INFORME DE AUDITORÍA DE SEGURIDAD WEB

- **Aplicación:** WebGoat 8.1.0
- **Cliente:** Proyécto KeepCoding – Ciberseguridad 101
- **Entorno:** Local (Docker)
- **Auditor:** Sergio García
- **Fecha:** 18 de enero de 2026

ÍNDICE

1. Introduccion

El presente informe recoge los resultados de una auditoría de seguridad realizada sobre la aplicación web **WebGoat 8.1.0**, desplegada en un entorno local mediante contenedores Docker.
El objetivo principal de la auditoría ha sido identificar vulnerabilidades comunes en aplicaciones web, alineadas con el **OWASP Top 10**, y evaluar su impacto sobre la confidencialidad, integridad y disponibilidad del sistema.

Durante la auditoría se identificaron múltiples vulnerabilidades de **alto impacto**, incluyendo inyección SQL, Cross-Site Scripting (XSS), XML External Entity Injection (XXE) y el uso de componentes vulnerables y desactualizados. Estas debilidades permitirían a un atacante acceder a información sensible, ejecutar código malicioso en el navegador de los usuarios y comprometer la seguridad general de la aplicación.

Aunque el entorno auditado es formativo y controlado, las vulnerabilidades detectadas reflejan problemas habituales en aplicaciones reales. La correcta aplicación de buenas prácticas de desarrollo seguro permitiría mitigar eficazmente los riesgos identificados.

2. Alcance y Metodología

La auditoría se llevó a cabo bajo una **metodología white-box**, contando con conocimiento previo de la aplicación y sus módulos vulnerables, al tratarse de un entorno de aprendizaje intencionadamente inseguro.

El alcance de la auditoría incluyó:

- Análisis de la lógica de la aplicación
- Identificación y explotación controlada de vulnerabilidades OWASP Top 10
- Evaluación del impacto de cada vulnerabilidad
- Propuesta de medidas de mitigación

No se realizaron pruebas destructivas ni escaneos agresivos, dado que el entorno era local y controlado.

3. Herramientas Utilizadas

- **Navegador con DevTools:** Análisis de ejecución de código en cliente
- **Burp Suite:** Interceptación y modificación de peticiones HTTP
- **Docker:** Despliegue controlado del entorno vulnerable
- **Nmap:** Identificación de puertos y servicios expuestos
- **SQLMap:** Automatización de pruebas de inyección SQL

4. Hallazgos de Seguridad

4.0 KC-00 -- Reconocimiento / Information Gathering

Durante la fase de reconocimiento se identificó:

- **Puerto expuesto:** 8080 (se utilizó nmap -p 8080 para confirmar el servicio)
- **Sistema operativo:** Linux
- **Lenguaje y framework:** Java (Spring / Servlet)
- **Entorno:** Aplicación desplegada en contenedor Docker

No se realizaron escaneos agresivos al tratarse de un entorno local controlado.

Request

Pretty Raw Hex

```
1 GET /WebGoat/login?error HTTP/1.1
2 Host: localhost:8080
3 Cache-Control: max-age=0
4 Accept-Language: en-US,en;q=0.9
5 Upgrade-Insecure-Requests: 1
6 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36
7 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
8 Sec-Fetch-Site: same-origin
9 Sec-Fetch-Mode: navigate
10 Sec-Fetch-User: ?1
11 Sec-Fetch-Dest: document
12 sec-ch-ua: "Chromium";v="143", "Not A(Brand";v="24"
13 sec-ch-ua-mobile: ?0
14 sec-ch-ua-platform: "Linux"
15 Referer: http://localhost:8080/WebGoat/login
16 Accept-Encoding: gzip, deflate, br
17 Cookie: hijack_cookie=1168895929553397763-1766170910701; JSESSIONID=0A367B59F549C662A0AC799DB2E566AB
18 Connection: keep-alive
```

4.1 KC-01 – Inyección SQL en módulo de autenticación

Severidad: Crítica
OWASP: A03 – Injection

Descripción
Se identificó una vulnerabilidad de inyección SQL en el módulo *A3 Injection – SQL Injection (intro)*. La aplicación construye consultas SQL concatenando directamente los datos introducidos por el usuario.

