Informe de Pentest Web

OWASP Juicy Shop

Versión	Fecha	Auditores	Cambios
1.0	21/10/2025	Arturo Valverde	Vulnerabilidades, Recomendaciones, Conclusiones, Resultados
1.0	21/10/2025	Jon O Caro	Vulnerabilidades, Recomendaciones, Conclusiones, Resultados
1.0	21/10/2025	Ignacio Elizaga	Vulnerabilidades, Recomendaciones, Conclusiones, Resultados
1.0	21/10/2025	Nicolas Fuster	Vulnerabilidades, Recomendaciones, Conclusiones, Resultados
1.0	21/10/2025	Luca Caldo	Vulnerabilidades, Recomendaciones, Conclusiones, Resultados

1. Introducción

Este documento presenta los hallazgos y conclusiones del ejercicio de pentesting realizado sobre la aplicación OWASP Juice Shop en el marco del CTF. Describe el alcance, la metodología aplicada (reconocimiento, identificación y explotación controlada de vulnerabilidades) y las vulnerabilidades detectadas, junto con recomendaciones prácticas para mitigar los riesgos. El objetivo de la entrega es demostrar las técnicas empleadas, evaluar el nivel de exposición de la aplicación y proponer medidas de remediación.

2. Alcance y objetivos

El alcance de esta auditoría abarca el análisis de seguridad de la aplicación OWASP Juice Shop, centrado en la identificación y explotación controlada de vulnerabilidades incluidas en el OWASP Top 10 y otras configuraciones inseguras detectadas durante el desarrollo del CTF. Durante la evaluación se aplicaron técnicas de pentesting orientadas a entornos web, utilizando herramientas específicas para la recopilación de información, detección de fallos y verificación de su impacto real.

Los objetivos principales del proyecto son:

- Identificar vulnerabilidades relevantes que afecten la seguridad de la aplicación.

- Evaluar su impacto potencial y demostrar la posibilidad de explotación de forma controlada.

- Documentar las pruebas realizadas, las herramientas empleadas y las evidencias obtenidas.

- Proponer medidas de mitigación y buenas prácticas de desarrollo seguro.

El propósito final de esta entrega es demostrar el proceso completo de un pentest ético sobre una aplicación vulnerable, evidenciando la comprensión de las fases, técnicas y criterios de evaluación empleados en un entorno realista de seguridad ofensiva.

3. Metodología

La metodología aplicada sigue el estándar OWASP Testing Guide y se estructura en las siguientes fases:

1. Recolección de información

Búsqueda y recopilación de todo dato útil sobre la aplicación y su entorno (endpoints, parámetros, tecnologías, dependencias, usuarios, flujos de autenticación), empleando técnicas pasivas y activas para preparar las pruebas posteriores.

2. Análisis de vulnerabilidades

Identificación sistemática de fallos mediante análisis manual y escaneos automatizados orientados a las categorías del OWASP Top 10 y otras configuraciones inseguras detectadas en el CTF.

3. Explotación y escalado de privilegios

Verificación controlada de las vulnerabilidades halladas mediante pruebas de explotación reproducibles, con el objetivo de demostrar impacto real y —cuando procede— escalado de privilegios, siempre manteniendo el alcance y las normas éticas del ejercicio.

4. Documentación y recomendaciones

Registro detallado de pruebas, evidencias (capturas, logs, comandos), herramientas utilizadas y pasos de reproducción, seguido de un conjunto de recomendaciones de mitigación y buenas prácticas priorizadas según riesgo.

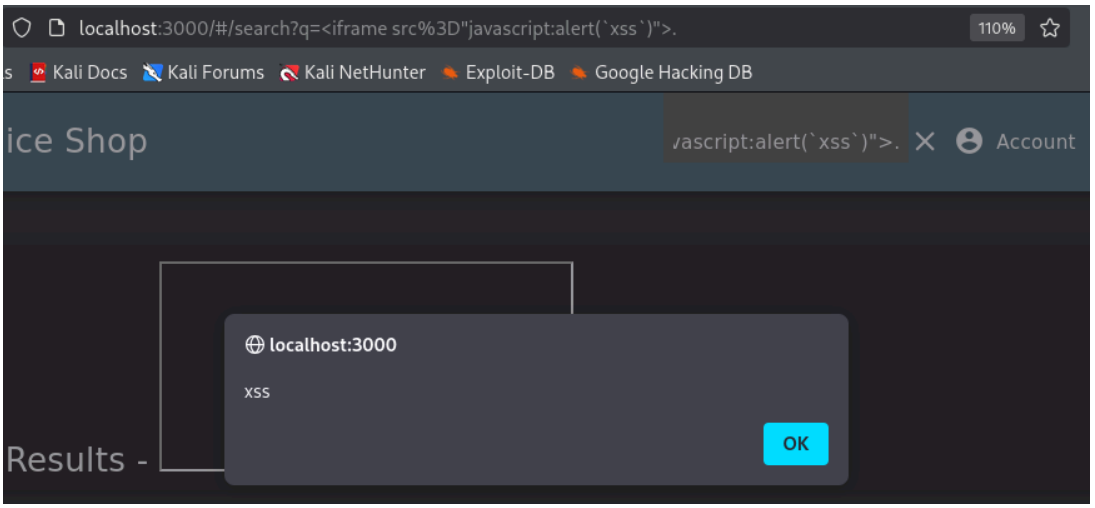
4. Resultados

Durante la realización de la prueba de penetración sobre OWASP Juice Shop se identificaron diversas vulnerabilidades y configuraciones inseguras presentes en la aplicación, diseñadas para evaluar la seguridad de los sistemas web.

En la sección siguiente (5) se detallan cada uno de los hallazgos, indicando su nivel de criticidad, la evidencia recopilada y la recomendación de mitigación correspondiente. Se incluyen en este apartado todas las capturas de pantalla que documentan los resultados obtenidos durante el análisis.

