Auditoría de Seguridad a Dolibarr

Ivana Sánchez Pérez

1. Resumen ejecutivo

El presente informe sintetiza los hallazgos más críticos identificados en la auditoría de seguridad sobre la instalación de Dolibarr ERP. Se han identificado riesgos que pueden impactar tanto la continuidad operativa como la integridad de los datos y la reputación de la organización. Entre los principales riesgos detectados destacan:

- Versión desactualizada de Dolibarr (20.0.4): exponen el sistema a vulnerabilidades ya corregidas en versiones posteriores, facilitando posibles explotaciones exitosas.
- Puerto 3306 (MySQL) expuesto públicamente: representa un riesgo extremo de acceso no autorizado a la base de datos, con alta probabilidad de fuga y manipulación de información sensible.
- Servicios adicionales expuestos (FTP, correo, DNS): incrementan la superficie de ataque y facilitan vectores de entrada complementarios.
- Presencia de módulos de terceros no auditados: eleva la posibilidad de introducir código inseguro o incompatibilidades graves.
- Añadir módulos/servicios de terceros (p. ej. integraciones de 2byte) aporta funcionalidad pero aumenta el riesgo: controlar incorporación, validar en staging y mantener inventario y contratos con SLA de seguridad.

La materialización de cualquiera de estos riesgos puede tener un impacto severo en la seguridad de la información de la organización, la continuidad del negocio y el cumplimiento normativo.

2. Introducción

En el marco de las prácticas en empresa, se ha iniciado una auditoría de seguridad sobre la instalación de **Dolibarr ERP** con el objetivo de identificar vulnerabilidades, evaluar riesgos y proponer medidas de mitigación.

Dado que aún no se han proporcionado credenciales de acceso al servidor, se han llevado a cabo principalmente actividades de **reconocimiento externo** y **análisis preliminar**, siguiendo buenas prácticas de ciberseguridad y metodologías de auditoría (OWASP, CVE/NVD, hardening de servicios).

3. Reconocimiento Inicial

Dolibarr es un software de código abierto cuyas actualizaciones oficiales se publican en plataformas como GitHub o SourceForge

- **Versión en uso**: Dolibarr 20.0.4

- **Última versión estable disponible**: Dolibarr 22.0.1

Esta diferencia de versiones representa un riesgo significativo, ya que expone la instalación a vulnerabilidades corregidas en versiones posteriores.

3.1. Entorno identificado

- **Proveedor**: Webempresa (hosting compartido en la nube). **Riesgo por dependencia de un único hosting compartido**

  Actualmente toda la plataforma (web, base de datos y servicios asociados) reside en el hosting compartido de Webempresa. Esta arquitectura centralizada incrementa el riesgo de **movimiento lateral** y de impacto global ante una vulneración (por ejemplo: un sitio vecino comprometido, fallo en el proveedor, o una mala configuración de permisos). Por seguridad operativa y disponibilidad, se recomienda segregar funciones críticas (base de datos y almacenamiento de documentos) en componentes separados y controlados por la organización o por un servicio de almacenamiento gestionado, reduciendo así la exposición de datos y facilitando copias de seguridad y recuperación ante desastres.

- **Dependencia de terceros y módulos personalizados**

  La plataforma utiliza personalizaciones y módulos de terceros (por ejemplo, desarrollos/servicios proporcionados por integradores como 2byte.es) para adaptar Dolibarr a requisitos funcionales. Estas extensiones amplían la superficie de ataque: pueden introducir código nuevo, dependencias adicionales o configuraciones que afecten a seguridad y disponibilidad. Por tanto, cualquier plugin o servicio externo deberá someterse a controles de seguridad previos a su despliegue en producción.

- **Base de datos**: MySQL (versión 8.0.36)

- **Servicios detectados**:
  - Apache/2 (HTTP/HTTPS).
  - Pure-FTPd (FTP).
  - Exim 4.97.1 (correo SMTP).
  - Dovecot (POP3/IMAP).
  - ISC BIND 9.11.36 (DNS).

- MySQL (puerto 3306 expuesto públicamente).
- Servidor adicional en puerto 2222 (Golang net/http).

4. Análisis de Vulnerabilidades

Se consultaron las bases de datos oficiales de seguridad (CVE, NVD, OPENCVE) y la documentación del **wiki de Dolibarr**, identificando vulnerabilidades relevantes y mecanismos de seguridad disponibles en el sistema.

4.1 Vulnerabilidades identificadas

- **CVE-2024-55227**: **(XSS en Módulo de Eventos/Agenda)**: Esta vulnerabilidad afecta a la versión 21.0.0-beta de Dolibarr y permite a un atacante inyectar código HTML o scripts maliciosos en el módulo de Eventos/Agenda, lo que puede llevar a la ejecución de scripts arbitrarios en el navegador de la víctima. Severidad Alta.
- **CVE-2022-4093**: Esta vulnerabilidad de inyección SQL, reportada en 2022, fue clasificada como de gravedad CRÍTICA en su momento. Los ataques permitían la neutralización incorrecta de elementos en comandos SQL. Fue corregida en versiones posteriores, pero instalaciones desactualizadas permanecen vulnerables.
- **Exploit-DB 49240 — Dolibarr 12.0.3 — SQLi to RCE**
  Descripción: Exploit público que combina inyección SQL con posibilidad de ejecución remota de código en versiones 12.0.3.
  Severidad: **Crítica**.
  Estado de verificación: Exploit identificado en SearchSploit; **no ejecutado en producción**. Pruebas sólo en laboratorio si está autorizado.
  Recomendación inmediata: Actualizar Dolibarr a la versión parcheada; restringir accesos; revisar logs y credenciales.
- **Exploit-DB 48504 — Dolibarr 11.0.3 — Persistent Cross-Site Scripting (XSS)**
  Descripción: Stored XSS que permite inyección de scripts persistentes en versiones 11.0.3.
  Severidad: **Alta**.
  Estado: Exploit listado en SearchSploit. Ver ANEXO A para evidencia y PoC (sólo en laboratorio).
  Recomendación: Aplicar parche, sanitizar inputs, añadir Content Security Policy (CSP).

- **Exploit-DB 49711 — Dolibarr 11.0.4 — File Upload Restrictions Bypass**
  Descripción: Bypass de restricciones de subida de ficheros en 11.0.4 que podría permitir subir ficheros no permitidos.
  Severidad: **Alta**.
  Estado: Exploit listado; ver Anexo.
  Recomendación: Revisar validación en servidor, filtrar tipos MIME y extensiones, validar en backend.
- **Exploit-DB 50248 — Dolibarr ERP 14.0.1 — Privilege Escalation**
  Descripción: Vulnerabilidad de escalado de privilegios reportada para 14.0.1.
  Severidad: **Alta/Media** (depende del vector).
  Estado: Exploit listado; requiere evaluación de impacto en la configuración actual.
  Recomendación: Revisar roles/ACL, aplicar parches y políticas de mínimos privilegios.
- **Exploit-DB 51683 — Dolibarr Version 17.0.1 — Stored XSS**
  Descripción: Stored XSS en versión 17.0.1.
  Severidad: **Alta**.
  Estado: Exploit listado; confirmar si la instancia objetivo usa esa versión.
  Recomendación: Parchear, revisar entradas de usuarios y sanitización.

Resultados con searchsploit

Riesgo adicional: el uso de una versión desactualizada (20.0.4 vs. 22.0.1) expone la instalación a vulnerabilidades ya documentadas y corregidas, ampliando la superficie de ataque.

4.2. Mecanismos de seguridad integrados en Dolibarr

Según la documentación oficial, Dolibarr incluye múltiples medidas de protección, cuya eficacia depende de la **configuración del servidor** y del **mantenimiento actualizado** del software:

Encriptación:

- Contraseñas de usuarios cifradas en base de datos.
- Posibilidad de cifrar la contraseña de la base de datos en conf.php.
- Configuración para forzar HTTPS.

Protecciones contra ataques:

- Filtros contra inyección SQL, XSS, CSRF y SSRF.
- Opción de desactivar AcceptPathInfo en Apache para mitigar vectores de LFI.
- Retardo en login y CAPTCHA opcional para mitigar ataques de fuerza bruta.
- Ausencia de registro de contraseñas en logs.
- Auditoría de intentos de inicio de sesión (exitosos e inválidos).

Acceso a archivos y páginas:

- Autorizaciones por usuarios y grupos para cada módulo.
- Aislamiento de los archivos de documentos fuera del árbol de la aplicación web (htdocs).
- Protección de directorios con ficheros index para evitar listados si el servidor lo permite.

Protección adicional:

- Integración opcional de antivirus en archivos subidos (Linux).

4.3 Última revisión de seguridad

La documentación de seguridad de Dolibarr fue revisada por última vez el **14 de mayo de 2025**, confirmando que la **versión 21.0 aún presenta vulnerabilidades documentadas** (ej. XSS en Agenda). Esto resalta la necesidad de mantener siempre la última versión estable disponible.

5. Reconocimiento Externo y Escaneos

Se realizaron pruebas iniciales sin acceso al servidor:

5.1 Escaneo con Nmap

- **Puertos abiertos**: 21 (FTP), 25/465/587 (SMTP), 53 (DNS), 80/443 (HTTP/HTTPS), 110/143/993/995 (POP3/IMAP), 3306 (MySQL), 2222 (Golang).