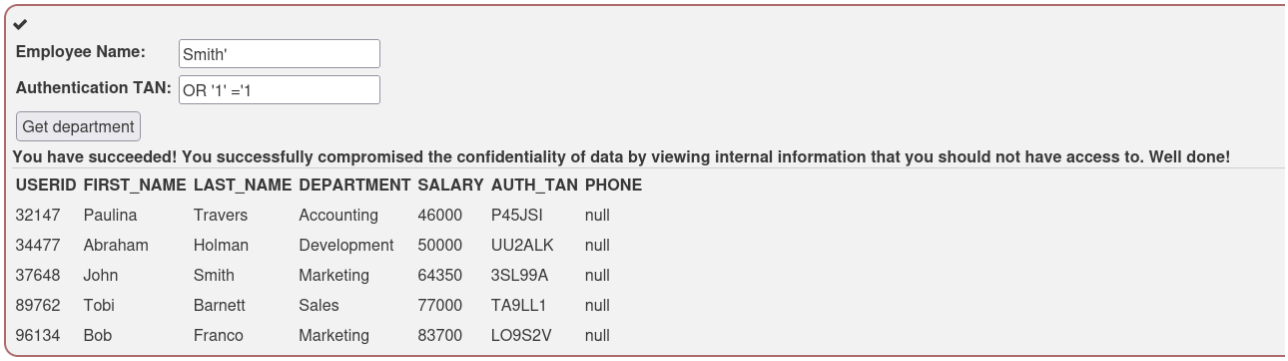
Impacto
Un atacante podría:

- Acceder a todos los registros de la base de datos
- Omitir controles de autenticación
- Comprometer la confidencialidad del sistema

Prueba de Concepto (PoC)
Ejemplo de payload utilizado:

Smith'

```
OR '1'='1
```



El payload fuerza que la condición SQL sea siempre verdadera, devolviendo todos los registros.

Recomendación
Implementar consultas parametrizadas (Prepared Statements) y evitar la concatenación directa de entradas del usuario.

Referencias

- https://owasp.org/www-community/attacks/SQL_Injection
- https://owasp.org/Top10/A03_2021-Injection/

***Payloads probados:**

- Employee Name: ' OR '1'='1
  Authentication TAN: ' OR '1'='1

- Employee Name: Smith' OR 1 = 1; --
  Authentication TAN: [1234]

- Employee Name: ' OR last_name IS NOT NULL --
  Authentication TAN: 1234
- Employee Name: ' OR '1'='1
  Authentication TAN: ' OR '1'='1
- Employee Name: Smith'
  Authentication TAN: ' OR '1'='1

Conclusión del hallazgo:
La vulnerabilidad permite el acceso no autorizado a información crítica de la base de datos y la evasión de mecanismos de autenticación, representando un riesgo severo para la confidencialidad e integridad del sistema.

4.2 KC-02 – Cross-Site Scripting (XSS) reflejado

Severidad: Alta
OWASP: A03 – Injection

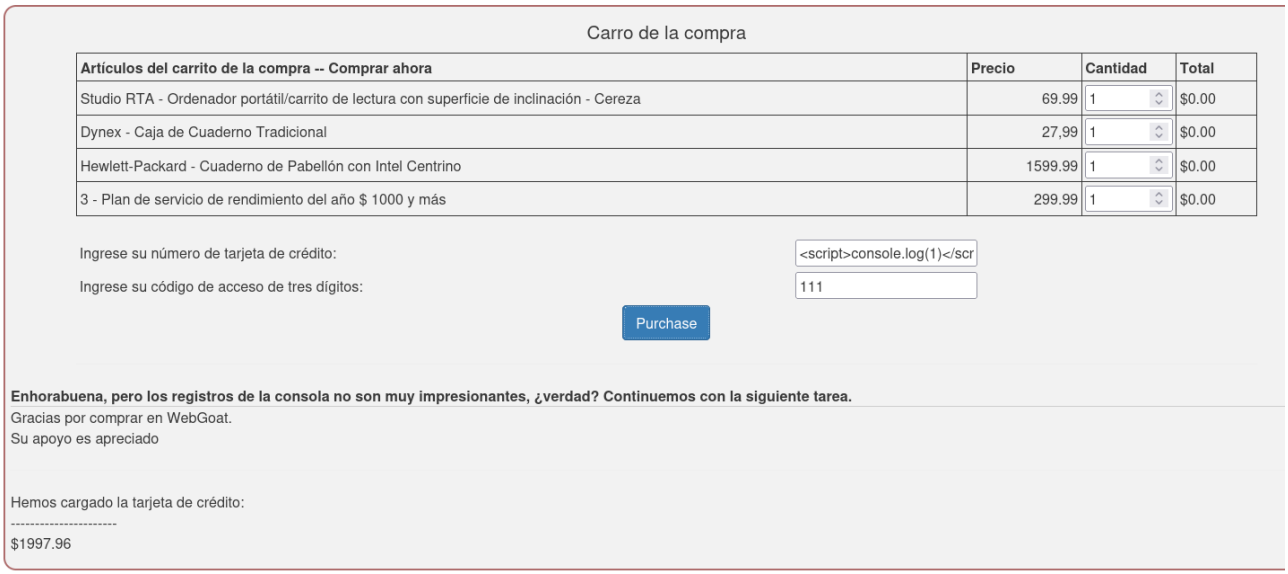
Descripción
Se detectó XSS reflejado en el módulo *Cross-Site Scripting (XSS)*, donde la aplicación renderiza directamente entradas del usuario sin validación ni escape.