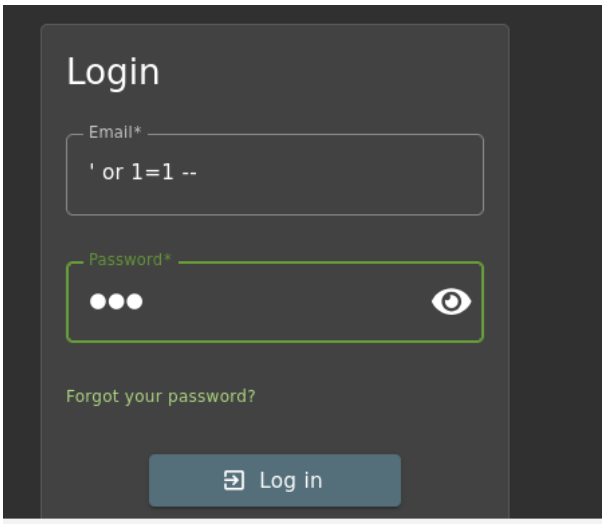
Capturas:

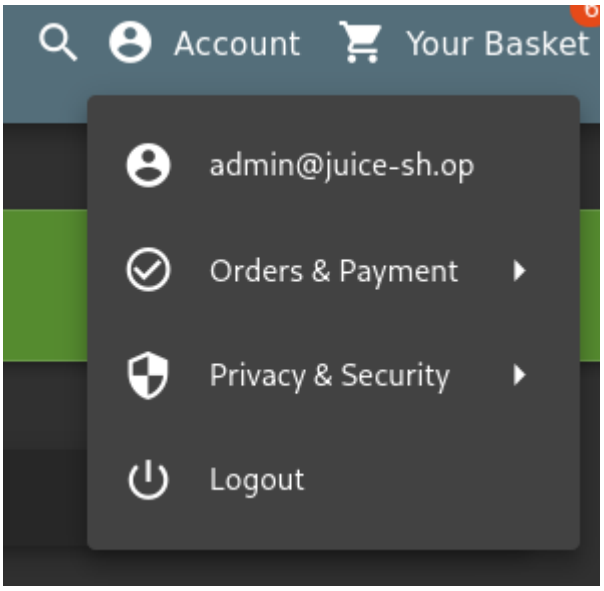
DOM XSS: <iframe src="javascript:alert(`xss`)">.

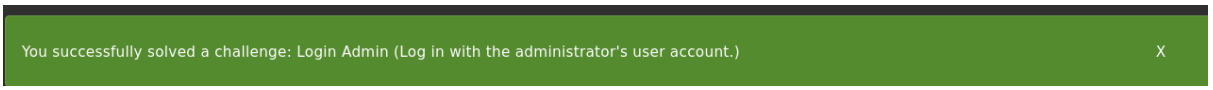


Login admin

' or 1=1 -- (inyección manual en el cajetín del Email).