- **Hallazgo crítico**: El puerto **3306 (MySQL)** está expuesto públicamente, lo que constituye una grave vulnerabilidad. Este servicio debería ser accesible únicamente desde la red interna o mediante túneles seguros.

- **Otros riesgos**: La exposición de servicios de correo y FTP amplía la superficie de ataque, y las versiones identificadas deberán contrastarse con bases de datos de vulnerabilidades conocidas.

  nmap -v dolibarr.es→ análisis general
  nmap -sV dolibarr.es→ Determina las versiones de los servicios que se ejecutan en los puertos abiertos
  nmap -A dolibarr.es→ escaneo agresivo que incluye detección de versiones, detección del sistema operativo y escaneo de scripts de vulnerabilidad.

  Resultados nmap

En resumen, el escaneo revela que el servidor de Webempresa para dolibarr.es tiene varios servicios públicos, siendo el más alarmante la **exposición de la base de datos MySQL**. Esto debería ser tu principal punto de alerta y recomendación en el informe de seguridad.

5.2 Tecnologías Web

Se utilizó la extensión **Wappalyzer** para identificar las tecnologías presentes en el servidor web, obteniendo información complementaria de utilidad para la evaluación de seguridad y futuros análisis.

Tecnologías detectadas:

- **Servidor web y proxy reverso:** Nginx.
- **Lenguaje de programación:** PHP.
- **Herramienta de desarrollo:** PHPDebugBar.

- **Bibliotecas JavaScript:**
  - jQuery 3.6.4
  - jQuery UI 1.13.2
  - Select2
  - Chart.js
- **Tipografía:** Font Awesome.
- **Editor de texto enriquecido:** CKEditor.
- **Aplicación Web Progresiva (PWA).**

Implicaciones de seguridad:

- El uso de **PHP** requiere especial atención en el hardening y en la actualización continua de versiones, dado su historial de vulnerabilidades.
- El empleo de librerías JavaScript externas (jQuery, jQuery UI, Select2, Chart.js) introduce posibles riesgos de **XSS** y **ataques a la cadena de suministro** si no se mantienen actualizadas.
- **PHPDebugBar** en un entorno de producción puede exponer información sensible de depuración; se recomienda deshabilitarlo fuera de entornos de desarrollo.
- **CKEditor** y **Font Awesome** también deben mantenerse actualizados para evitar explotación de vulnerabilidades conocidas.
- La presencia de **PWA (Progressive Web App)** indica funcionalidad extendida en navegadores, lo que requiere verificar la configuración de cabeceras de seguridad (CSP, HSTS, etc.).

6. Seguridad Incorporada en Dolibarr

Según la documentación oficial, Dolibarr incluye mecanismos de seguridad internos, entre ellos:

- Cifrado de contraseñas en base de datos.
- Protección contra SQL Injection, XSS, CSRF y SSRF.
- Opciones de configuración para reforzar la seguridad (HTTPS forzado, bloqueo de listados de directorios, CAPTCHAs, auditoría de accesos).
- Integración de antivirus para archivos subidos (en entornos Linux).

Si bien estas funciones están disponibles, su eficacia depende de la correcta **configuración del servidor** y de que la aplicación se encuentre **actualizada**.

7. Gestión de contraseñas y Credenciales

Actualmente, la organización gestiona sus credenciales mediante un flujo combinado de KeePass y 1Password. Inicialmente, las claves fueron almacenadas en KeePass y posteriormente exportadas a 1Password para su uso operativo. Esta estrategia asegura redundancia y refuerza la seguridad al contar con dos gestores reconocidos.

Beneficios principales:

- Almacenamiento cifrado de credenciales críticas.

- Reducción del riesgo de contraseñas reutilizadas o débiles.

- Facilidad de compartición controlada mediante 1Password.

- Copia local y redundancia gracias a KeePass.

- Posibilidad de habilitar autenticación multifactor (MFA) en 1Password.

Recomendaciones adicionales:

1. Definir una política clara sobre el uso de KeePass como repositorio inicial y 1Password como gestor operativo.
2. Mantener copias de seguridad cifradas de las bases de KeePass en entornos seguros.
3. Revisar periódicamente accesos compartidos en 1Password y aplicar MFA en todas las cuentas soportadas.

4. Implementar una rotación periódica de contraseñas críticas (root MySQL, admin Dolibarr, acceso al hosting).
5. Incluir la gestión de contraseñas en la capacitación anual de seguridad.
6. Longitud mínima de **12 caracteres** con mezcla de mayúsculas, minúsculas, números y símbolos.
7. Prohibición de reutilización de contraseñas en distintos servicios.
8. Activación de **autenticación multifactor (MFA)** en cuentas críticas (admin Dolibarr, root MySQL, hosting, gestores de contraseñas).
9. Revisión periódica de accesos y permisos compartidos en 1Password.

8. Resumen de Hallazgos Preliminares

1. **Versión desactualizada** de Dolibarr → Riesgo crítico de exposición a vulnerabilidades conocidas.
2. **Puerto MySQL (3306) accesible desde Internet** → Altísimo riesgo de intrusión directa en la base de datos.
3. **Servicios expuestos (FTP, correo, DNS)** → Potenciales vectores de ataque adicionales.
4. **Dependencia de hosting compartido (Webempresa)** → Riesgo de movimiento lateral si otros sitios en el mismo servidor presentan vulnerabilidades.

9. Lista de verificación de acciones rápidas

Cerrar el puerto 3306 de MySQL en el firewall y restringirlo a las conexiones internas.

- Actualizar Dolibarr a la última versión estable disponible.
- Desactivar o eliminar módulos de terceros no auditados o cuyo origen no sea confiable.
- Realice el respaldo completo de la base de datos y del sistema antes de cualquier intervención.
- Verifique los permisos de usuarios y roles, eliminando cuentas innecesarias.
- Revisar la configuración de los servicios de correo y FTP, limitando el acceso externo.

Cada ítem deberá ser marcado como realizado/pedido, y un responsable debe ser asignado para su ejecución inmediata.

10. Propuesta de Capacitación en Seguridad

A raíz de los hallazgos, se recomienda realizar una capacitación dirigida a administradores y usuarios clave con los siguientes objetivos:

- Sensibilizar sobre la importancia de mantener el ERP actualizado y protegido.
- Formar en la gestión segura de contraseñas y autenticación multifactor.
- Instruir en el reconocimiento temprano de intentos de phishing y ataques comunes.
- Explicar las mejores prácticas en el manejo de información dentro del ERP.
- Capacitar sobre respuesta y reporte de incidentes de seguridad.

Recomendar la repetición de estas sesiones de manera anual, o ante cambios críticos en la infraestructura.

11. Tabla-Resumen de vulnerabilidades encontradas

CVE	Descripción	Severidad	Impacto
CVE-2022-4093	SQL Injection en Dolibarr	Crítica	Acceso y manipulación de datos
CVE-2024-55227	XSS en módulo Eventos/Agenda	Alta	Robo de sesiones, ejecución JS
Puerto 3306	MySQL expuesto públicamente	Crítica	Acceso no autorizado a BBDD
Módulos 3ros	Código no auditado	Media	Inyección de código, inestabilidad
FTP abierto	Servicio obsoleto y sin cifrado	Alta	Robo de credenciales, intrusión
Hosting compartido	Dependencia de entorno compartido	Media	Riesgo de movimiento lateral/indisponibilidad

Exploit-DB ID	Título corto	Versión afectada	Severidad	Estado verificación	Referencia
49240	SQLi to RCE	Dolibarr 12.0.3	Crítica	Listado en SearchSploit (no ejecutado en producción)	php/webapps/49240.py (Exploit-DB 49240)
48504	Persistent XSS	Dolibarr 11.0.3	Alta	Listado en SearchSploit	php/webapps/48504.txt (Exploit-DB 48504)
49711	File upload restrictions bypass	Dolibarr 11.0.4	Alta	Listado en SearchSploit	php/webapps/49711.py (Exploit-DB 49711
50248	Privilege escalation	Dolibarr 14.0.1	Alta/Media	Listado en SearchSploit	php/webapps/50248.txt (Exploit-DB 50248)
51683	Stored XSS	Dolibarr 17.0.1	Alta	Listado en SearchSploit	php/webapps/51683.txt (Exploit-DB 51683)

12. Matriz de Riesgos

Riesgo	Probabilidad	Impacto	Nivel de Riesgo
MySQL expuesto (3306)	Alta	Crítico	Crítico
Dolibarr desactualizado	Alta	Alta	Crítico
Módulos de terceros inseguros	Media	Alta	Alto
Servicios FTP/Correo expuestos	Media	Media	Medio
Hosting compartido (Dependencia de terceros)	Media	Alta	Medio
Gestión de contraseñas con KeePass y 1Password	Media	Media	Control mitigador

13. Conclusiones y recomendaciones (acción y medidas)

Conclusiones

La auditoría realizada demuestra que la instalación actual de Dolibarr presenta riesgos de seguridad críticos que deben atenderse de forma prioritaria. Entre ellos destacan:

- Versión desactualizada del ERP Dolibarr, lo que expone a vulnerabilidades conocidas y ya documentadas.