Impacto
Permite:

- Ejecución de JavaScript arbitrario
- Robo de cookies de sesión
- Suplantación de identidad

Prueba de Concepto (PoC)

```
<script>alert(1)</script>
```



Carro de la compra

Artículos del carrito de la compra -- Comprar ahora	Precio	Cantidad	Total
Studio RTA - Ordenador portátil/carrito de lectura con superficie de inclinación - Cereza	69.99	1	$0.00
Dynex - Caja de Cuaderno Tradicional	27,99	1	$0.00
Hewlett-Packard - Cuaderno de Pabellón con Intel Centrino	1599.99	1	$0.00
3 - Plan de servicio de rendimiento del año $ 1000 y más	299.99	1	$0.00

Ingrese su número de tarjeta de crédito: <script>console.log(1)</scr

Ingrese su código de acceso de tres dígitos: 111

Purchase

Enhorabuena, pero los registros de la consola no son muy impresionantes, ¿verdad? Continuemos con la siguiente tarea.

Gracias por comprar en WebGoat.
Su apoyo es apreciado

Hemos cargado la tarjeta de crédito:

$1997.96

Recomendación
Validar entradas y aplicar codificación de salida (output encoding) antes de renderizar datos en el navegador.

Referencias

- https://owasp.org/www-community/attacks/xss/
- https://cheatsheetseries.owasp.org/cheatsheets/XSS_Prevention_Cheat_Sheet.html

Conclusión del hallazgo:
La ejecución de código JavaScript en el navegador del usuario puede derivar en el robo de sesiones, suplantación de identidad y manipulación de contenido, afectando directamente a la confidencialidad y confianza en la aplicación.

4.3 KC-03 – Security Misconfiguration (XXE)

Severidad: Crítica
OWASP: A05 – Security Misconfiguration

Descripción
La aplicación procesa XML permitiendo entidades externas, lo que posibilita ataques XXE.

