**Pruebas de Penetración en OWASP Juice Shop**

|  |  |
| --- | --- |
| Título del informe: Pruebas de Penetración en OWASP Juice Shop | |
| Portafolio:  Lección 1: | Entorno:  OWASP Juice Shop web app |
| Clasificación del documento:  Entorno vulnerable para pruebas – publica | Modulo:  Principios de seguridad en aplicaciones web |
| Autor(es): Sebastián Hernández Téllez | Fecha de elaboración: 23 de agosto de 2025 |

**1. Resumen Ejecutivo**

El laboratorio tuvo como objetivo realizar pruebas de penetración en OWASP Juice Shop, una aplicación web vulnerable diseñada para la enseñanza de ciberseguridad. Mediante técnicas manuales y el uso de herramientas como OWASP ZAP y Burp Suite, se identificaron vulnerabilidades críticas: inyección SQL, Cross-Site Scripting (XSS) y Cross-Site Request Forgery (CSRF). Estas fallas fueron explotadas de forma controlada, evidenciando riesgos como acceso no autorizado, robo de sesiones y manipulación de datos. Se propusieron medidas de mitigación como consultas parametrizadas, validación de entradas, tokens CSRF, cabeceras de seguridad y autenticación multifactor. El ejercicio demostró la importancia del pentesting ético en la prevención de ataques reales y en la mejora de la seguridad de aplicaciones web empresariales.

***Escenario***

El entorno vulnerable **OWASP Juice Shop** fue diseñado para la enseñanza de prácticas de seguridad en aplicaciones web. En este ejercicio se realizaron pruebas de penetración avanzadas con herramientas como **OWASP ZAP** y **Burp Suite**, con el objetivo de identificar y explotar vulnerabilidades presentes en la aplicación, así como proponer medidas de mitigación.

**2. Metodología de Pruebas**

* **Exploración manual:** Navegación por formularios de registro, login y campos de búsqueda, intentando inyecciones SQL y ataques XSS.
* **OWASP ZAP:** Escaneo automatizado para detectar vulnerabilidades comunes (XSS, SQL Injection, CSRF).
* **Burp Suite:** Interceptación y modificación de solicitudes HTTP para probar manipulación de parámetros y elevación de privilegios.
* **Pruebas de explotación controlada:** Validación de vulnerabilidades explotables con ejemplos prácticos.

**3. Vulnerabilidades Identificadas**

***VULN-001: Inyección SQL en formulario de login***

* **Descripción:** El campo de usuario aceptó payloads de SQL (' OR '1'='1 --).
* **Impacto:** Acceso no autorizado a cuentas de usuario.
* **Evidencia:** Login exitoso sin credenciales válidas.
* **Mitigación:** Uso de **consultas parametrizadas (Prepared Statements)** y validación de entradas.

***VULN-002: Cross-Site Scripting (XSS) en campo de búsqueda***

* **Descripción:** El campo de búsqueda reflejó código malicioso (<script>alert('XSS')</script>).
* **Impacto:** Ejecución de scripts arbitrarios en navegadores de usuarios. Robo de cookies o redirecciones maliciosas.
* **Mitigación:** Escapar/validar entradas y aplicar políticas **CSP (Content Security Policy)**.

***VULN-003: CSRF en actualización de perfil***

* **Descripción:** Se construyó un enlace malicioso que permitía cambiar el correo del usuario autenticado sin su consentimiento.
* **Impacto:** Modificación no autorizada de información de usuarios legítimos.
* **Mitigación:** Inclusión de **tokens CSRF únicos** en formularios y validación en el servidor.

**4. Resultados de Escaneo Automático por medio de OWASP ZAP**

* **Alertas altas:** XSS reflejado, Inyección SQL.
* **Alertas medias:** Posible CSRF, uso de cookies inseguras (sin flag HttpOnly).
* **Alertas bajas:** Cabeceras de seguridad ausentes (X-Frame-Options, X-XSS-Protection).

**5. Explotación de Vulnerabilidades**

* **XSS:** Ejecución de alertas en navegador y demostración de robo de sesión.
* **SQL Injection:** Acceso a cuentas administrativas mediante bypass de login.
* **CSRF:** Ejecución de un script externo para modificar información de perfil.

**6. Recomendaciones de Seguridad**

1. Implementar **validación y saneamiento estricto de entradas**.
2. Usar **consultas parametrizadas** en todas las operaciones SQL.
3. Aplicar **tokens CSRF** en formularios sensibles.
4. Configurar cabeceras de seguridad HTTP (CSP, HSTS, X-Frame-Options).
5. Implementar **autenticación multifactor (MFA)** para accesos administrativos.
6. Considerar el uso de un **WAF (Web Application Firewall)** para mitigar ataques comunes.

**7. Reflexión**

Este ejercicio evidenció cómo una aplicación vulnerable puede ser explotada mediante ataques relativamente simples. El uso combinado de exploración manual, herramientas automatizadas y técnicas de explotación permitió entender la criticidad de estas vulnerabilidades. Además, se refuerza la importancia de integrar **ciberseguridad en el ciclo de vida del desarrollo de software (SDLC)** y aplicar controles preventivos antes del despliegue en producción.

**8. Conclusión**

Las pruebas de penetración en **OWASP Juice Shop** demostraron la presencia de vulnerabilidades críticas como **SQL Injection, XSS y CSRF**, que comprometen la confidencialidad e integridad de la aplicación. Se logró explotarlas de forma controlada y documentar las pruebas con evidencias. Se concluye que la aplicación requiere implementar medidas de mitigación inmediatas para reducir la superficie de ataque. Este laboratorio refuerza el valor del **pentesting ético** como herramienta fundamental para garantizar la seguridad de aplicaciones web en entornos empresariales.