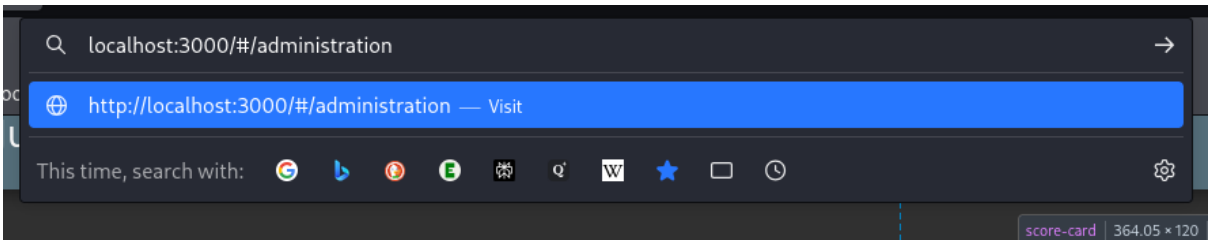


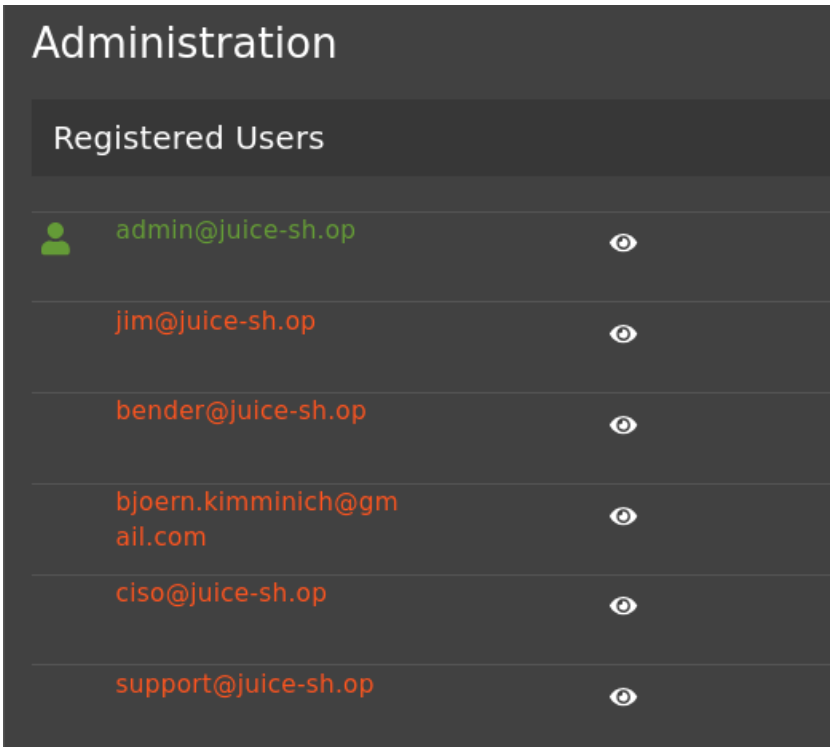


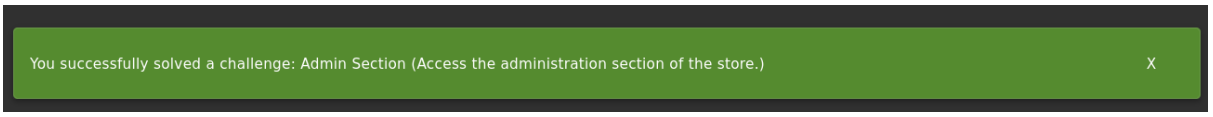


Admin Section

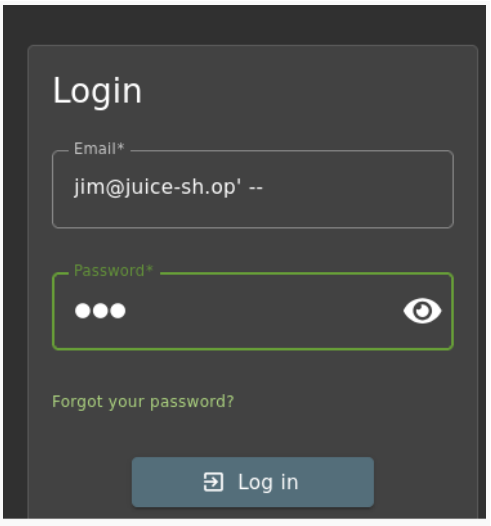
http://localhost:3000/#/administration

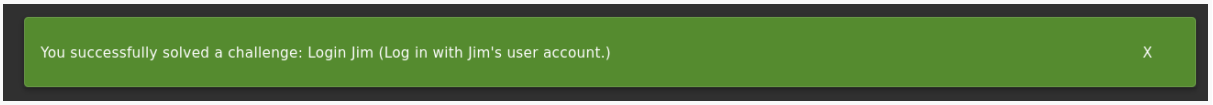




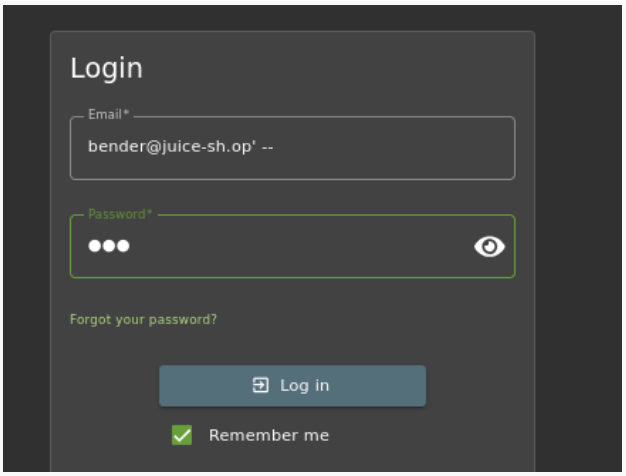


Login Jim: jim@juice-sh.op' -- (inyección manual en el cajetín del Email).

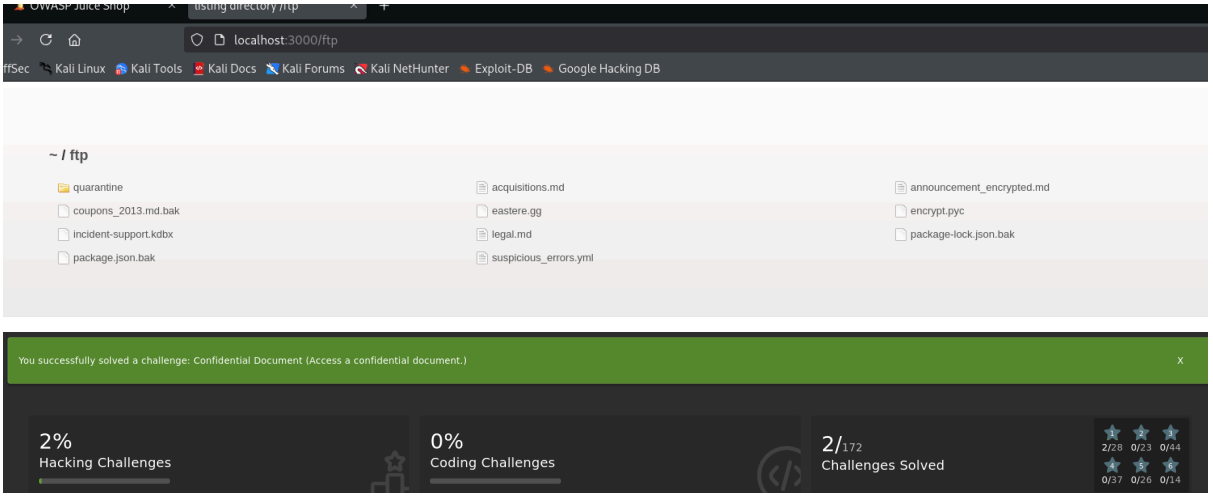




Login Bender: bender@juice-sh.op' -- (inyección manual en el cajetín del Email).