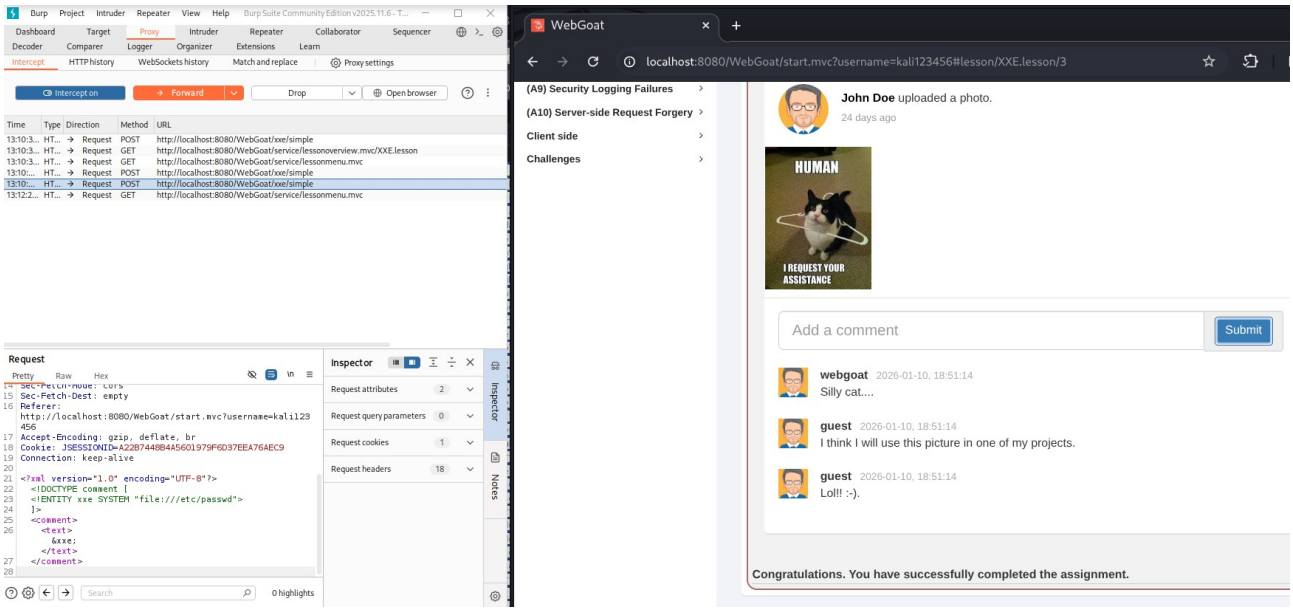
Impacto
Un atacante podría:

- Leer archivos del sistema
- Acceder a información sensible

- Realizar peticiones internas (SSRF)

Prueba de Concepto (PoC)

```
<!DOCTYPE comment [
 <!ENTITY xxe SYSTEM "file:///etc/passwd">]>
```



Recomendación
Deshabilitar entidades externas y DTD en el parser XML.

Referencias

- https://owasp.org/www-community/vulnerabilities/XML_External_Entity_(XXE)_Processing

Conclusión del hallazgo:
La vulnerabilidad XXE permite el acceso a archivos internos del sistema y a información sensible del servidor, lo que supone un riesgo crítico para la confidencialidad y puede facilitar ataques más avanzados contra la infraestructura.

4.4 KC-04 – Uso de componentes vulnerables y desactualizados

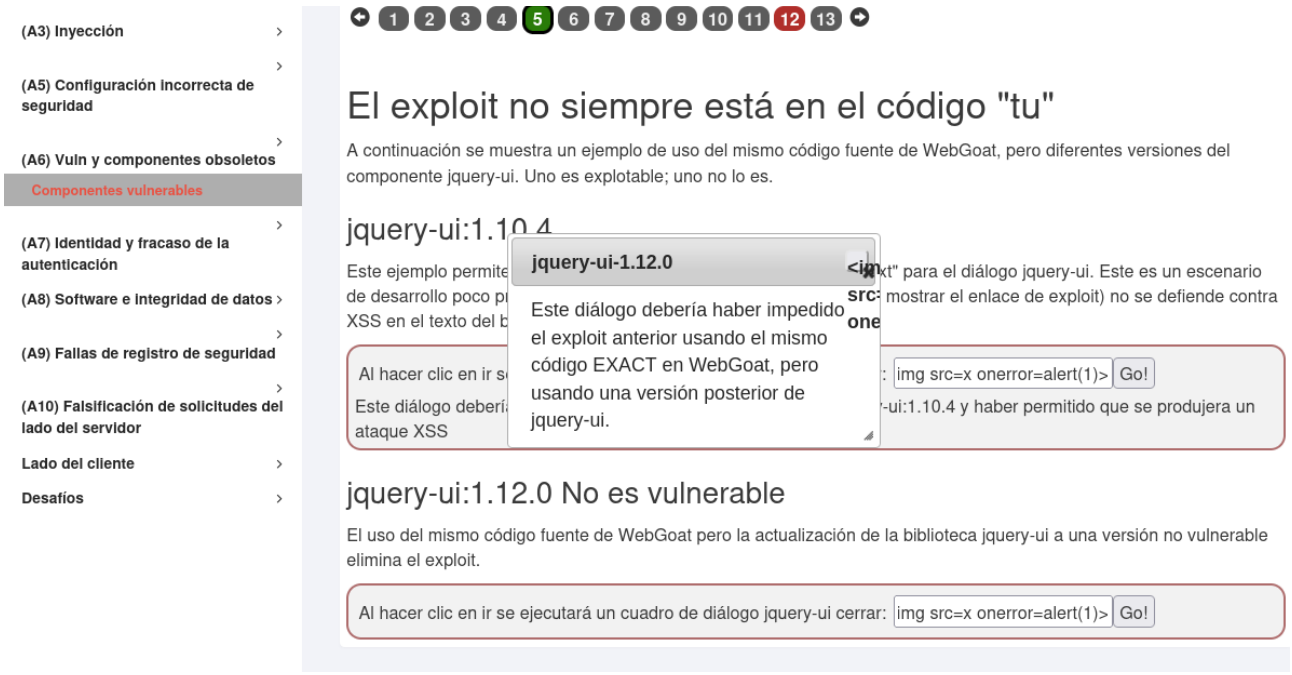
Severidad: Alta
OWASP: A06 – Vulnerable and Outdated Components

