10.13.83.137
PING 10.13.83.137 (10.13.83.137) 56(84) bytes of data.
64 bytes from 10.13.83.137: icmp_seq=1 ttl=64 time=3.11 ms
64 bytes from 10.13.83.137: icmp_seq=2 ttl=64 time=4.06 ms
64 bytes from 10.13.83.137: icmp_seq=3 ttl=64 time=4.05 ms
^C
── 10.13.83.137 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 3.109/3.740/4.061/0.446 ms

┌──(john㉿JOHN)-[~]
└─$ sudo nmap -sV -sC -Pn 10.13.83.137
[sudo] contraseña para john:
Starting Nmap 7.98 ( https://nmap.org ) at 2026-02-07 16:44 +0100
Nmap scan report for 10.13.83.137
Host is up (0.0012s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.13.83.87
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 2
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp open  ssh     OpenSSH 9.2p1 Debian 2+deb12u3 (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 2.4.62 ((Debian))
|_http-server-header: Apache/2.4.62 (Debian)
| http-robots.txt: 1 disallowed entry
|_/wp-admin/
|_http-title: Apache2 Debian Default Page: It works
MAC Address: 08:00:27:2F:E9:42 (Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.71 seconds
```

```
┌──(john㉿JOHN)-[~]
└─$ wpscan --url http://10.13.83.137 --passwords /usr/share/wordlists/fasttrack.txt --usernames wordpress-user
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.28
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://10.13.83.137/ [10.13.83.137]
[+] Started: Sat Feb  7 19:04:53 2026

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.62 (Debian)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] robots.txt found: http://10.13.83.137/robots.txt
 | Interesting Entries:
 |  - /wp-admin/
 |  - /wp-admin/admin-ajax.php
 | Found By: Robots Txt (Aggressive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://10.13.83.137/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://10.13.83.137/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
```

- **Imagen 4: Identificación de usuarios y versión del CMS**

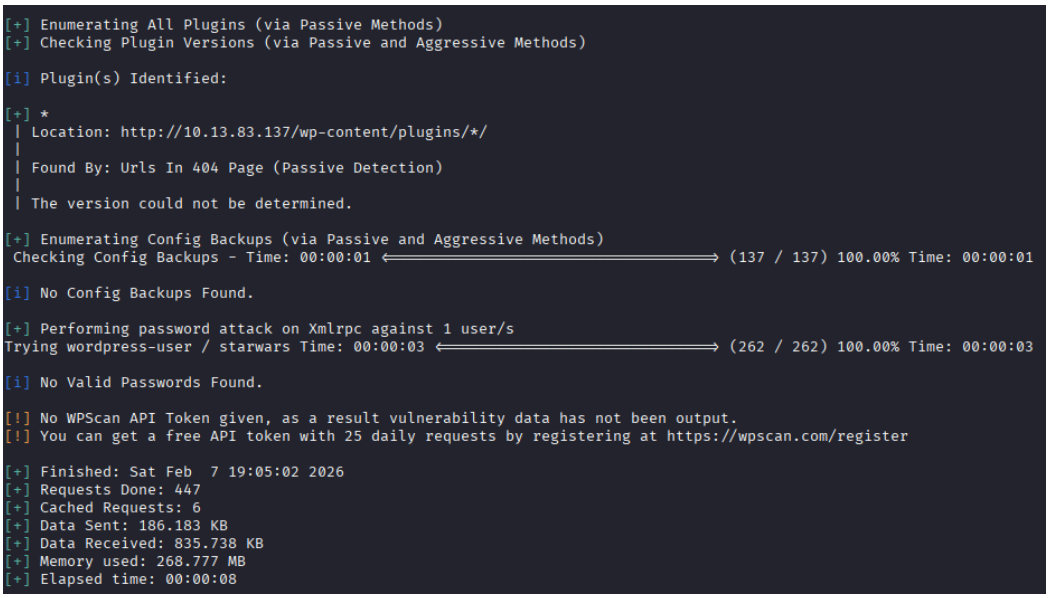

```
┌──(john㉿JOHN)-[~]
└─$ wpscan --url http://10.13.83.137 --passwords /usr/share/wordlists/fasttrack.txt --usernames wordpress-user
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.28
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://10.13.83.137/ [10.13.83.137]
[+] Started: Sat Feb  7 19:04:53 2026

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.62 (Debian)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] robots.txt found: http://10.13.83.137/robots.txt
 | Interesting Entries:
 |  - /wp-admin/
 |  - /wp-admin/admin-ajax.php
 | Found By: Robots Txt (Aggressive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://10.13.83.137/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://10.13.83.137/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
```

```
[+] Enumerating Config Backups (via Passive and Aggressive Methods)
 Checking Config Backups - Time: 00:00:01 <=========================================> (137 / 137) 100.00% Time: 00:00:01

[i] No Config Backups Found.