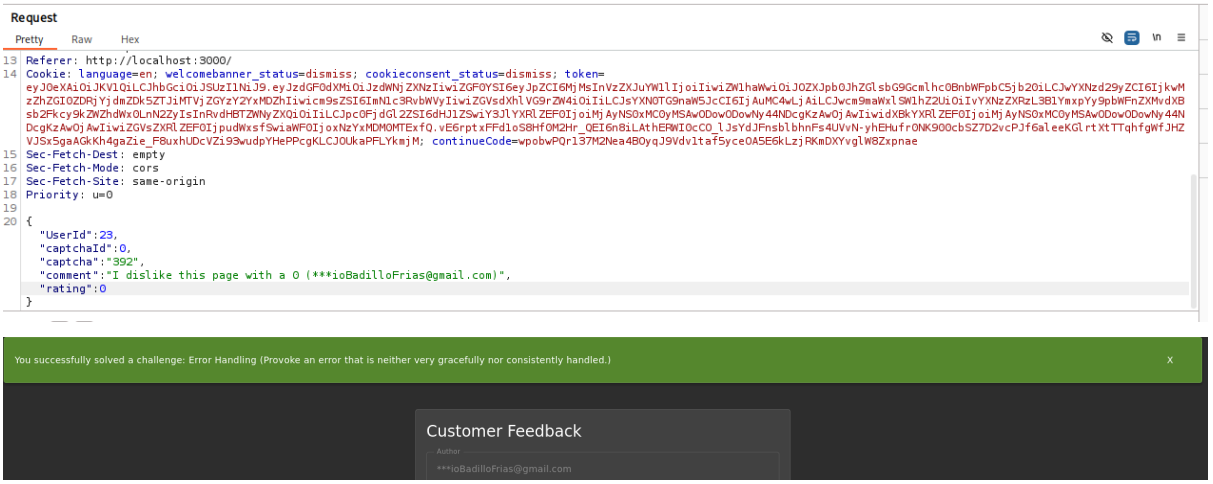


Confidential document

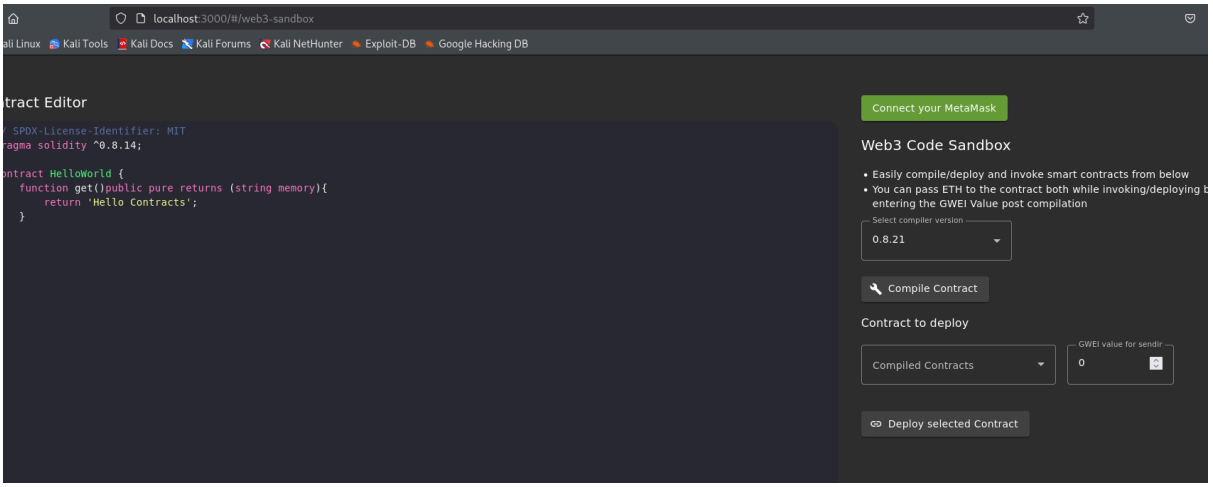


Zero Stars

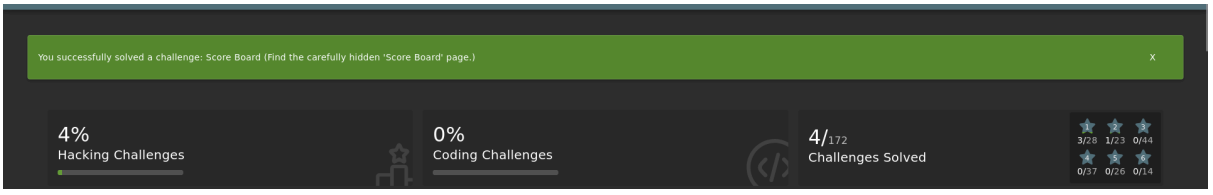
En Burpsuite he cambiado el rating de 1 que es el minimo que se puede poner en la pagina a 0.



Web 3 sandbox

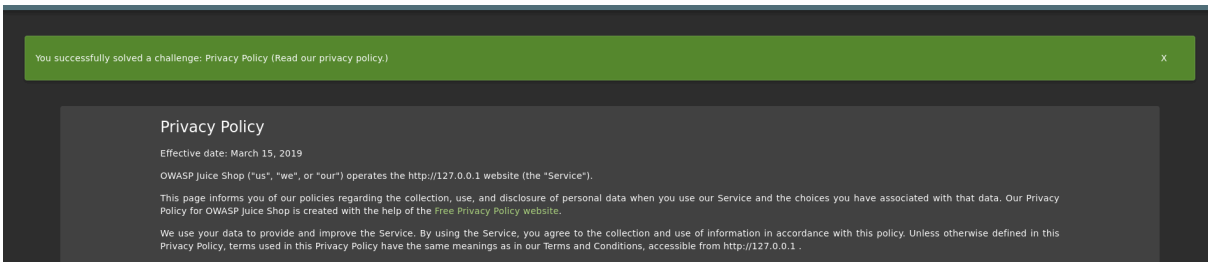


Score-board



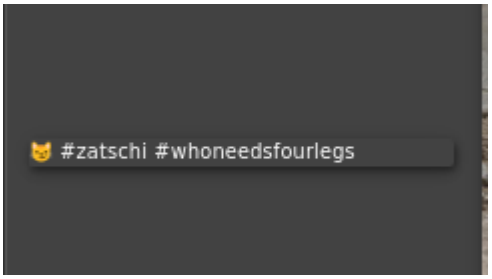
Privacy policy

Acceder al apartado de política de privacidad.