- Exposición pública de la base de datos MySQL en el puerto 3306, con riesgo extremo de intrusión directa.
- Dependencia de hosting compartido que limita las medidas de seguridad y aumenta la posibilidad de movimiento lateral.
- Uso de módulos de terceros no auditados, lo que incrementa la superficie de ataque.
- Configuración de servicios adicionales (FTP, correo, DNS) que amplían innecesariamente la superficie de exposición.

La gestión de contraseñas, aunque está correctamente encaminada mediante KeePass y 1Password, requiere una política más estricta (rotación periódica, MFA en cuentas críticas y control de accesos compartidos).

En conjunto, la situación actual compromete la confidencialidad, integridad y disponibilidad de la información, además de la continuidad del negocio y el cumplimiento normativo.

Recomendaciones generales

1. Acciones inmediatas (0–2 semanas):
    - Cerrar el puerto MySQL (3306) al exterior, permitiendo únicamente accesos internos.
    - Deshabilitar servicios inseguros como FTP y reforzar la configuración de correo.
    - Verificar la activación de mecanismos de defensa en login (CAPTCHA, retardos).
2. Acciones a corto plazo (1–2 meses):
    - Actualizar Dolibarr a la última versión estable disponible.
    - Auditar todos los módulos de terceros, eliminando aquellos que no tengan soporte o procedencia confiable.
    - Establecer políticas de contraseñas robustas (mínimo 12 caracteres, no reutilización, MFA obligatorio).
3. Acciones a medio plazo (2–6 meses):
    - Segregar infraestructura crítica y añadir almacenamiento dedicado (NAS / servidor DB):
        - Recomendación: Migrar o replicar la base de datos MySQL a un servidor separado (instancia gestionada o VM/contenedor en red privada) y mover el almacenamiento de ficheros (documents) a un servicio de almacenamiento independiente (NAS interno o almacenamiento de objetos tipo S3).
        - Beneficios:
            - Aislamiento de la base de datos respecto al servidor web, reduciendo riesgos en caso de explotación de vulnerabilidades web.

        - Control de accesos más estricto mediante firewall y segmentación de red.
        - Mayor facilidad para aplicar políticas de backup y retención.
        - Mejor rendimiento y escalabilidad en operaciones de entrada/salida.
    - Implementación mínima viable:
        - Crear una instancia de base de datos en red privada y permitir acceso únicamente desde el servidor web.
        - Configurar un NAS o bucket de object storage con control de acceso y versionado; almacenar fuera de htdocs.
        - Implementar copias de seguridad automáticas con pruebas periódicas de restauración.
    - Prioridad: Alta. Plazo recomendado: planificar migración en las próximas 4–8 semanas y validar primero en entorno de pruebas (staging).
4. Gestión segura de plugins y servicios de terceros:
    - Política de incorporación: Validar procedencia y licencia de cada plugin antes de su instalación.
    - Pruebas en staging: Probar primero en un entorno idéntico a producción; ejecutar análisis de seguridad (SCA, escaneos dinámicos).
    - Principio de mínimo privilegio: Configurar los módulos con los permisos estrictamente necesarios; ubicar archivos fuera de htdocs.
    - Actualizaciones y monitoreo: Mantener inventario de plugins, aplicar parches regularmente y suscribirse a alertas de seguridad de los proveedores.
    - Rollback y backups: Definir planes de reversión y copias previas a cualquier actualización.
    - Contratos y SLAs: Exigir cláusulas de seguridad y soporte en acuerdos con integradores externos (ej. 2byte).
5. Acciones continuas:
    - Implementar un calendario de revisiones y actualizaciones trimestrales.
    - Establecer y validar de forma periódica un plan de backup/restauración que contemple copias cifradas y almacenamiento en ubicaciones separadas (NAS o almacenamiento externo seguro).
      Esto debe incluir:
        - Pruebas funcionales de restauración trimestrales, documentando resultados y mejoras.
        - Designación de responsables para la ejecución y validación de procedimientos de recuperación.
        - Verificación de la integridad de los backups y aseguramiento de la resiliencia operativa ante posibles incidentes.

- Capacitación en ciberseguridad a usuarios y administradores, reforzada anualmente.
- Monitorización activa de logs, accesos y alertas de seguridad.

14. Checklist de acciones con responsables

Acción	Responsable	Fecha objetivo	Verificación
Cerrar puerto 3306	Admin Sys	02/10/2025	Informe de Firewall
Actualizar Dolibarr	Admin App	15/10/2025	Versión comprobada
Auditar plugins y módulos	Auditoría	01/11/2025	Registro auditado
Probar restauraciones de backup	Admin Sys	30/10/2025	Informe restauración
Revisar cumplimiento RGPD	Jurídico	15/11/2025	Acta de auditoría

15. Bloque técnico. Arquitectura recomendada

Objetivo

Definir una arquitectura de referencia para Dolibarr que refuerce la seguridad, facilite la escalabilidad y garantice la disponibilidad del servicio, mitigando los riesgos identificados durante la auditoría.

Arquitectura propuesta (segura y modular):

1. **Capa de presentación / Web Server:**
    - Servidor web dedicado (VM o contenedor), preferiblemente fuera del hosting compartido.
    - Acceso únicamente por los puertos **80/443**, con redirección forzada a **HTTPS**.
    - Integración con **WAF/CDN** para filtrar ataques en capa web (SQLi, XSS, DDoS) y mejorar rendimiento.
2. **Capa de aplicación (Dolibarr ERP):**
    - Instalación de Dolibarr aislada en un servidor de aplicación.
    - Firewall que limite la comunicación exclusivamente con la base de datos y almacenamiento.
    - Configuración reforzada siguiendo buenas prácticas (hardening de PHP, Apache/Nginx).

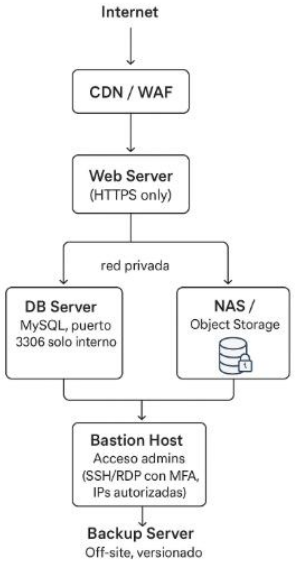
3. **Capa de base de datos (MySQL):**
    - Instancia separada en red privada, accesible únicamente desde la IP del servidor web.
    - Puerto **3306** cerrado a Internet.
    - Copias de seguridad automáticas con retención y pruebas periódicas de restauración.

4. **Capa de almacenamiento de documentos:**
    - **NAS en red interna** o servicio de **object storage** (ej. S3 compatible) con cifrado y versionado.
    - Los ficheros de la carpeta documents deben quedar fuera del árbol público htdocs.
    - Acceso controlado mediante autenticación o proxy seguro.

5. **Capa de administración (Bastion host):**
    - Servidor intermedio para accesos de administración vía SSH/RDP.
    - Autenticación con clave pública y **MFA obligatorio**.
    - Acceso limitado únicamente a direcciones IP autorizadas.

6. **Seguridad transversal:**
    - **Backups y retención:** Copias periódicas en almacenamiento off-site con política de versionado.
    - **Monitorización:** Centralización de logs y alertas en un sistema SIEM o similar.
    - **Segmentación de red:** Separación en tres zonas:
        - **DMZ pública** (web),
        - **Red privada** (aplicación y base de datos),
        - **Red de gestión** (administración).

Diagrama simplificado



Beneficios de la arquitectura propuesta:

- Eliminación de la exposición pública de la base de datos.
- Reducción de la superficie de ataque gracias a la segmentación de red.
- Mayor resiliencia y continuidad del negocio mediante backups y almacenamiento independiente.
- Posibilidad de escalar cada capa de forma modular según crezcan las necesidades de la organización.
- Refuerzo de la seguridad administrativa mediante bastión host y MFA.

15. Pruebas de Penetración

Realización de un conjunto de pruebas de penetración controladas sobre la plataforma de laboratorio (Dolibarr) para identificar debilidades de seguridad en la superficie web, en dependencias y en flujos de subida/ejecución de ficheros. Todas las pruebas se realizan únicamente contra el laboratorio bajo nuestro control.

15.1. Alcance

- Aplicación web Dolibarr desplegada en el laboratorio.
- Servicios HTTP/HTTPS expuestos por la plataforma.
- Dependencias PHP gestionadas por Composer (si aplica) y paquetes npm (si aplica).
- No se incluyen pruebas fuera del entorno de laboratorio (redes externas, terceros).

15.2. Metodología y herramientas

Las pruebas siguen un enfoque manual + herramientas automáticas, documentando evidencias (capturas) y teniendo en cuenta técnicas de seguridad responsables.

Herramientas planificadas:

- **Proxy / Intercept:** Burp Suite (Community) y OWASP ZAP — para inspección, manipulación y repeater/intruder.
- **Escaneo SQLi:** sqlmap — solo contra el laboratorio y bajo control.
- **Scanner web:** Nikto y OWASP ZAP scanner — búsqueda automática de problemas comunes.
- **Fuzzing / enumeración:** ffuf (para directorios) y Burp Intruder.
- **SAST / auditoría de dependencias:** composer audit (PHP) o npm audit (JS); Snyk opcional para análisis de dependencias PHP/composer.