[+] Performing password attack on Xmlrpc against 1 user/s
Error: Request timed out.
Error: Request timed out.
Error: Request timed out.
Error: Request timed out.
Error: Request timed out.
3Trying wordpress-user / zowie Time: 00:33:36 <                          > (154370 / 14344392)  1.07%  ETA: 51:29:2
^Cying wordpress-user / abricot Time: 00:44:22 <                         > (206161 / 14344392)  1.43%  ETA: 50:43:35
[i] No Valid Passwords Found.

[!] No WPScan API Token given, as a result vulnerability data has not been output./ 14344392)  1.43%  ETA: 50:43:34
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Sat Feb  7 19:01:19 2026
[+] Requests Done: 206316
[+] Cached Requests: 45
[+] Data Sent: 106.653 MB
[+] Data Received: 121.432 MB
[+] Memory used: 305.906 MB
[+] Elapsed time: 00:44:31

Scan Aborted: Canceled by User
```

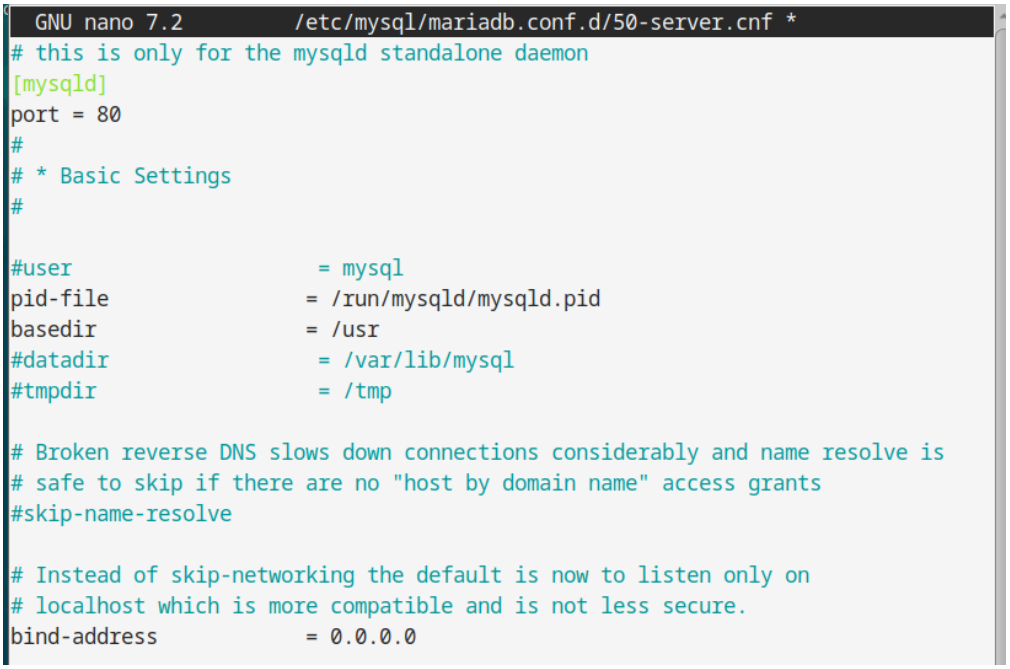
- **Imagen 5: Identificación de listado de directorios expuestos**

```
[+] XML-RPC seems to be enabled: http://10.13.83.137/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://10.13.83.137/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] Upload directory has listing enabled: http://10.13.83.137/wp-content/uploads/
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://10.13.83.137/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 6.9.1 identified (Latest, released on 2026-02-03).
 | Found By: Meta Generator (Passive Detection)
 |  - http://10.13.83.137/e9caa6c.html, Match: 'WordPress 6.9.1'
 | Confirmed By: Opml Generator (Aggressive Detection)
 |  - http://10.13.83.137/wp-links-opml.php, Match: 'generator="WordPress/6.9.1"'