Descripción
La aplicación utiliza **jquery-ui 1.10.4**, versión vulnerable a XSS.

Impacto
Ejecución de JavaScript arbitrario en el navegador del usuario.

Prueba de Concepto (PoC)

<img src=x onerror=alert(1)>



Recomendación
Actualizar dependencias a versiones seguras y realizar análisis periódicos de dependencias (SCA).

Referencias

- https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/

Conclusión del hallazgo:
El uso de librerías externas vulnerables introduce riesgos indirectos que pueden permitir la ejecución de código malicioso en el navegador del usuario, comprometiendo la seguridad de la aplicación incluso cuando el código propio no presenta fallos.

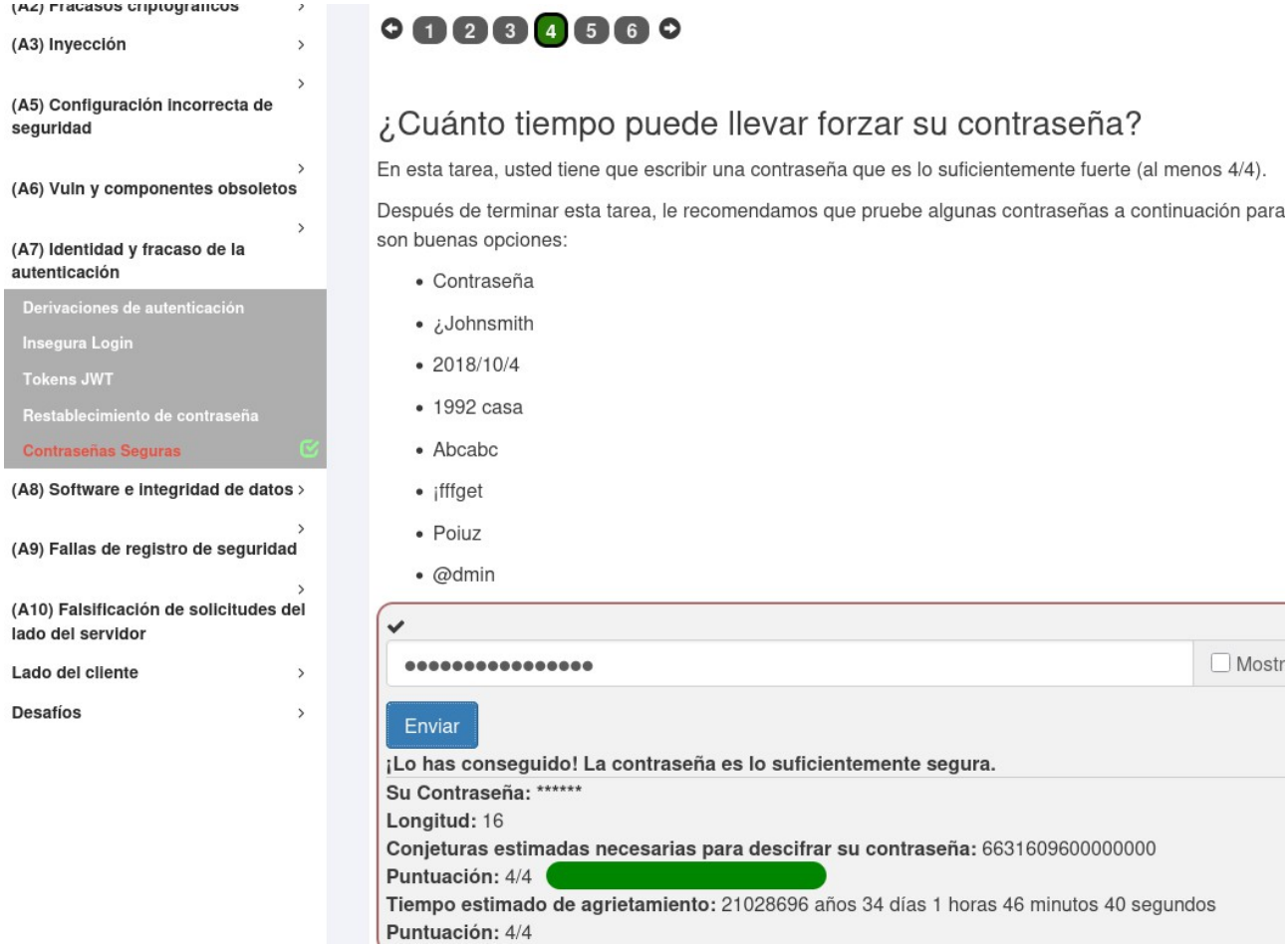
4.5 KC-05 – Fallos en autenticación y gestión de contraseñas