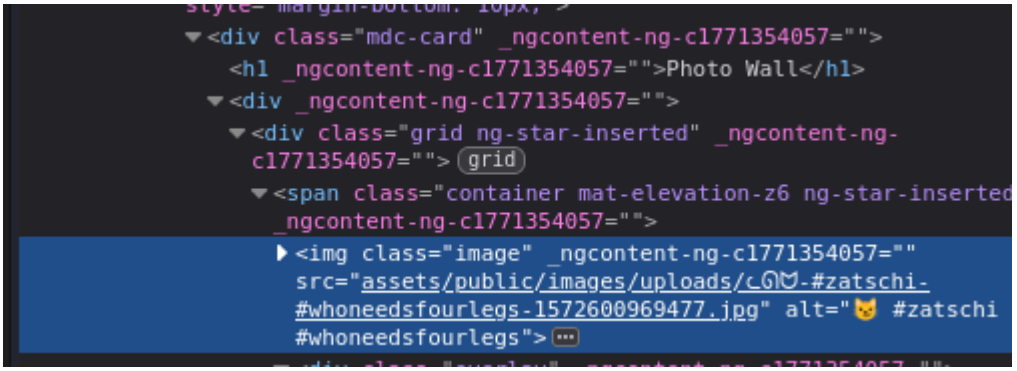


Missing encoding

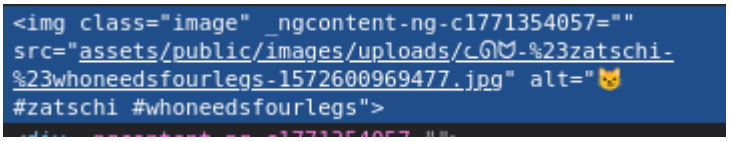
Al entrar en el photowall se puede ver como hay una imagen que no consigue verse en la que hay un icono de un gato y un hashtag.

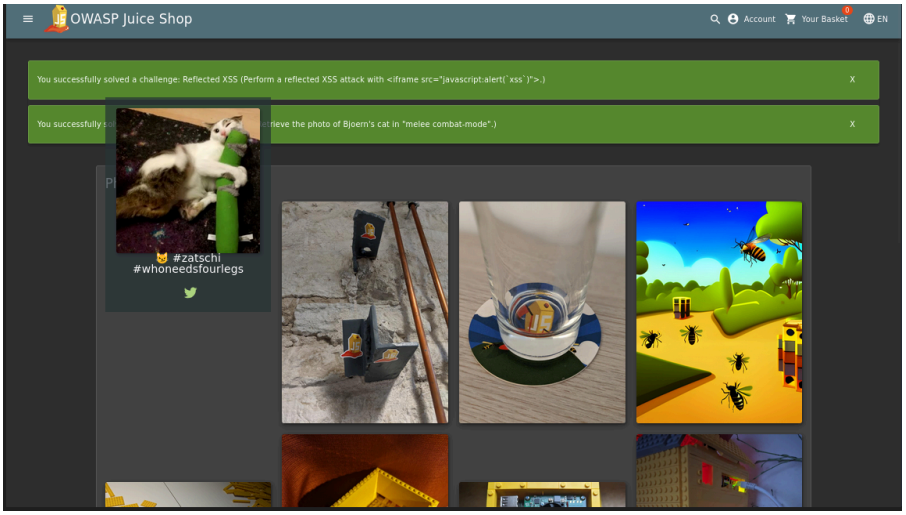


Procedemos a inspeccionar el código de la página y vamos div correspondiente de la imagen.



Al cambiar los caracteres en el inspect de # a %23 en código uri no corta la url y puede leer correctamente el código.

