**Informe Pentest Ético OWASP Juice Shop – Fin de Modulo**

|  |  |
| --- | --- |
| Título del informe: Informe de pentesting ético – OWASP Juice-Shop web app | |
| Portafolio:  Informe final de modulo | Entorno:   * Kali Linux desde virtualbox * Imagen de OWASP Juice-Shop levantada con Docker Compose. |
| Clasificación del documento:   * Acceso hipotético a información confidencial con fines educativos. * Entorno vulnerable para pruebas(OWASP juice shop) | Modulo:  Análisis de Amenazas y Vulnerabilidades en Aplicaciones Web |
| Autor(es): Sebastián Hernández Téllez | Fecha de elaboración: 25 de agosto de 2025 |

**1. Resumen Ejecutivo**

Se realizó un pentest controlado sobre OWASP Juice Shop desplegado en Docker y evaluado desde Kali Linux en VirtualBox. El escaneo con Nmap confirmó la exposición del puerto 3000 y con SQLMap se detectó y explotó una vulnerabilidad de inyección SQL en el parámetro q. Esta falla permitió enumerar bases de datos, extraer credenciales y demostrar un impacto crítico en la confidencialidad de la información. El crackeo de hashes evidenció el riesgo de escalamiento no autorizado. Se recomendaron controles como validación de entradas, consultas parametrizadas y algoritmos de hash seguros. Todas las pruebas se ejecutaron en un entorno aislado bajo principios éticos y legales.

***Principales hallazgos:***

* VULN-001: inyección SQL en parámetro “q”.
* VULN-002: Contraseñas débiles (hashes fácilmente crackeables)

***Recomendaciones clave:***

* Implementar consultas parametrizadas y validación estricta de entradas para prevenir inyecciones SQL.
* Migrar a algoritmos de hash seguros (bcrypt, Argon2) y aplicar políticas de contraseñas robustas.
* Segregar privilegios en la base de datos para limitar el impacto en caso de explotación.
* Configurar un WAF que detecte patrones de ataque comunes como inyecciones.
* Establecer un proceso continuo de pruebas de seguridad y capacitación en OWASP Top 10 para el equipo de desarrollo.

***Nivel de riesgo general***:

**Alto/Critico** – La aplicación presenta vulnerabilidades que permiten comprometer credenciales y acceder a información sensible. De no mitigarse, estas fallas podrían conducir a pérdida de datos, comprometer cuentas de usuarios y habilitar un escalamiento de privilegios.

**2. Alcance de la Evaluación**

***Tipo de aplicación:***  
E-commerce vulnerable de prueba.

***Entorno evaluado***:  
OWASP Juice Shop desplegado en Docker, puerto 3000.

**3. Metodología**

*- Fase de reconocimiento (Nmap) → validación de superficie de ataque.*

*- Pruebas de vulnerabilidades (SQLMap) → inyección SQL confirmada.*

*- Explotación controlada (SQLMap) → extracción de datos sensibles (usuarios/contraseñas).*

*- Post-explotación (SQLMap) → crackeo de hashes cifrados con md5.*

***Herramientas utilizadas***:

Nmap y SQLMap.

**4. Hallazgos de Seguridad**

***4.1 VULN-001: inyección SQL en parámetro “q”***

***1. Identificador***: VULN-001

***2. Descripción***: Se identificó una vulnerabilidad de **inyección SQL** en el parámetro q del endpoint /rest/products/search. Esta falla permite ejecutar consultas maliciosas contra la base de datos subyacente.

***3. Clasificación***: OWASP A03:2021 - Injection

***4. Impacto potencial***: Acceso no autorizado a datos sensibles (usuarios, contraseñas, correos). Riesgo de exfiltración completa de la base de datos.

***5. Evidencia técnica***:

Comando ejecutado:

**sqlmap -u "http://<IP>:3000/rest/products/search?q=1" --batch --level=2**

Resultado: Confirmación de inyección y motor de base de datos SQLite.  
  
***6. Reproducción paso a paso***:

* Acceder al endpoint vulnerable /rest/products/search.
* Inyectar el parámetro con 1' OR '1'='1.
* Validar respuesta anómala.
* Automatizar con SQLMap para enumerar bases de datos y usuarios.