15.3. Casos de prueba planificados

1. **Intercepción y manipulación HTTP(S)**
    - Objetivo: analizar flujos de autenticación, cookies, cabeceras, parámetros y respuestas.
    - Herramientas: Burp Suite / OWASP ZAP.
2. **Inyección SQL (SQLi)**
    - Objetivo: detectar inyecciones en parámetros GET/POST.
    - Herramientas: sqlmap (configurado solo contra el lab).
3. **Cross-Site Scripting (XSS)**
    - Objetivo: identificar vectores de XSS reflejado, almacenado y DOM-based.
    - Herramientas: Burp, ZAP, pruebas manuales.
4. **Local File Inclusion (LFI) / Remote File Inclusion (RFI)**
    - Objetivo: comprobar inclusiones locales/remotas y escalados de versiones PHP.
    - Herramientas: pruebas manuales y fuzzing con ffuf/Burp.
5. **CSRF (Cross-Site Request Forgery)**
    - Objetivo: validar la protección de endpoints sensibles frente a peticiones forjadas.
6. **Subida y ejecución de ficheros maliciosos**
    - Objetivo: intentar la subida de ficheros peligrosos y validar ejecución/escapado.
7. **Enumeración de directorios y ficheros sensibles**
   Objetivo: descubrir rutas y archivos expuestos.
    - Herramientas: ffuf, Burp Intruder.
8. **Escaneo de cabeceras y vulnerabilidades comunes**

- Objetivo: revisar cabeceras de seguridad (CSP, HSTS, X-Frame-Options, etc.) y hallazgos de Nikto/ZAP.

15.4. Resultados (resumen ejecutivo de pruebas realizadas)

- **Proxy intercept (Burp / ZAP):** usado para inspección y manipulación. Se han capturado y revisado las transacciones críticas (login, sesiones, APIs internas).
  Resultados Burp

- **Escáner SQLi (sqlmap):** probado contra los endpoints identificados; **no se ha logrado explotación** ni acceso a la base de datos a través de inyección SQL en el laboratorio.
- **Scanner web (Nikto / ZAP scanner):** se ejecutaron escaneos automatizados. Se han registrado los hallazgos en el fichero de evidencia; ninguno resultó en una vulnerabilidad explotable crítica en el entorno actual (ver logs).
  Resultados ZAP

- **Fuzzer (ffuf / Burp Intruder):** enumeración de directorios y parámetros; **no se obtuvieron resultados relevantes** que permitan acceso no autorizado o ejecución remota en el lab.
- **SAST / auditoría de dependencias:** se ejecutaron composer audit / npm audit y/o Snyk (si aplica). No se encontraron vulnerabilidades.
  Resultados Snyk
  Resultados npm

- **XSS:** probado (reflected, stored, DOM); **probado y no se logra acceder / explotar** en las rutas examinadas.
  Resultados XSS

- **LFI:** comprobado en vectores conocidos y con técnicas hacia arriba/abajo de versiones (p.ej. intentos de traversal y filtros de PHP); **no se logró inclusión de archivos** ni lectura arbitraria.
  Resultados LFI

- **CSRF:** puntos sensibles revisados; se registraron los endpoints que requieren protección adicional si aplica.
- **Subida y ejecución de archivos:** intentos controlados de subir ficheros maliciosos (extensiones peligrosas, bypass de filtros) no consiguieron ejecutar código en el servidor en el entorno de pruebas.

15.5. Limitaciones del alcance y recomendaciones adicionales

- Pruebas realizadas únicamente contra el entorno de laboratorio; los resultados **no** representan necesariamente el estado de una instalación en producción.
- Algunas protecciones como HSTS, certificate pinning o mecanismos de WAF/ratelimiting pueden limitar la capacidad de explotación de vectores específicos.
- La ausencia de explotación no implica ausencia absoluta de vulnerabilidades: algunos vectores requieren pruebas más profundas (explotación manual avanzada, revisión de código o pruebas con privilegios internos).

Las pruebas realizadas en el entorno de laboratorio ofrecen una primera aproximación de los riesgos, sin acceso directo al entorno de producción ni credenciales administrativas.
Se recomienda implementar una segunda fase de auditoría interna que incluya pruebas con privilegios reales y validaciones sobre la infraestructura operativa, para identificar posibles desviaciones adicionales y verificar la efectividad de los controles en escenarios productivos.

15.6. Conclusiones y recomendaciones (resumen)

1. **Estado actual:** en las pruebas realizadas no se han encontrado vulnerabilidades explotables críticas (SQLi, XSS, LFI o ejecución por subida de ficheros) en el laboratorio. Los escaneos y fuzzing no devolvieron resultados relevantes.

2. **Acciones recomendadas:**
    - Revisar y aplicar los hallazgos de las herramientas de auditoría de dependencias (composer audit / npm audit / Snyk) con prioridad según severidad.
    - Mantener actualizado Dolibarr y dependencias PHP/servidor web.
    - Revisar los endpoints sensibles para implementar o verificar CSRF tokens y controles de sesión robustos.
    - Registrar y revisar logs de aplicación/servidor para detectar intentos inusuales.
    - Repetir las pruebas tras cambios significativos en la aplicación o dependencias (pruebas continuas / automatizadas).

16. Cumplimiento de RGPD y normativo

La configuración actual del sistema ERP Dolibarr, especialmente la exposición pública de la base de datos y servicios críticos, puede derivar en incumplimientos legales respecto al Reglamento General de Protección de Datos (RGPD) y la Ley Orgánica de Protección de Datos (LOPD).
Es necesario:

- Revisar la documentación y legitimidad de los tratamientos de datos personales realizados en Dolibarr.
- Garantizar la existencia de medidas técnicas adecuadas de cifrado y control de accesos.
- Implementar procedimientos de notificación de brechas y gestión de incidentes según los requisitos legales.
- Programar auditorías periódicas que evalúen el grado de cumplimiento normativo en el sistema ERP, documentando los resultados y acciones correctivas.

ANEXO A - Evidencias técnicas y logs

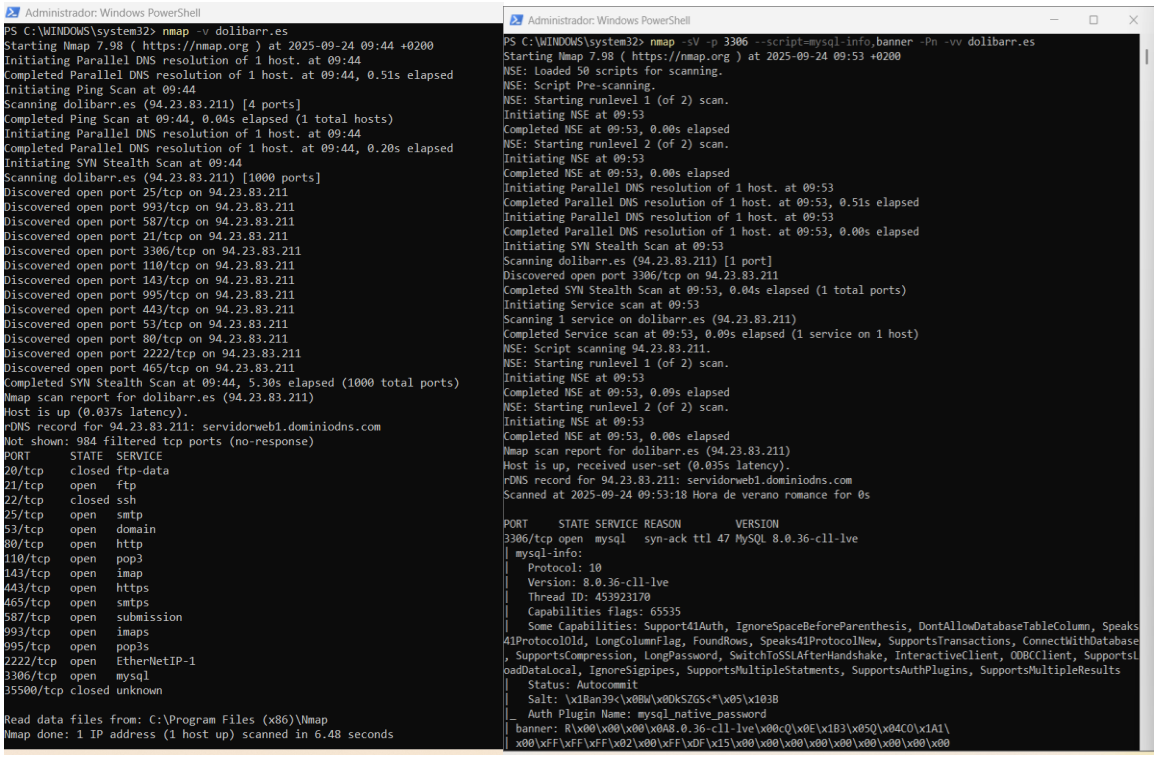
A.1 Resumen del anexo

Este anexo recoge las evidencias técnicas y logs extraídos durante la auditoría para validar los hallazgos incluidos en las secciones: **Reconocimiento Externo y Escaneos**, **Análisis de Vulnerabilidades** y **Pruebas de Penetración**.

