Ingeniería en Desarrollo de Software

Actividad | 3 | Cross Site Scripting (XSS)

Auditoría Informática

Ingeniería en Desarrollo de Software

TUTOR: Jessica Hernández Romero

ALUMNO: Carlos Fco Estrada Salazar

FECHA: 25/ Oct /2025

INDICE

GitHub Link:

INTRODUCCIÓN

En la Actividad 3 se abordará la vulnerabilidad conocida como **Cross Site Scripting (XSS)** aplicada en un entorno controlado y previamente desplegado (la aplicación subida en la Actividad 1). El objetivo práctico es demostrar cómo un atacante puede inyectar código malicioso en páginas web que no validan ni sanitizan correctamente entradas de usuario, y cómo esa inyección puede usarse para capturar credenciales cuando un usuario inicia sesión. Para ello se empleará **Burp Suite** como herramienta de intermediación y manipulación de tráfico: se interceptarán las peticiones de login, se registrarán las credenciales transmitidas y se analizará cómo un vector XSS puede alterar la página para capturar datos en el DOM o redirigir formularios.

La actividad combina teoría (mecanismos de XSS: reflejado, persistente y basado en DOM) y práctica ética en un laboratorio propio, mostrando el flujo completo —desde la inyección hasta la exfiltración de credenciales— y su mitigación. Además, se verificará la posibilidad de modificar credenciales en tránsito (manipulación de parámetros/form-data) para evaluar el impacto sobre la autenticación y la integridad de la sesión.

DESCRIPCIÓN

El escenario planteado obliga a interpretar dos riesgos centrales: la **exposición de credenciales ante comunicaciones no seguras** y la **capacidad de XSS para interactuar con elementos de la interfaz y el DOM**. En la práctica, se usará la aplicación subida en la Actividad 1 como blanco legítimo; Burp Suite interceptará las solicitudes HTTP/HTTPS de inicio de sesión para mostrar cómo, si el sitio es vulnerable, un script inyectado puede capturar valores de campos username y password cuando el usuario los envía. El ejercicio incluye pasos concretos: localizar puntos de entrada susceptibles (campos de búsqueda, formularios, parámetros GET/POST), inyectar payloads apropiados (por ejemplo, "><script>...</script> o variantes basadas en DOM), y observar la exfiltración de credenciales hacia un endpoint controlado por el atacante (o simplemente su registro en la herramienta interceptora).

La actividad requiere manipular las peticiones con Burp (repeater, intruder o modificar el cuerpo POST) para comprobar si alterando valores en tránsito se puede forzar un inicio de sesión diferente o manipular la lógica de autenticación. Este proceso demuestra la relación directa entre validación insuficiente de entrada y pérdida de confidencialidad/ integridad. Finalmente, se documentan las evidencias: capturas de Burp, payloads usados, logs que contengan las credenciales capturadas y análisis del impacto (por ejemplo, posibilidad de suplantación o escalada de privilegios).

JUSTIFICACIÓN

Emplear un enfoque práctico que combine **la aplicación local/hosteada**, **Burp Suite** y ejercicios de XSS es la solución más adecuada para esta actividad por varias razones.

Primero, permite replicar un escenario realista y controlado donde las vulnerabilidades pueden ser identificadas, explotadas y mitigadas sin riesgo legal ni daño a terceros; esto es esencial para el aprendizaje responsable.

Segundo, Burp Suite ofrece un conjunto de herramientas integradas (proxy, repeater, intruder, scanner en versiones avanzadas) que facilitan interceptar, modificar y volver a enviar peticiones; esto es clave para entender cómo se comporta la aplicación ante inputs maliciosos y para verificar manualmente las hipótesis sobre la explotación XSS.