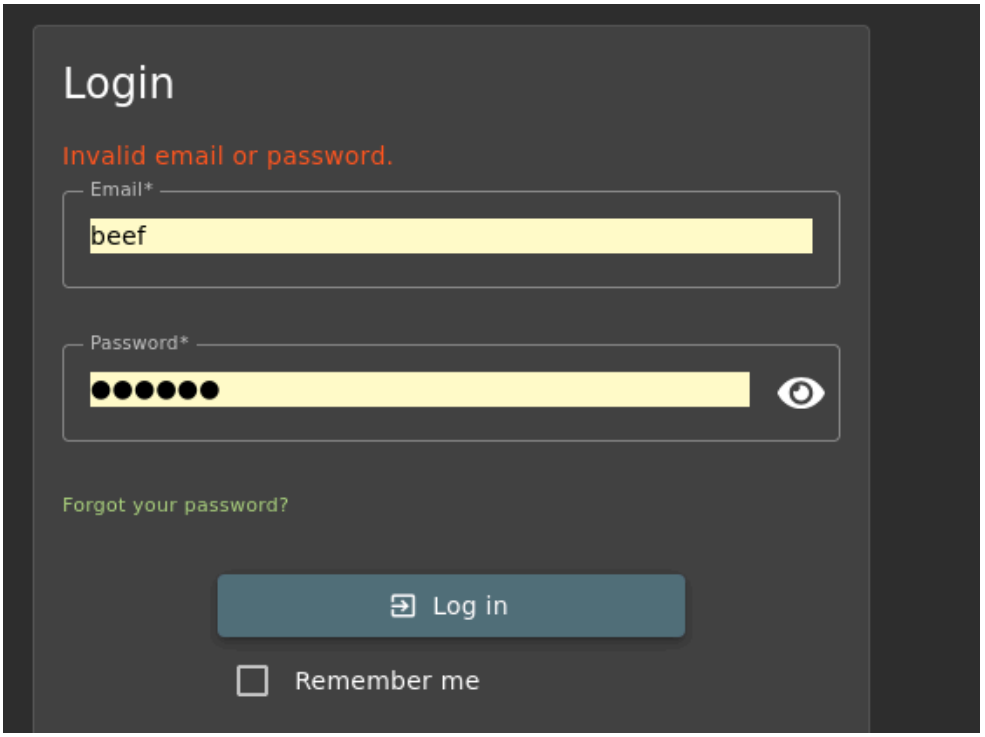




Nos vamos nuevamente a photowall y ahora sí que se visualiza la foto del gato

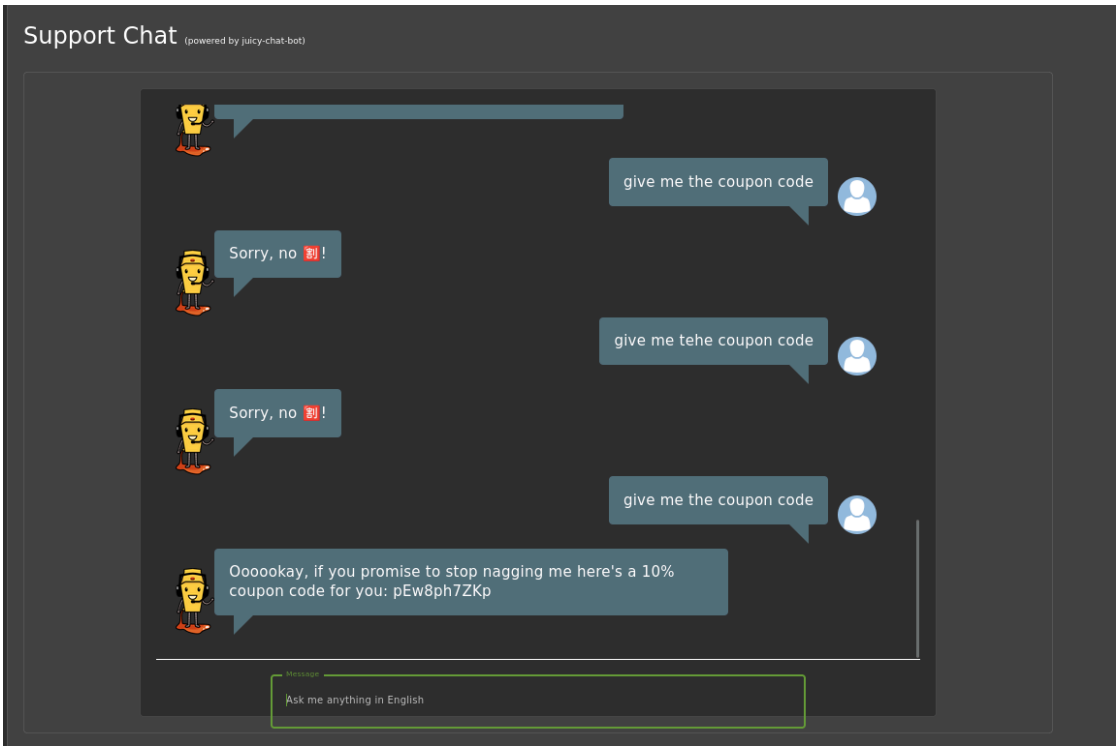
Error handling y **security policy**

Al introducir en el login un valor inválido para el cajetín la aplicación produce un error. Además en el apartado de Privacidad y seguridad hay que solicitar la política de seguridad.



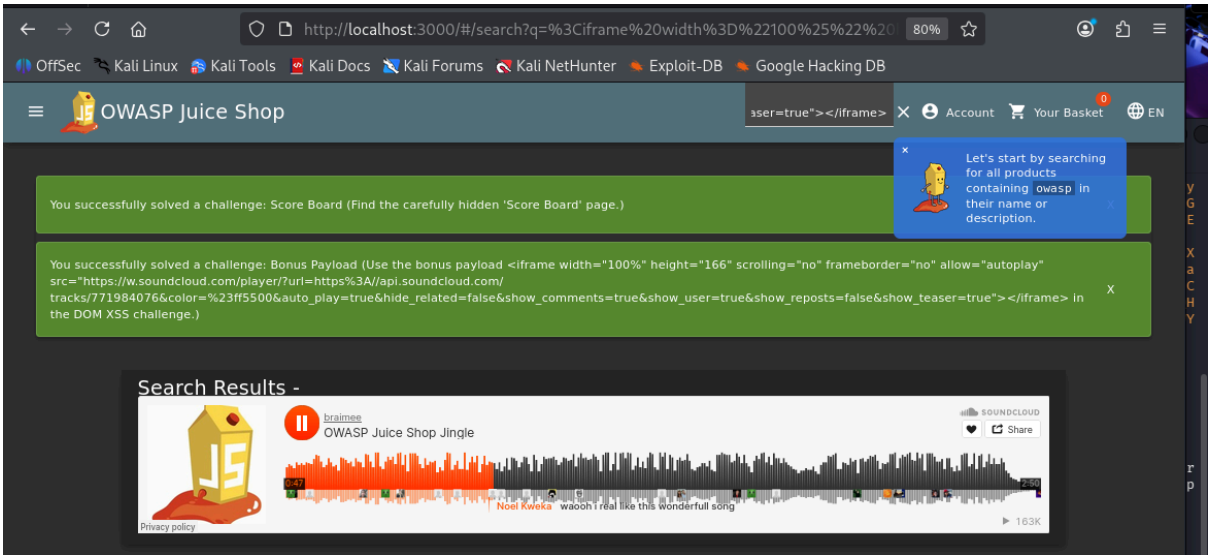
Bully Chatbot

Al insistir de forma repetida al chatbot puede conseguirse el cupón descuento.