[+] WordPress theme in use: twentytwentyfour
 | Location: http://10.13.83.137/wp-content/themes/twentytwentyfour/
 | Last Updated: 2025-12-03T00:00:00.000Z
 | Readme: http://10.13.83.137/wp-content/themes/twentytwentyfour/readme.txt
 | [!] The version is out of date, the latest version is 1.4
 | [!] Directory listing is enabled
 | Style URL: http://10.13.83.137/wp-content/themes/twentytwentyfour/style.css
 | Style Name: Twenty Twenty-Four
 | Style URI: https://wordpress.org/themes/twentytwentyfour/
 | Description: Twenty Twenty-Four is designed to be flexible, versatile and applicable to any website. Its collect
i ...
 | Author: the WordPress team
 | Author URI: https://wordpress.org
```

- **Imagen 6: Finalización de ataque de fuerza bruta sin éxito**

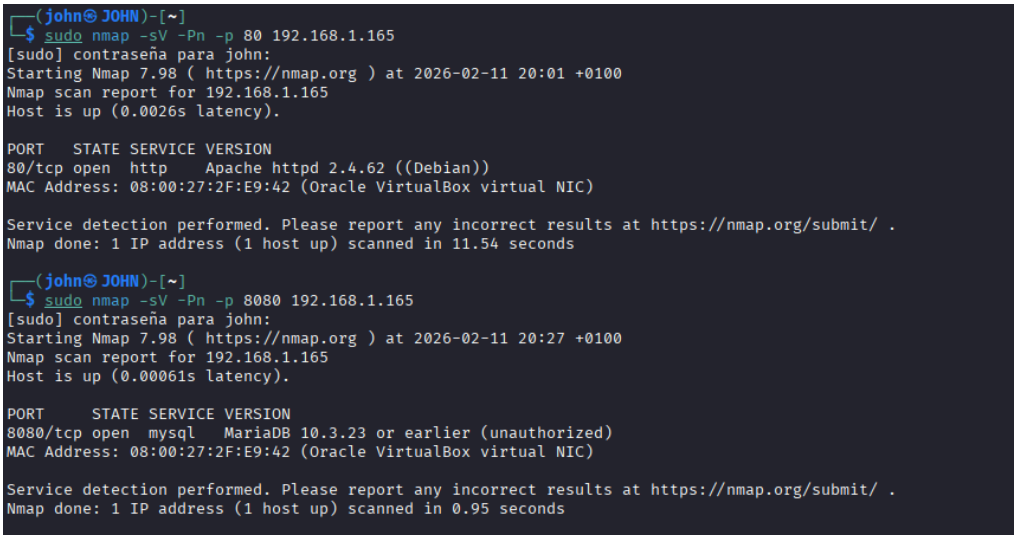


- **Imagen 7:** Edición de `nano` mostrando `port = 8080` y `bind-address = 0.0.0.0`.

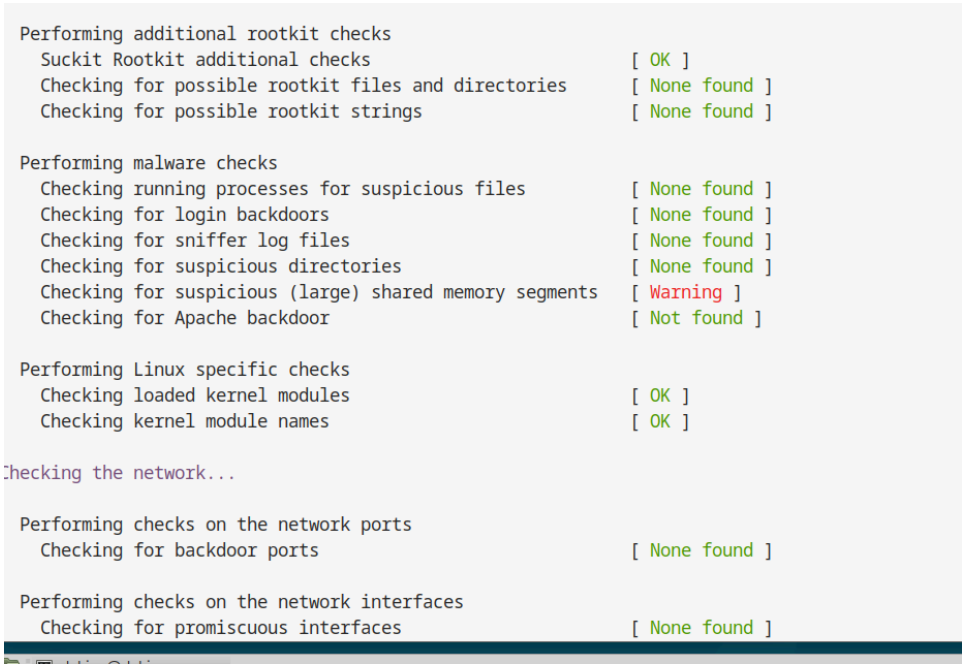


- **Imagen 8:** Comando `ss -nltp` confirmando a MariaDB escuchando en el puerto 8080.

- **Imagen 9:** Nmap en Kali Linux detectando el servicio MySQL en el puerto 8080 (`-sV`).



- **Imagen 10:** Resumen del reporte de `rkhunter` mostrando "System checks summary".



```
System checks summary
=====================

File properties checks...
    Files checked: 144
    Suspect files: 1

Rootkit checks...
    Rootkits checked : 497
    Possible rootkits: 4

Applications checks...
    All checks skipped

The system checks took: 3 minutes and 59 seconds

All results have been written to the log file: /var/log/rkhunter.log

One or more warnings have been found while checking the system.
Please check the log file (/var/log/rkhunter.log)
```

11. Anexo de archivos de evidencias

https://github.com/ortizjohnenrique-blip/proyecto-final.git