Tercero, la metodología expone tanto la técnica del atacante (inyección y exfiltración) como las contramedidas defensivas (escape/sanitización, políticas de Content Security Policy, uso de HttpOnly/Secure en cookies, validación del lado servidor), lo que convierte el ejercicio en una lección completa de prevención y respuesta. Además, probar la manipulación de credenciales en tránsito y la captura vía DOM permite evaluar la eficacia de controles existentes (por ejemplo, protección de formularios, encriptación en transporte).

Finalmente, para el entorno laboral, esta aproximación fortalece competencias críticas: auditoría proactiva, documentación de pruebas, comunicación de riesgos y propuesta de soluciones técnicas y operativas, todo lo cual es directamente aplicable en revisiones de seguridad reales y procesos de desarrollo seguro.

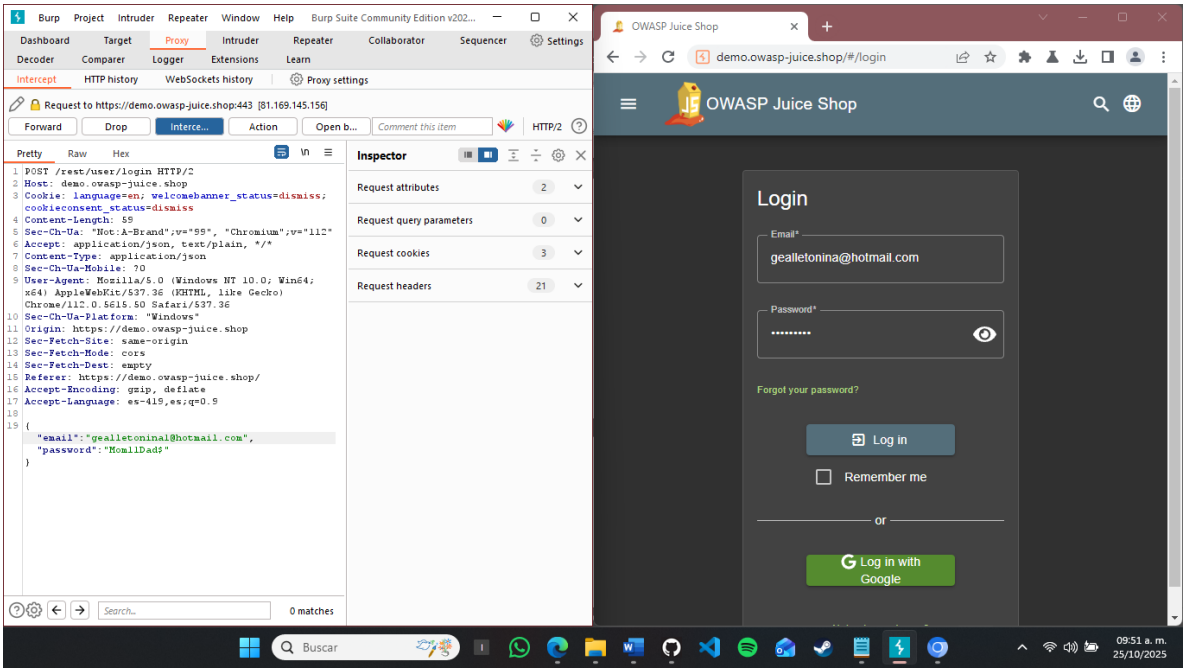
DESARROLLO

Descripción del sitio:

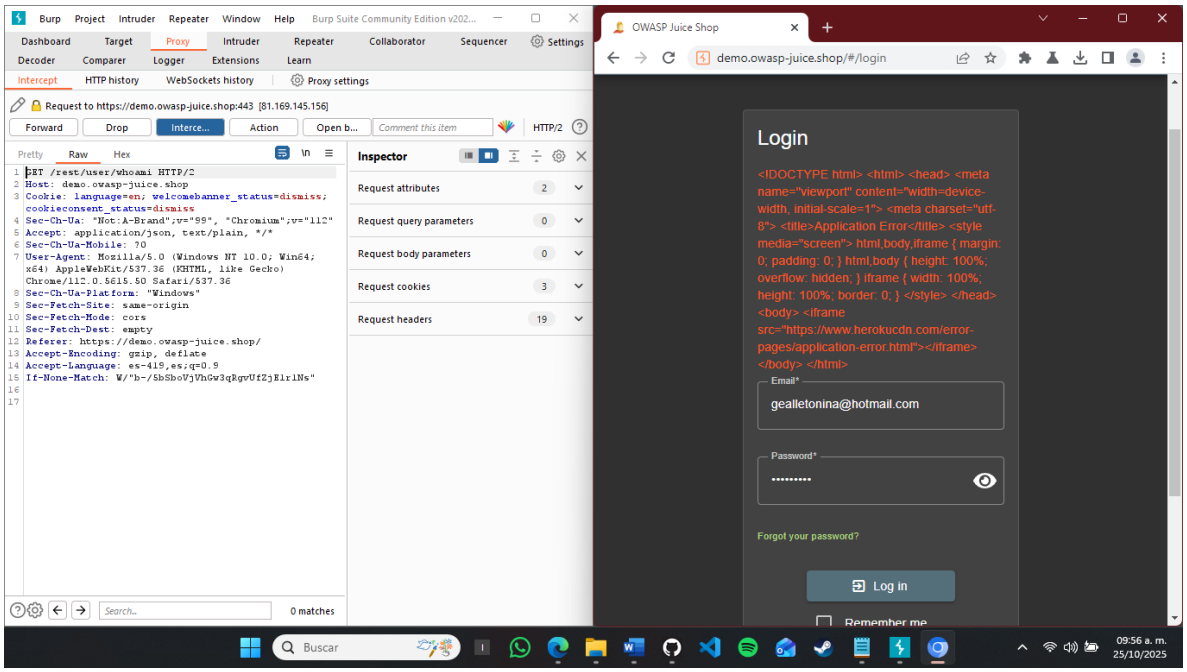
OWASP Juice Shop es una aplicación web intencionalmente vulnerable creada por el proyecto OWASP (Open Web Application Security Project) con fines educativos y de práctica en ciberseguridad. Su objetivo principal es ayudar a estudiantes, desarrolladores y analistas de seguridad a comprender y detectar las vulnerabilidades más comunes en aplicaciones web modernas. Está desarrollada con tecnologías como Node.js, Express y Angular, simulando una tienda en línea donde los usuarios pueden realizar compras ficticias. Sin embargo, la aplicación contiene múltiples fallos de seguridad basados en el OWASP Top 10, como inyección SQL, cross-site scripting (XSS), exposición de datos sensibles y fallas de autenticación. Juice Shop se utiliza ampliamente en cursos, laboratorios y competiciones de hacking ético (CTFs), ofreciendo una forma segura y controlada de aprender cómo funcionan los ataques reales y cómo proteger los sistemas frente a ellos.

Ataque al sitio:

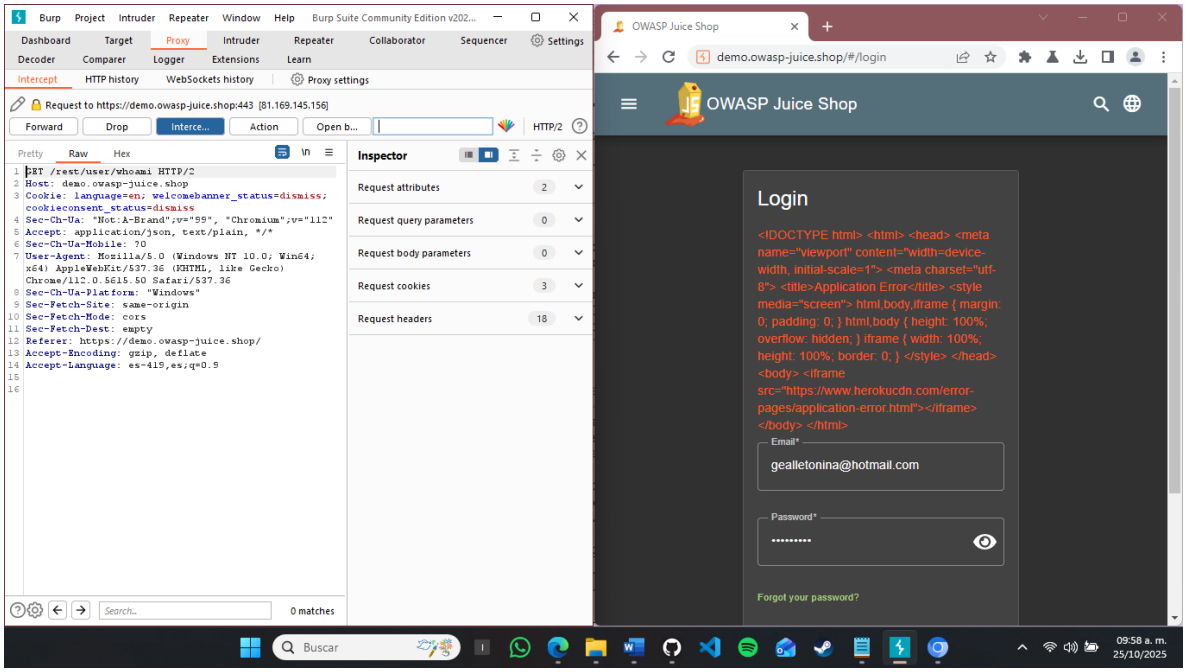
Interceptando datos.



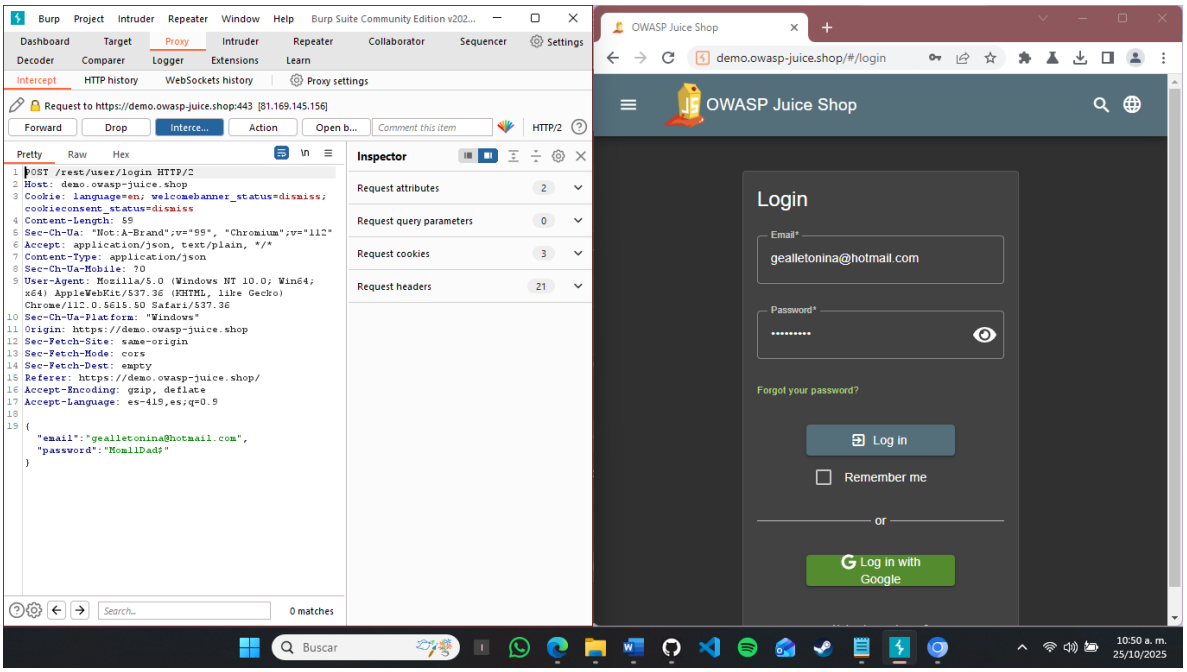
Modificando correo.