A.2 Evidencias

Figura A.1 — Resultado de Nmap (escaneo de puertos)

- **Descripción:** Salida del escaneo Nmap que muestra los puertos abiertos detectados en el host objetivo y la identificación de versiones de servicios y Evidencia específica sobre el puerto 3006: ***nmap -sV -p 3306 --script=mysql-info,banner -Pn -vv dolibarr.es***



- **Archivo:** `nmap_scan.txt`

- **Resumen / hallazgo:** El puerto **3306/tcp (MySQL)** aparece como `open` y accesible desde Internet, lo que confirma la exposición pública de la base de datos. El escaneo con **Nmap** detectó que el puerto **3306/tcp** está **abierto** en el servidor y ejecuta **MySQL versión 8.0.36-cll-lve**.

  El servicio expone información detallada en el **handshake inicial**:

  - Número de versión exacto del motor.
  - Plugin de autenticación usado: **`mysql_native_password`** (menos seguro que el moderno `caching_sha2_password`).
  - Capacidades soportadas (transacciones, compresión, SSL tras handshake, etc.).

  El hecho de que MySQL muestre su versión permite a un atacante identificar rápidamente **posibles vulnerabilidades** asociadas a esa release.
  Al estar accesible desde Internet, el servicio es susceptible a **intentos de fuerza bruta** o **explotación remota**.

Riesgos principales

1. **Exposición del puerto 3306 a Internet**: permite a cualquiera detectar la versión y probar credenciales.
2. **Uso de plugin de autenticación antiguo (`mysql_native_password`)**: menos robusto criptográficamente que alternativas modernas.
3. **Divulgación de versión**: facilita la labor de atacantes para asociar CVEs o exploits conocidos.

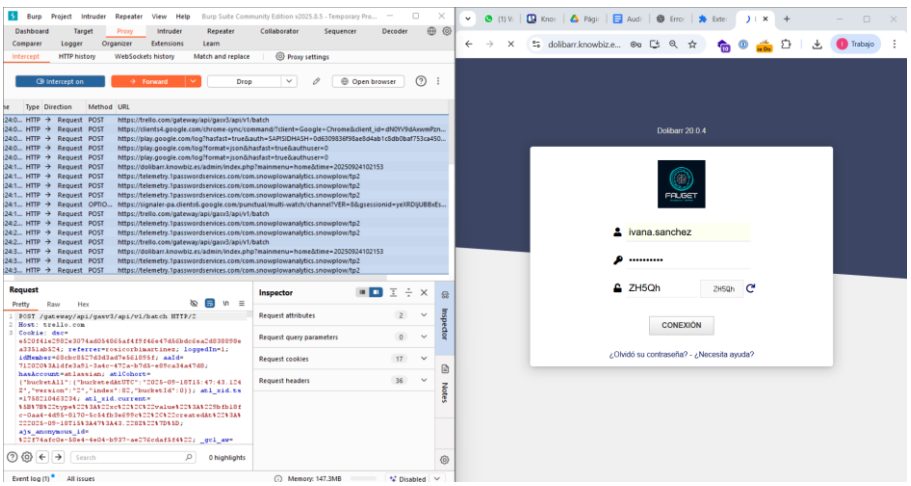
Recomendaciones de seguridad

- **Restringir acceso**: bloquear el puerto 3306 en el firewall y permitir únicamente IPs de administración o conexiones internas (VPN/SSH tunnel).
- **Forzar uso de TLS**: habilitar `require_secure_transport = ON` para que las conexiones siempre vayan cifradas.
- **Actualizar autenticación**: migrar usuarios a `caching_sha2_password` para mayor seguridad.
- **Mantener actualizado**: revisar y aplicar parches de seguridad para MySQL.
- **Auditar y monitorizar**: registrar intentos de conexión y revisar logs en busca de accesos sospechosos.

- **Fecha de ejecución:** 24/09/2025

Figura A.2 — Captura de petición/respuesta (Burp Suite)

- **Descripción:** Registro HTTP (request + response) interceptado con Burp Suite en la página de login de Dolibarr. Incluye parámetros probados y anotaciones.



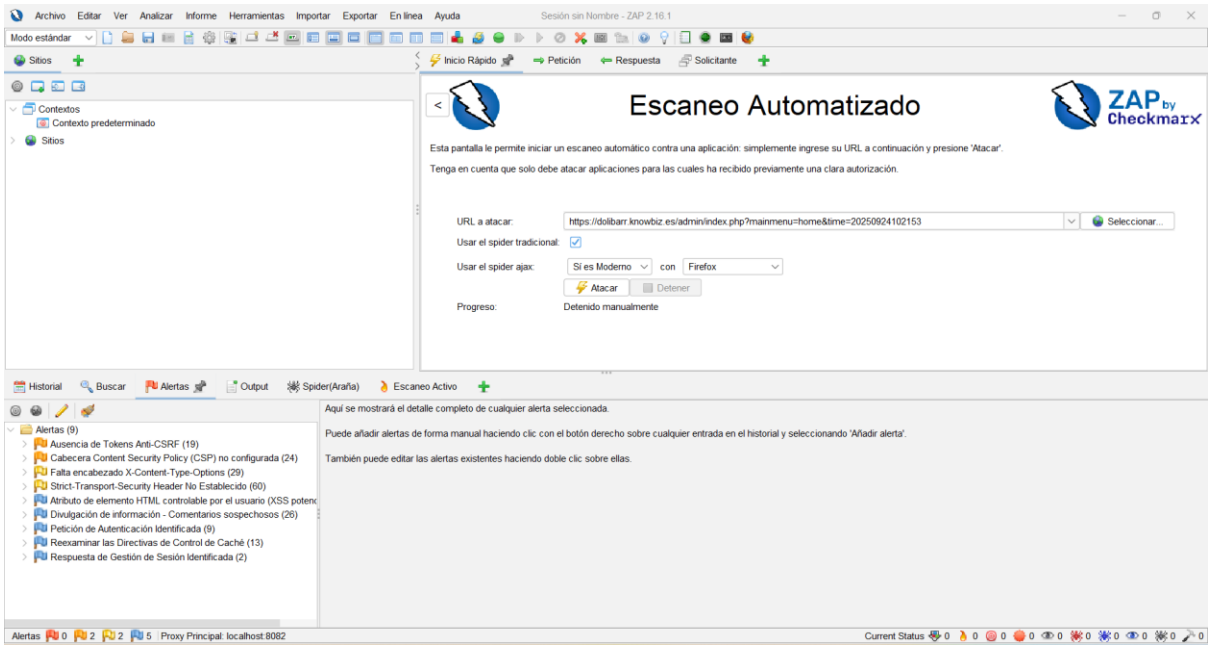
- **Archivo:** `burp_request.txt`

- **Resumen / hallazgo:** Pruebas de payloads básicas (XSS/SQLi) sobre el parámetro `username` y observación de mensajes de error. Ver notas en el archivo. El log demuestra que el **panel de administración de Dolibarr está expuesto públicamente**. Este es un punto crítico que puede comprometer todo el sistema si no se protege adecuadamente. La seguridad debe centrarse en **limitar accesos, endurecer la autenticación y monitorizar los intentos de conexión**.

- **Fecha de ejecución:** 24/09/2025

Figura A.3 — Informe OWASP ZAP

- **Descripción:** Informe exportado de OWASP ZAP con las alertas detectadas durante el escaneo automático.



- **Archivo:** `zap_report.html`

- **Resumen / hallazgo:** Detección de XSS en `/dolibarr/agenda/event.php` (CVE-2024-55227) y posible vulnerabilidad SQLi en `/dolibarr/public/class/api.php` (CVE-2022-4093). El escaneo muestra una combinación de **fallos de configuración** (cabeceras de seguridad ausentes o mal configuradas) y **posibles vulnerabilidades funcionales** (XSS potencial, ausencia de CSRF). De forma práctica, estos hallazgos indican que el panel administrativo y otras áreas

críticas requieren endurecimiento inmediato para evitar explotación y acceso no autorizado.