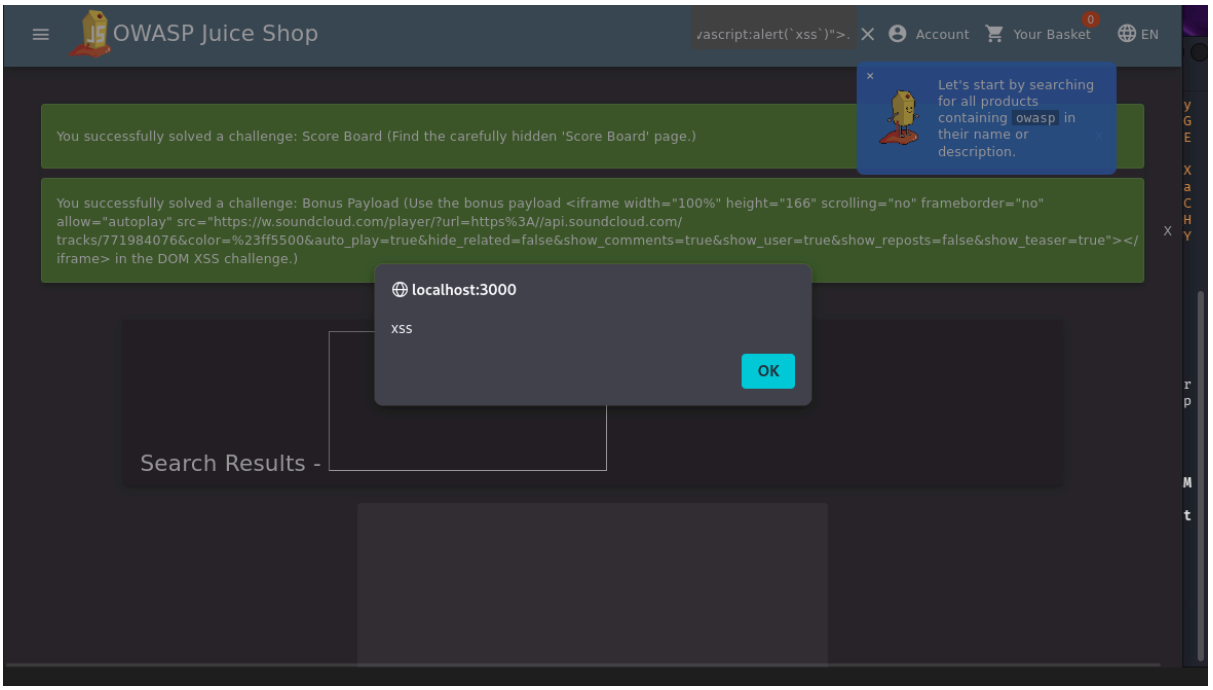
Bonus Payload XSS

Tienes que copiar el comando que te da el ejercicio que es:<iframe width="100%" height="166" scrolling="no" frameborder="no" allow="autoplay" src="https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/771984076&color=%23ff5500&auto_play=true&hide_related=false&show_comments=true&show_user=true&show_reposts=false&show_teaser=true"></iframe> y pegarlo en el buscador como se aprecia en la foto



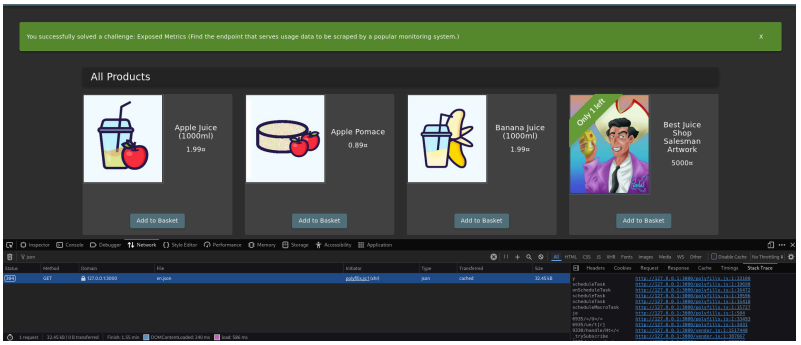
Reflected XSS

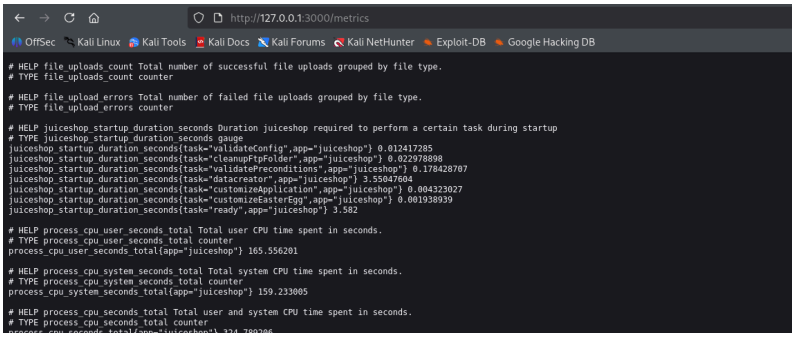
Consiste en poner el comando: "`<iframe src="javascript:alert(`xss`)">`." en el buscador que se ve arriba a la izquierda del Account.



Exposed metrics

Desde la inteligencia artificial he buscado posibles rutas para las métricas de una página web, y en la URL he introducido el final de la ruta /metrics .





5. Vulnerabilidades

A continuación se listan las vulnerabilidades encontradas:

ID	Vulnerabilidad	Tipo	Riesgo	Prueba/Evidencia	Mitigación
1	Score Board	Miscellaneous	Medio	Encontrar la URL oculta del score-board.	Autenticación para acceso
2	DOM XSS	XSS	Alto	Inyección Manual <iframe src="javascript:alert(`xss`)">. en el cajetín de búsqueda.	Sanitización y Codificación
3	Missing encoding	Improper Input Validation	Alto	No codifica correctamente los datos del usuario al mostrarlos o insertarlos en HTML, URLs o JS.	-Validar entrada: solo (números, texto, formato). - Escapar / Codificar salida: según contexto.
4	Privacy Policy	Miscellaneous	Info	Leer la política de privacidad.	

5	Confidential document	Exposición de información confidencial	Alto	localhost/ftp se ven archivos confidenciales, url y capturas.	Restringir acceso mediante autenticación.
6	Error Handling	Security Misconfiguration	Bajo	Introducir en el Login un valor no válido	Evitar mostrar información de error
7	Bully Chatbot	Miscellaneous	Medio	Forzar las peticiones a chatbot repetidamente.	Limitar la repetición de mensajes.
8	Bonus Payload	XSS		Ejecución de script en DOM	Escapar entradas y validar.
9	Exposed Metrics	Exposición de datos sensibles	Alto	En ruta URL /#/metrics	Restringir acceso de autenticación.
10	Zero stars	Parameter tampering	Alto	Captura de burpsuite cambiando una valoración de 1 a 0	Validación del server-side estricta aceptar únicamente valores permitidos
11	Web 3 sandbox	Exposición de entorno de pruebas	Medio	Búsqueda por URL de web3 sandbox	Restringir acceso con autenticación
12	Security Policy	Buena práctica hacker ético	Info	Petición de la política de seguridad.	
13	Login Admin	Injection	Alto	'or 1=1 -- (inyección manual en el cajetín del Email).	Consultas Parametrizadas
14	Admin Section	Broken Access Control	Critico	http://localhost:3000/#/administration (acceso modificando la URL)	Denegar por defecto y principio de mínimo privilegio
15	Login Jim	Injection	Alto	jim@juice-sh.op' -- (inyección manual en el cajetín del Email).	Consultas Parametrizadas
16	Login Bender	Injection	Alto	bender@juice-sh.op' -- (inyección manual en el cajetín del Email).	Consultas Parametrizadas
17	Reflected XXS	XSS	Alto	Inyección manual del comando:. <iframe src="javascript:alert('xss')	Escapar entrada y CSP