***7. Recomendación de mitigación***:

* Implementar **consultas parametrizadas** (Prepared Statements).
* Validar y sanear entradas de usuario.
* Incorporar un **WAF** que detecte patrones de inyección.

***8. Nivel de riesgo***: **Crítico** (CVSS > 9).

* 1. ***VULN-002: Contraseñas débiles***

**1. *Identificador***: VULN-002

***2. Descripción***: Se detectó el uso de **contraseñas débiles y hasheadas en MD5**, un algoritmo inseguro y obsoleto. Algunos hashes fueron crackeados exitosamente.

***3. Clasificación***: OWASP A02:2021 – Cryptographic Failures

***4. Impacto potencial***: Compromiso de cuentas de usuario, escalamiento de privilegios si un atacante obtiene credenciales administrativas.

***5. Evidencia técnica***:

* Hash extraído: 0c36e517e3fa95aabf1bbffc6744a4ef
* Crackeo exitoso: contraseña trivial obtenida desde diccionario.

**6. Reproducción paso a paso:**

1. Extraer credenciales con SQLMap (--dump).
2. Crackeo de contraseñas mediante diccionario “md5\_generic\_passwd”

***7. Recomendación de mitigación***:

* Migrar a algoritmos modernos (bcrypt, Argon2).
* Implementar políticas de contraseñas robustas (longitud mínima, complejidad).
* Forzar rotación periódica de credenciales.

***8. Nivel de riesgo***: **Alto**.

**5. Evaluación de Riesgos**

***5.1 Matriz de Riesgo***

| **Riesgo** | **Probabilidad** | **Impacto** | **Nivel de Riesgo** | **Recomendación Prioritaria** |
| --- | --- | --- | --- | --- |
| VULN-001 | Alta | Crítico | Crítico | Mitigación inmediata con validación de entradas y consultas parametrizadas |
| VULN-002 | Media | Alto | Alto | Migrar a algoritmos seguros y aplicar políticas de contraseñas |

***5.2 Resumen por Categoría OWASP***

| **Categoría OWASP** | **Vulnerabilidades Detectadas** | **Severidad Promedio** |
| --- | --- | --- |
| A03:2021 – Injection | VULN-001 | Crítico |
| A02:2021 – Cryptographic Failures | VULN-002 | Alto |

**6. Recomendaciones Generales**

* Adoptar un ciclo de **SDLC seguro**, integrando revisiones de código y pruebas automatizadas de seguridad.
* Implementar **escaneo periódico de vulnerabilidades**.
* Capacitar al equipo de desarrollo en **OWASP Top 10** y buenas prácticas de seguridad.
* Segregar ambientes (desarrollo, pruebas, producción).

**7. Conclusiones**

El ejercicio demostró que **OWASP Juice Shop** contiene vulnerabilidades críticas que permiten el acceso no autorizado a información sensible. La explotación de inyección SQL y contraseñas débiles evidenció fallas graves en la validación de entradas y en la gestión de credenciales. Estas vulnerabilidades, de encontrarse en un entorno productivo, tendrían un impacto severo en la confidencialidad, integridad y disponibilidad de la información. La aplicación de controles como **consultas parametrizadas**, **hashing robusto** y **políticas de contraseñas seguras** reduciría significativamente los riesgos. El laboratorio permitió comprender la importancia de pruebas éticas y controladas, resaltando el rol preventivo del pentesting en la seguridad organizacional.

Me asegure de realizar las prueba en un entorno local, controlado y ficticio, sin embargo fue de mucha utilidad para aprender procedimientos prácticos reales.

**8. Anexos**

***Anexo – Capturas de pantalla***



* **Verificar acceso al objetivo por medio de nmap**

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

* **Confirmación de vulnerabilidad SQLi**

A computer screen shot of text

AI-generated content may be incorrect.

A screen shot of a computer screen

AI-generated content may be incorrect.

**enumeración y extracción de datos**

* **Visualizar tablas en la base de datos:**

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* **visualización de las columnas en la tabla Users:**

A computer screen shot of a computer

AI-generated content may be incorrect.

* **Extracción de credenciales de acceso y crackeo de contraseñas:**

A screenshot of a computer

AI-generated content may be incorrect.