Severidad: Media
OWASP: A07 – Identification and Authentication Failures

Descripción
Se evidenció la importancia del uso de contraseñas robustas frente a ataques de fuerza bruta.

Impacto
Contraseñas débiles pueden permitir accesos no autorizados y comprometer cuentas de usuario.

Prueba de Concepto (PoC)



Recomendación

- Políticas de contraseñas robustas
- Uso de hashing seguro (bcrypt / Argon2)
- Protección frente a fuerza bruta

Referencias

- https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/

Conclusión del hallazgo:
La utilización de contraseñas débiles reduce significativamente la seguridad de las cuentas de usuario, facilitando ataques de fuerza bruta y accesos no autorizados que afectan a la confidencialidad y control de acceso del sistema.

4.6 Evaluación global de riesgos (OWASP)

A continuación, se presenta una tabla resumen de evaluación de riesgos basada en la metodología OWASP, donde se clasifican las vulnerabilidades detectadas según su severidad, probabilidad de explotación e impacto potencial sobre el sistema.

Esta evaluación permite priorizar las acciones correctivas y establecer un orden de remediación acorde al riesgo real que cada vulnerabilidad representa para la aplicación.

Evaluación de riesgos basada en metodología OWASP

Tabla con calculadora de riesgos OWASP

ID	Vulnerabilidad	Severidad	Probabilidad	Impacto	Estado
4.1	Inyección SQL	CRÍTICA	Alta	Muy Alto	Explotado
4.2	XSS Reflejado	ALTA	Media	Medio-Alto	Explotado
4.3	XXE (XML External Entity)	CRÍTICA	Media-Alta	Muy Alto	Explotado
4.4	Componentes Obsoletos (jQuery UI)	ALTA	Alta	Medio-Alto	Confirmado
4.4	Fallos de Autenticación (jQuery UI)	ALTA	Alta	Medio-Alto	Confirmado
4.5	Fallos de Autenticación (Passwords)	MEDIA	Alta	Alto	Confirmado

CRÍTICA ALTA
MEDIA Confirmado

https://microhackers.ai/es/herramientas-ciberseguridad/calculadora-evaluacion-riesgos/

5. Recomendaciones Generales

- Uso sistemático de consultas parametrizadas
- Validación y escape de entradas y salidas
- Deshabilitar XXE
- Actualizar dependencias
- Aplicar políticas de contraseñas fuertes
- Implementar defensas adicionales como CSP

6. Conclusión

La auditoría demuestra que una aplicación web puede presentar vulnerabilidades críticas tanto por errores propios como por el uso de componentes externos inseguros.
La adopción de buenas prácticas de desarrollo seguro y el mantenimiento continuo de las dependencias son medidas fundamentales para reducir el riesgo y mejorar la postura de seguridad de las aplicaciones web.

En conjunto, las vulnerabilidades identificadas demuestran que una explotación encadenada podría comprometer de forma significativa la confidencialidad de los datos, la integridad de la aplicación y la confianza de los usuarios.