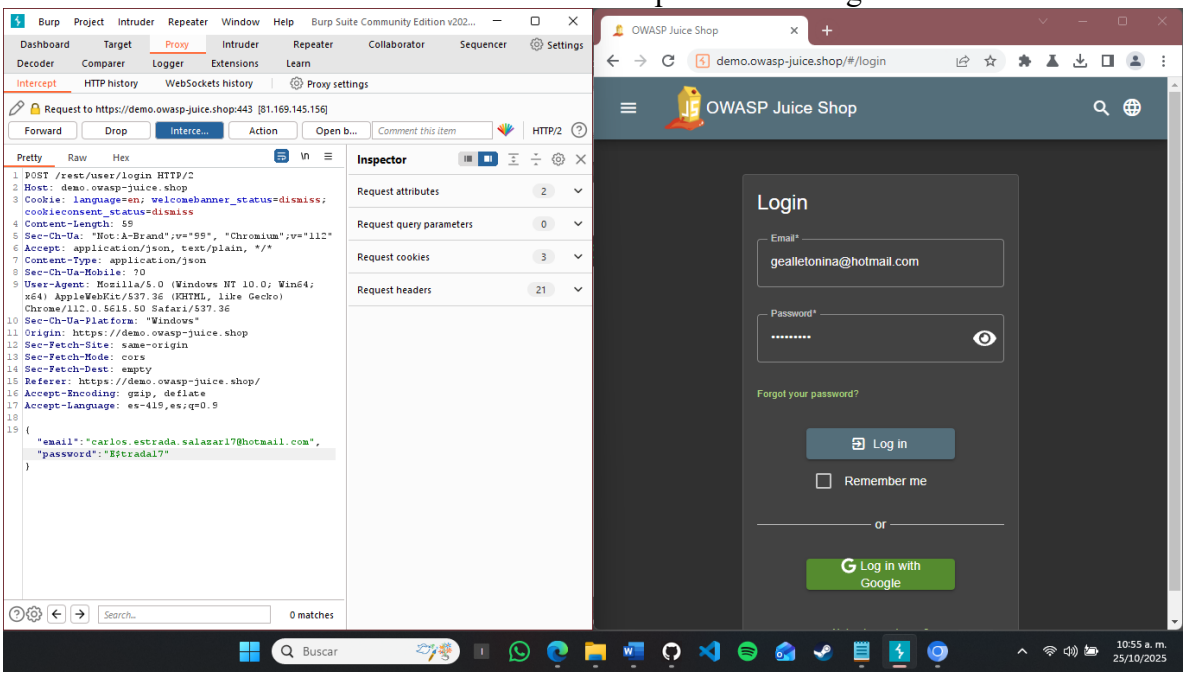
Modificando contraseña.