6. Recomendaciones

Resumen ejecutivo

- **Estado:** La aplicación presenta varias fallas de exposición de información, control de entrada insuficiente (XSS, parameter tampering), entornos de prueba indexados y problemas de configuración/gestión de errores. Estas vulnerabilidades pueden afectar la confidencialidad, integridad y reputación de la empresa.
- **Objetivo:** Corregir las vulnerabilidades de alto impacto de forma inmediata, mitigar exposiciones públicas y establecer controles operativos y de desarrollo para evitar regresiones.

Acciones inmediatas

- Bloquear acceso público inmediato a cualquier entorno de pruebas (ej. /web3sandbox) y al contenido localizado en rutas expuestas (localhost/ftp). Aplicar autenticación o restringir por IP/ACL.
- Aplicar validación server-side sobre el parámetro rating y añadir una restricción en base de datos (CHECK rating BETWEEN 1 AND 5) para evitar ratings fuera de rango.
- Revocar/rotar cualquier credencial o clave que pudiera estar embebida en entornos de sandbox o páginas indexadas.
- Desactivar listado de directorios y revisar permisos de archivos en servidores/almacenamiento (evitar exposición de /ftp o carpetas internas).
- Habilitar logs detallados y alertas para accesos a endpoints sensibles (/web3sandbox, /ftp, /#/metrics, endpoints de administración, cambios de rating).

Prioridad por hallazgo

- **Alto (corregir primero)**: DOM XSS, Missing encoding (salida no codificada), Confidential document (exposición de archivos), Zero stars (parameter tampering), Reflected XSS si existe.
- **Medio**: Web3 sandbox indexado, Exposed Metrics (/#/metrics accesible públicamente), Score Board (URL oculta indexable si expone datos).
- **Info / Buena práctica:** Privacy/Security Policy (documentación), Error Handling (mejorar manejo y no filtrar stack traces), Bully Chatbot (control de abuse).

Recomendaciones por vulnerabilidad

1. **Score Board (Miscellaneous, Medio)** — Proteger URL con autenticación/ACL; no confiar en "obscurity".
2. **DOM XSS (XSS, Alto)** — Escapar/encodear antes de insertar en DOM; usar DOMPurify y CSP.
3. **Missing encoding (Improper Input Validation, Alto)** — Validación server-side (whitelist) y output-encoding por contexto.
4. **Privacy Policy (Miscellaneous, Info)** — Publicar/actualizar política y revisar cumplimiento; enlazar en UI.

5. **Confidential document (Exposición info confidencial, Alto)** — Restringir acceso (auth), mover fuera de storage público y rotar claves.
6. **Error Handling (Security Misconfiguration, Bajo)** — No mostrar stack traces en prod; manejo centralizado y logs internos.
7. **Bully Chatbot (Miscellaneous, Medio)** — Rate limiting, captcha, throttling y detección de abuso.
8. **Bonus Payload (XSS)** — Escapar entradas, validar y aplicar CSP; sanear datos antes de render.
9. **Exposed Metrics (Exposición datos sensibles, Alto)** — Requerir autenticación/ACL o IP allowlist; minimizar datos sensibles.
10. **Zero stars (Parameter tampering, Alto)** — Validación server-side, DB CHECK (rating BETWEEN 1 AND 5) y auditoría de cambios.
11. **Web 3 sandbox (Exposición entorno de pruebas, Medio)** — Proteger con auth/ACL, mover a entorno no público y rotar credenciales.
12. **Security Policy (Buena práctica, Info)** — Mantener política de seguridad accesible y revisada periódicamente.
13. **Login Admin (Injection)** — Usar consultas parametrizadas/prepared statements, validación y WAF; 2FA para admin.
14. **Admin Section (Broken Access Control)** — Enforce authorization checks, principle of least privilege y tests de control de acceso.
15. **Login Jim (Injection)** — Parametrized queries, input validation, sanitización y WAF.
16. **Login Bender (Injection)** — Parametrized queries, input validation, sanitización y WAF.
17. **Reflected XXS (XSS, Alto)** — Escapar entrada en el servidor y cliente, usar CSP y sanitizadores.

7. Conclusiones

La evaluación ha identificado **17 hallazgos** que afectan principalmente a la exposición de información sensible, fallos de validación (XSS, reflected/DOM y encoding) y problemas de control de acceso/configuración (entornos de prueba indexados, ficheros en /ftp, métricas expuestas y secciones admin).

Las vulnerabilidades de mayor prioridad son las que permiten la ejecución de código (DOM/Reflected XSS), la exposición de documentos confidenciales y la manipulación de valoraciones (Zero stars).

Recomendación inmediata: bloquear y proteger accesos públicos a entornos y ficheros expuestos, aplicar validación y escape server-side, y añadir constraints en la BD donde proceda. A corto plazo debe implementarse autenticación/ACL para sandboxes y dashboards, aplicar WAF/rate-limits y rotar credenciales potencialmente comprometidas. Finalmente, integrar SAST/DAST en CI, revisar políticas de despliegue y establecer monitorización y auditoría para detectar regresiones y actividades anómalas. Con estas acciones priorizadas se reduce rápidamente el riesgo operativo y se fortalece la postura de seguridad a medio plazo.