



# 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**

1.1 Agradecimientos

## **2. Marco Legal y Normativo**

## **3. Fase 1: Corrección de un Hackeo (Análisis Forense y Mitigación)**

3.1 Identificación del incidente y recolección de evidencias

3.2 Escaneo y eliminación de malware (Rootkits - rkhunter)

3.3 Bloqueo del exploit y hardening inmediato (FTP)

## **4. Fase 2: Detección y Corrección de Nueva Vulnerabilidad**

4.1 Escaneo completo del sistema (Shadow IT y Port Spoofing)

4.2 Explotación controlada y análisis de impacto (MariaDB)

4.3 Escalada de privilegios y persistencia

4.4 Corrección y validación de seguridad

## **5. Fase 3: Plan de Respuesta a Incidentes y Certificación**

5.1 Plan de respuesta basado en NIST SP 800-61

5.2 Sistema de Gestión de Seguridad de la Información (SGSI) - ISO 27001

5.3 Políticas de Prevención de Pérdida de Datos (DLP)

## **6. Conclusiones y Recomendaciones**

## **7. Bibliografía, Herramientas y Estándares**

7.1 Marco normativo

7.2 Herramientas y bibliografía técnica

7.3 Herramientas utilizadas en la auditoria

## **8. Anexo de Evidencias Fotográficas**

---

## 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.

### 1.1 AGRADECIMIENTOS

Deseo expresar mi más sincero agradecimiento a la **Profesora Daniela Maissi** y al **Profesor Simon Cervantes**, por su guía, apoyo técnico y por facilitar el entorno académico necesario para llevar a cabo este proyecto de auditoría. Sus enseñanzas han sido fundamentales para comprender la complejidad de la gestión de incidentes y la importancia del endurecimiento de sistemas en entornos críticos.

De igual manera, extiendo mi gratitud a **4Geeks Academy** por proporcionar los recursos, la plataforma y el ecosistema de aprendizaje que permitieron el desarrollo de este análisis.

Finalmente, agradezco a la comunidad de software libre por proveer las herramientas que hicieron posible este análisis forense, permitiendo el desarrollo de competencias esenciales en el ámbito de la ciberseguridad ética.

## 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.
- 

## 3. FASE 1: CORRECCIÓN DE UN HACKEO

**3.1 Identificación y Evidencias:** 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`.

**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).

### 3.3 Mitigación (FTP):

- 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:** 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.

**4.2 Explotación y Análisis:** 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:** 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:** 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:

- 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 auditoria

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 <code>/var/log/auth.log</code> y <code>/var/log/vsftpd.log</code> 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.

<b>Red Team (Ataque)</b>	Nmap	Descubrimiento de puertos y detección de servicios mediante <i>Service Fingerprinting</i> (-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.
<b>Análisis Forense</b>	rkhunter	Verificación de integridad de binarios y búsqueda de Rootkits/Backdoors en el kernel.
	ss / netstat	Ánalisis de sockets y conexiones activas para detectar procesos "disfrazados".
	find	Localización de archivos con permisos SUID mal configurados para análisis de escalada.

---

## 8. ANEXO DE EVIDENCIAS (Nomenclatura Final)

- **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
```

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

```

[+] 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.
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**

```
[+] Enumerating All Plugins (via Passive Methods)
[+] Checking Plugin Versions (via Passive and Aggressive Methods)

[!] Plugin(s) Identified:

[+] *
| Location: http://10.13.83.137/wp-content/plugins/*
|
| Found By: Urls In 404 Page (Passive Detection)
|
| The version could not be determined.

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

[!] No Config Backups Found.

[+] Performing password attack on Xmlrpc against 1 user/s
Trying wordpress-user / starwars Time: 00:00:03 → (262 / 262) 100.00% Time: 00:00:03

[!] No Valid Passwords Found.

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

[+] Finished: Sat Feb 7 19:05:02 2026
[+] Requests Done: 447
[+] Cached Requests: 6
[+] Data Sent: 186.183 KB
[+] Data Received: 835.738 KB
[+] Memory used: 268.777 MB
[+] Elapsed time: 00:00:08
```

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

```
GNU nano 7.2          /etc/mysql/mariadb.conf.d/50-server.cnf *
# this is only for the mysqld standalone daemon
[mysqld]
port = 80
#
# * Basic Settings
#

#user          = mysql
pid-file      = /run/mysqld/mysqld.pid
basedir       = /usr
#datadir      = /var/lib/mysql
#tmpdir        = /tmp

# Broken reverse DNS slows down connections considerably and name resolve is
# safe to skip if there are no "host by domain name" access grants
#skip-name-resolve

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address   = 0.0.0.0
```

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

```
debian@debian:~$ sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
debian@debian:~$ sudo systemctl restart mariadb
debian@debian:~$ ss -nltp | grep 8080
LISTEN 0      80          0.0.0.0:8080      0.0.0.0:*
debian@debian:~$
```

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

```
(john@JOHN)-[~]
$ sudo nmap -sV -Pn -p 80 192.168.1.165
[sudo] contraseña para john:
Starting Nmap 7.98 ( https://nmap.org ) at 2026-02-11 20:01 +0100
Nmap scan report for 192.168.1.165
Host is up (0.0026s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.62 ((Debian))
MAC Address: 08:00:27:2F:E9:42 (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.54 seconds

(john@JOHN)-[~]
$ sudo nmap -sV -Pn -p 8080 192.168.1.165
[sudo] contraseña para john:
Starting Nmap 7.98 ( https://nmap.org ) at 2026-02-11 20:27 +0100
Nmap scan report for 192.168.1.165
Host is up (0.00061s latency).

PORT      STATE SERVICE VERSION
8080/tcp  open  mysql   MariaDB 10.3.23 or earlier (unauthorized)
MAC Address: 08:00:27:2F:E9:42 (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.95 seconds
```

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

```
Performing additional rootkit checks
  Suckit Rootkit additional checks [ OK ]
  Checking for possible rootkit files and directories [ None found ]
  Checking for possible rootkit strings [ None found ]

Performing malware checks
  Checking running processes for suspicious files [ None found ]
  Checking for login backdoors [ None found ]
  Checking for sniffer log files [ None found ]
  Checking for suspicious directories [ None found ]
  Checking for suspicious (large) shared memory segments [ Warning ]
  Checking for Apache backdoor [ Not found ]

Performing Linux specific checks
  Checking loaded kernel modules [ OK ]
  Checking kernel module names [ OK ]

Checking the network...

Performing checks on the network ports
  Checking for backdoor ports [ None found ]

Performing checks on the network interfaces
  Checking for promiscuous interfaces [ None found ]
```

```
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)
```