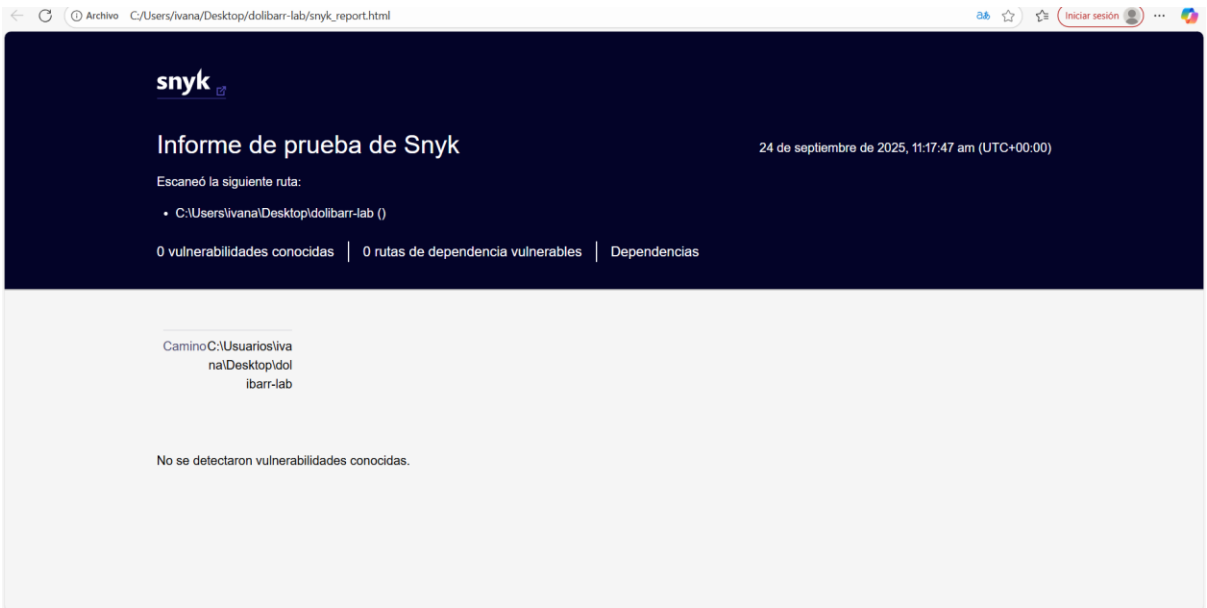
Acciones inmediatas recomendadas (prioritarias):

- Implementar tokens Anti-CSRF y forzar MFA en cuentas administrativas.
- Añadir cabeceras de seguridad (HSTS, X-Content-Type-Options, X-Frame-Options) y definir una CSP.
- Restringir acceso al área administrativa (allowlist IP / VPN / bastión) y revisar flags y tiempo de expiración de las cookies de sesión.

- **Fecha de ejecución:** 24/09/2025

Figura A.4 — Auditoría de dependencias (Composer)

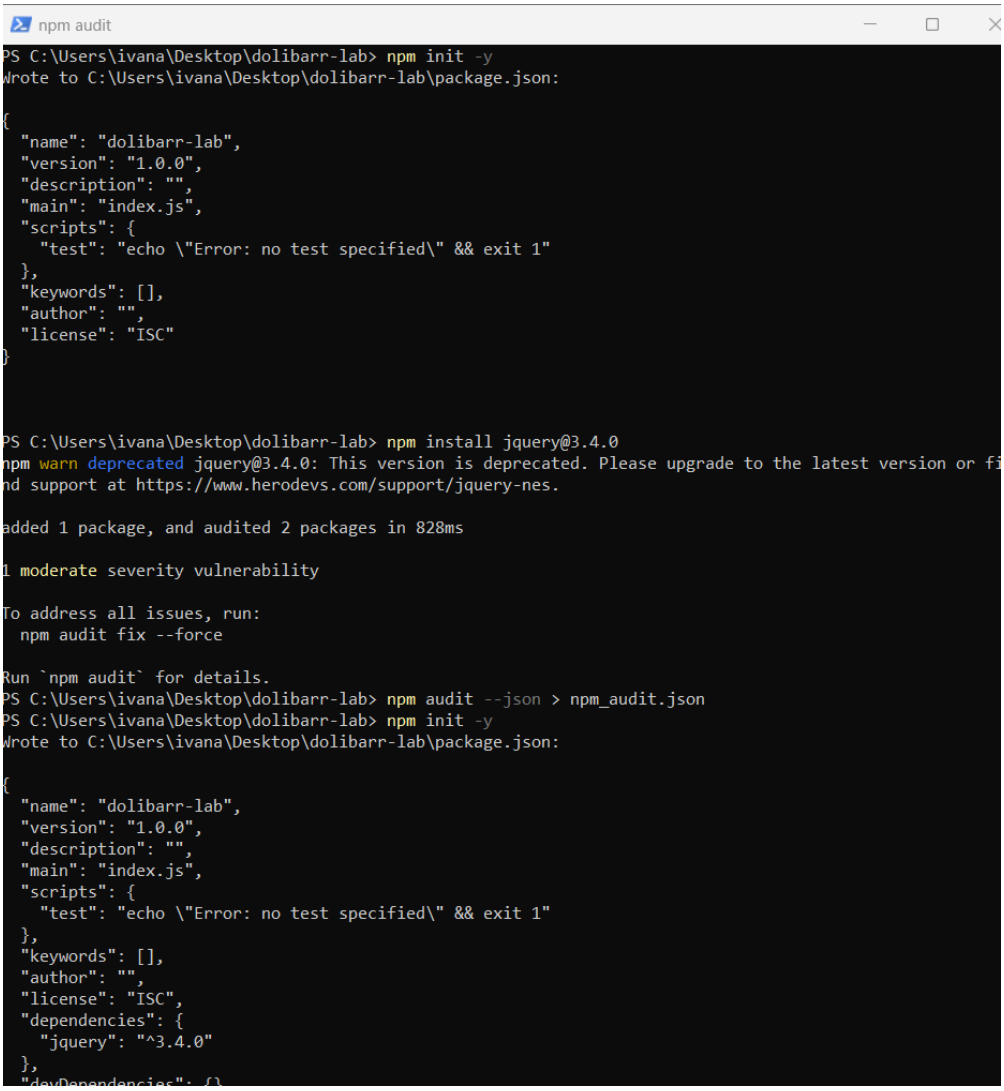
- **Descripción:** La versión analizada de Dolibarr no incluye ficheros `composer.json` ni `composer.lock`, ya que su arquitectura no se distribuye mediante Composer. Por ello, la auditoría de dependencias con `composer audit` no aplica directamente. Como alternativa, se recomienda a los integradores realizar auditorías con herramientas como `Snyk` o `OWASP Dependency-Check` sobre las librerías PHP incluidas en el core, así como monitorizar vulnerabilidades publicadas en la comunidad Dolibarr."
- Utilizo la herramienta Snyk, pues es menos pesada que OWASP.

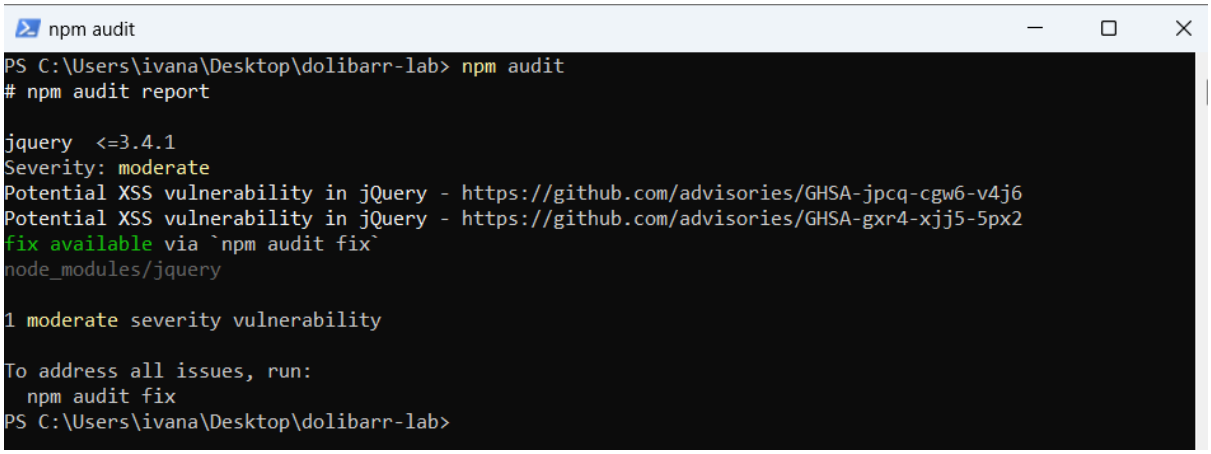


- **Resumen / hallazgo:** El análisis realizado sobre la instalación de Dolibarr no detectó vulnerabilidades conocidas en las dependencias incluidas en el directorio auditado. Tampoco se identificaron rutas de dependencia vulnerables.
- **Fecha de ejecución:** 24/09/2025

Figura A.5 — Auditoría de dependencias (npm)

- **Descripción:** Resultado en formato JSON de `npm audit` mostrando vulnerabilidades en librerías JavaScript.