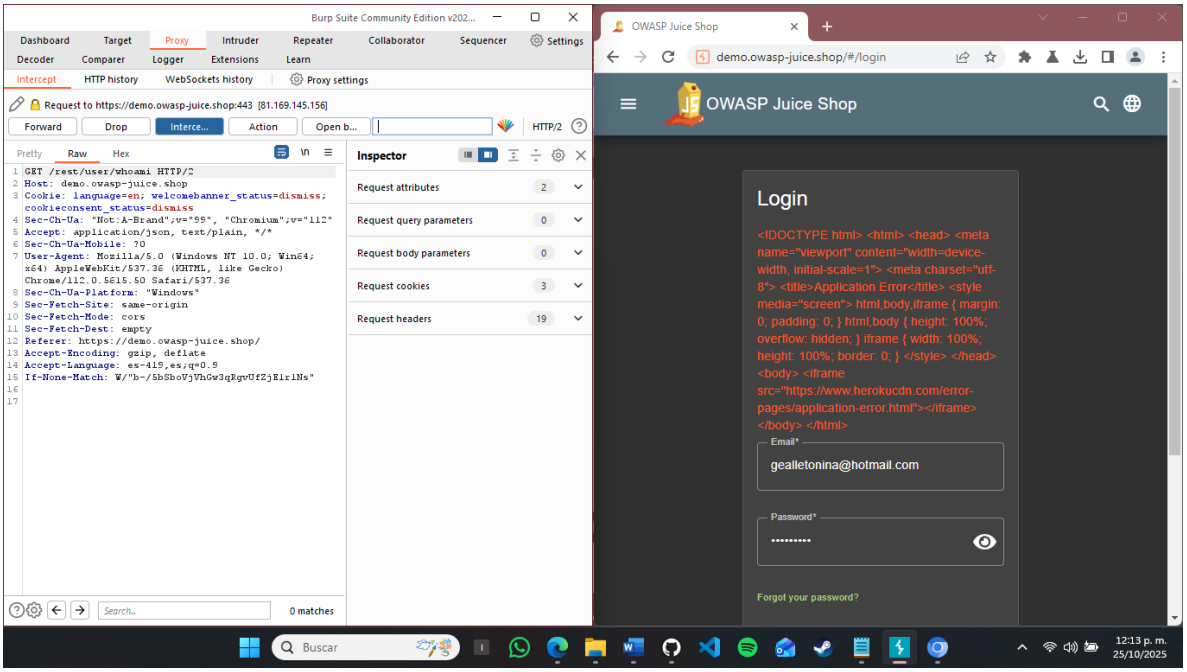
Cambiando las credenciales del usuario por otras con previo registro.



Se colocan las credenciales de otro usuario previamente registrado.



Resultado del ataque



COLCUSIÓN

La realización de la actividad 3 sobre **Cross Site Scripting (XSS)** permitió comprender de manera práctica la gravedad de esta vulnerabilidad y su impacto directo en la seguridad de la información tanto en entornos laborales como en la vida cotidiana. En el ámbito profesional, este tipo de ataques representa uno de los riesgos más comunes y peligrosos para las aplicaciones web, ya que permite a un atacante inyectar código malicioso que puede robar credenciales, secuestrar sesiones o manipular el contenido mostrado a los usuarios. Mediante el uso de **Burp Suite**, se pudo observar cómo un fallo en la validación de entradas abre la puerta a ataques de inyección que comprometen la confidencialidad e integridad de los datos.

Desde la perspectiva laboral, este ejercicio fortalece habilidades fundamentales en auditoría informática, análisis de vulnerabilidades y pruebas de penetración controladas, competencias esenciales para cualquier profesional de desarrollo o ciberseguridad. Además, resalta la importancia de aplicar buenas prácticas de programación segura, como la sanitización de entradas, el uso de codificación adecuada en HTML y la implementación de políticas de seguridad del contenido (CSP).

En la vida cotidiana, la práctica demuestra por qué los usuarios deben ser conscientes de los sitios en los que ingresan sus datos y de los riesgos de interactuar con páginas sin cifrado ni certificados válidos. En conjunto, esta actividad fomenta una cultura de prevención y responsabilidad digital, esencial para proteger tanto sistemas empresariales como la información personal en un entorno cada vez más interconectado.

REFERENCIAS

- *Session Management - OWASP Cheat Sheet Series*. (s. f.). https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html?utm_source=chatgpt.com

- *Web application security, testing, & scanning - PortSwigger*. (s. f.). https://portswigger.net/

- *PortSwigger - labs*. (s. f.). https://portswigger-labs.net/

- *Home - Wireshark Wiki*. (s. f.). https://wiki.wireshark.org/