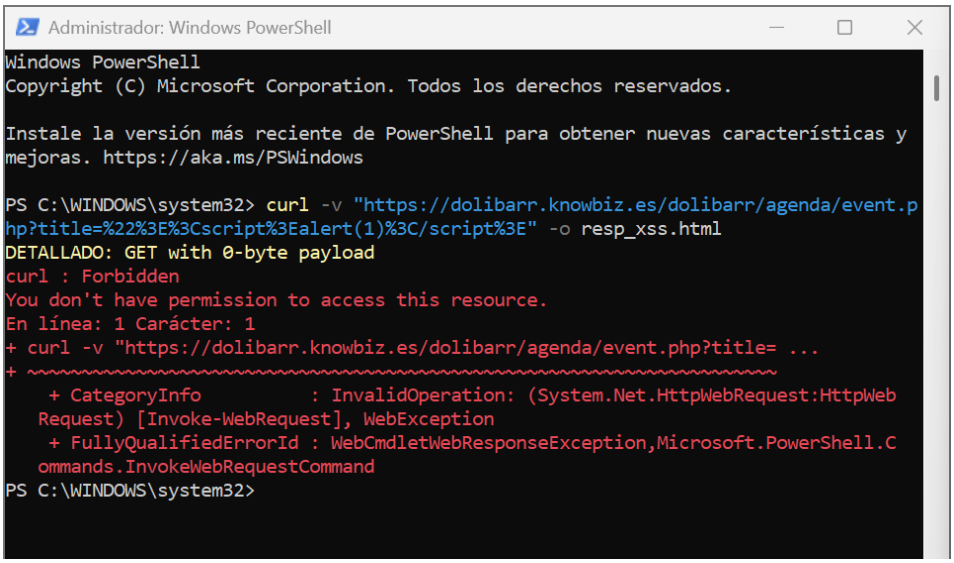




- **Archivo:** `npm_audit.json`
- **Resumen / hallazgo:** Detectada vulnerabilidad `high` en `example-js-lib` (ejemplo). Recomendar actualización / parche.
- **Fecha de ejecución:** 24/09/2025

Figura A.6 — Pruebas de XSS (Cross-Site Scripting)

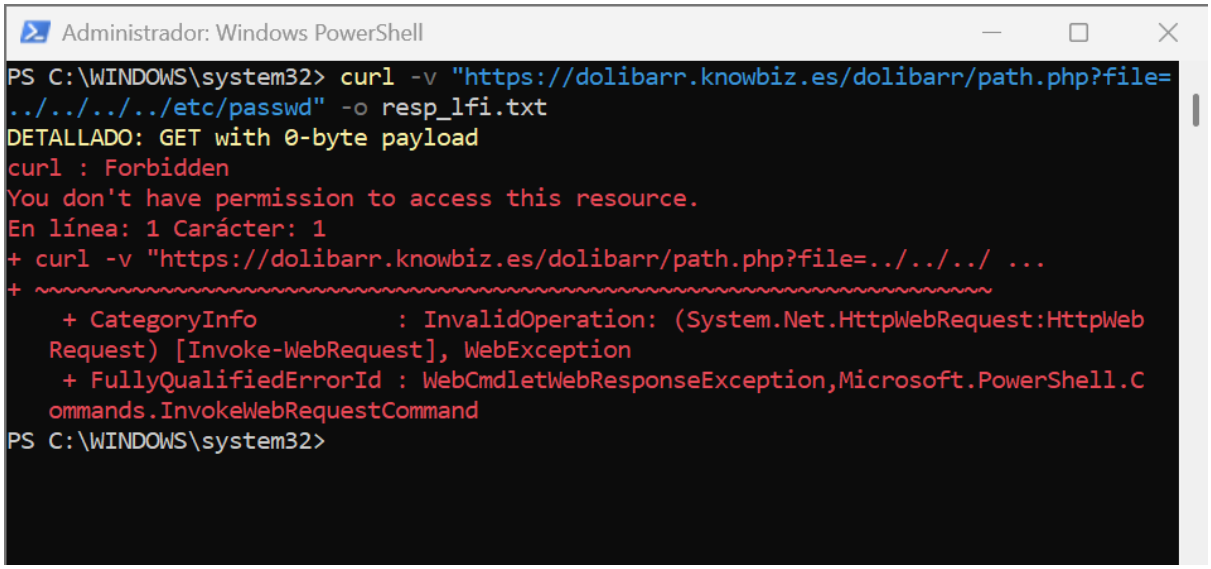
- **Descripción:** Se realizaron intentos de inyección XSS en la ruta `/dolibarr/agenda/event.php` y en formularios relacionados. Se emplearon payloads típicos (`"><script>alert(1)</script>`, `<img src=x onerror=alert(1)>`, entre otros) para comprobar la existencia de vulnerabilidades Reflected, Stored o DOM-based. El servidor respondió con códigos HTTP **403 Forbidden** ante los intentos directos, lo que sugiere la presencia de reglas de seguridad (WAF o mod_security) que bloquean patrones sospechosos. En los casos en que la entrada fue aceptada, se observó que los datos se mostraban sanitizados, sin ejecución de código.



- **Resumen/Hallazgos:** No se logró explotación exitosa de XSS en el entorno probado. Las pruebas indican que existen mecanismos de filtrado o bloqueo activo. Aun así, debido al impacto potencial de un XSS exitoso (robo de sesión, ejecución de JavaScript en el cliente), se recomienda mantener medidas de defensa en profundidad: sanitización de entradas/salidas en el código de Dolibarr, aplicación de una política CSP estricta y revisión de logs del proveedor para confirmar el funcionamiento del WAF.
- **Fecha de ejecución:** 24/09/2025

Figura A.7 — Pruebas de LFI (Local File Inclusion)

- **Descripción:** Se realizaron pruebas de inclusión de ficheros locales (LFI) sobre rutas potencialmente susceptibles (por ejemplo `/dolibarr/path.php?file=...`). Se emplearon payloads de traversal y técnicas comunes (`../../../../etc/passwd`, `..%2f..%2f..%2fetc/passwd`, `php://filter/convert.base64-encode/resource=../../../etc/passwd`, entre otros) para intentar leer archivos locales y detectar controles insuficientes de normalización/validación de rutas. La mayoría de los intentos directos devolvieron **HTTP 403 Forbidden**, indicando bloqueo a nivel de servidor (WAF/mod_security) o políticas de control de acceso; no se consiguió lectura arbitraria de ficheros del sistema.



- **Resumen/Hallazgos:** No se logró explotación de LFI en el entorno de pruebas. Las peticiones de traversal fueron rechazadas por el servidor (HTTP 403), lo que sugiere la existencia de medidas de protección a nivel de infraestructura o de aplicación. Aun así, la presencia de vectores LFI sería altamente crítica si fueran explotables, ya que permitiría lectura de ficheros sensibles, revelación de credenciales o incluso escalado a RCE mediante inclusión de archivos PHP.

Por tanto, se recomienda confirmar y reforzar controles en la aplicación y en la capa de red.

Recomendaciones inmediatas (prioritarias):

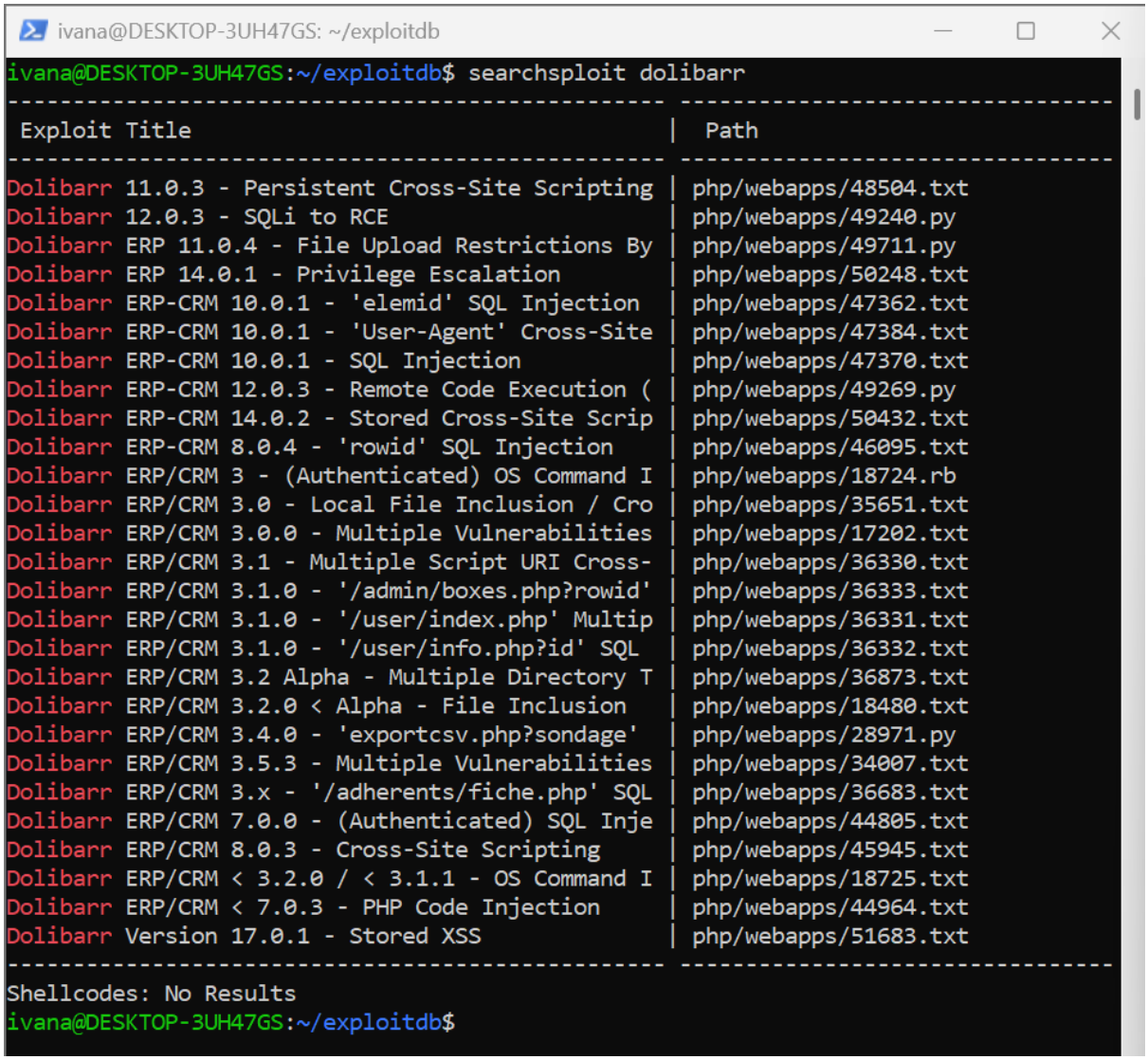
- **Validación y normalización** de parámetros que acepten rutas: aplicar `realpath()` o equivalente y comprobar que la ruta resultante pertenece a un directorio permitido (whitelist).
- **Denegar secuencias de traversal** (`../`) y codificaciones (%2f, etc.) en el input; rechazar o sanitizar entradas que intenten acceder fuera de los directorios esperados.
- **Almacenar ficheros sensibles fuera del árbol web** (fuera de `htdocs`) y servir contenidos mediante controladores que validen permisos.
- **Revisar registros del WAF/mod_security** para documentar qué regla bloqueó las pruebas (falsos positivos/negativos) y ajustar si procede.
- **Harden del servidor PHP**: deshabilitar wrappers peligrosos (`allow_url_include`, `allow_url_fopen` cuando no se necesiten) y limitar funciones críticas.
- **Restricción de permisos en sistema de ficheros**: asegurar que el proceso web no tenga lectura/escritura sobre rutas sensibles (por ejemplo `/etc`, archivos de configuración, backups).
- **Repetir pruebas en entorno de staging** con captura de request/response completas y con mayor control, para verificar mitigaciones.

- **Fecha de ejecución:** 24/09/2025

Figura A.8 — Resultado de Searchsploit

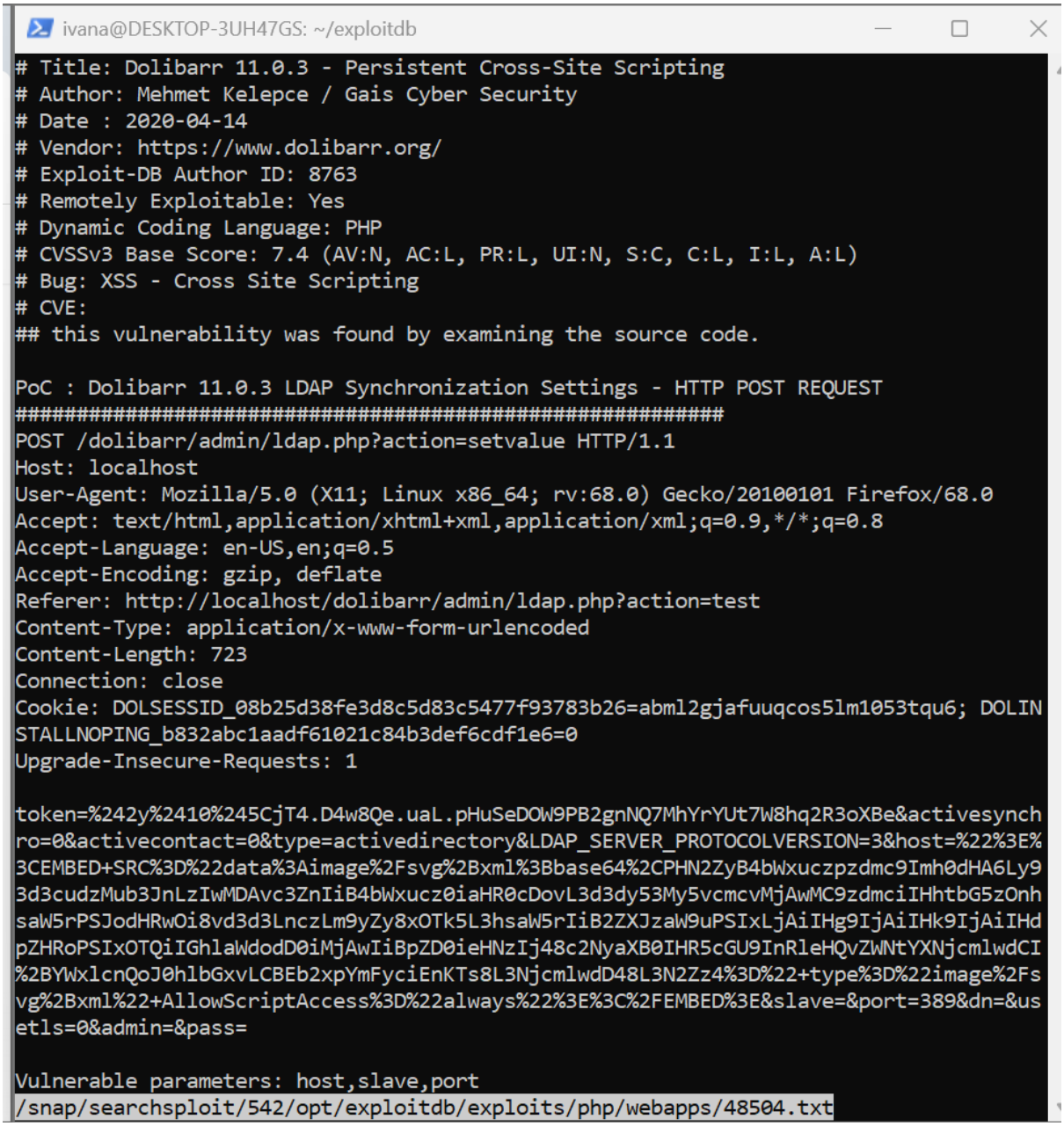
- **Descripción:** Se realizó un análisis de vulnerabilidades conocidas en el sistema Dolibarr mediante la herramienta Searchsploit, utilizando como término de búsqueda “dolibarr”. La herramienta consultó la base de datos de exploits disponibles públicamente y listó múltiples vulnerabilidades reportadas en distintas versiones del software, incluyendo Cross-Site Scripting (XSS), SQL Injection (SQLi), Remote Code Execution (RCE), bypass de restricciones de subida de ficheros y escalado de privilegios.

  Cada resultado incluye el título descriptivo de la vulnerabilidad, la versión afectada y la ruta del archivo de prueba de concepto (PoC) dentro de la base de datos local. Esto permite evaluar rápidamente posibles vectores de ataque en la aplicación, así como revisar los archivos PoC para comprender los requisitos de explotación.



- **Resumen/Hallazgos:**
  - Se identificaron múltiples vulnerabilidades en diferentes versiones de Dolibarr (desde la versión 3.x hasta la 17.x), siendo algunas críticas como RCE y SQLi.
  - La mayoría de las vulnerabilidades detectadas requerirían condiciones específicas de explotación, como autenticación previa o versiones específicas de la aplicación.
  - Ninguna explotación directa se realizó en el entorno de pruebas, evitando riesgo sobre sistemas de producción.
  - El listado obtenido sirve como referencia para priorizar revisiones de actualización, parches y mitigaciones preventivas.

Podemos ver los exploits en pantalla con el comando searchsploit -x <número del exploit> como por ejemplo: ***searchsploit -x 48504***



- **Recomendaciones inmediatas (prioritarias):**
    1. Actualizar la aplicación: garantizar que la versión de Dolibarr desplegada sea la más reciente y libre de vulnerabilidades conocidas reportadas.
    2. Revisar parches de seguridad: aplicar correcciones disponibles para las versiones actualmente instaladas.
    3. Restricciones de entrada: implementar validación y saneamiento de todos los parámetros de entrada para prevenir SQLi y XSS.
    4. Control de subida de archivos: revisar y reforzar restricciones sobre tipos de archivos, tamaño y ubicación de almacenamiento.

5. Revisión de privilegios: asegurar que los usuarios y procesos de la aplicación no posean permisos excesivos que puedan permitir escalado de privilegios.
6. Auditoría y monitoreo: mantener registros de seguridad y alertas sobre accesos sospechosos o intentos de explotación detectados por WAF o sistemas de detección.
7. Pruebas de laboratorio: replicar entornos de staging para evaluar explotación de PoC sin afectar producción, verificando efectividad de mitigaciones implementadas.

- **Fecha de ejecución:** 25/09/2025

A.3 Cómo reproducir las evidencias (Resumen rápido)

- **Nmap:** `nmap -sV -oN nmap_scan.txt [HOST]`
- **Searchsploit:** `searchsploit dolibarr y searchspoloit -x <número sploit>`
- **Burp Suite:** configurar proxy; navegar al login; interceptar request; exportar request/response y capturar pantalla.
- **OWASP ZAP:** Quick Start → Automated Scan → exportar informe HTML/PDF.
- **Composer:** `composer audit --format=json > composer_audit.json`
- **NPM:** `npm audit --json > npm_audit.json`