**Informe de Seguridad en Aplicación Web – App ShopSecure**

|  |  |
| --- | --- |
| Título del informe: Informe de Evaluación de Seguridad – App ShopSecure | |
| Autor(es): Sebastián Hernández Téllez | Entidad o cliente evaluado: Retail Chile S.A. |
| Clasificación del documento: Confidencial | Fecha de elaboración: 10 de julio de 2025 |

**2. Resumen Ejecutivo**

App ShopSecure es una plataforma de comercio electrónico para Retail Chile S.A. que permite a usuarios registrados comprar productos en línea. El objetivo de esta evaluación fue identificar vulnerabilidades relevantes y proponer acciones de mitigación.

***Principales hallazgos:***

* VULN-001: Inyección SQL en endpoint /api/products (Crítico)
* VULN-002: XSS reflejado en formulario de búsqueda (Alto)

***Recomendaciones clave:***

* Implementar consultas parametrizadas en todas las consultas SQL.
* Escapar o sanitizar entradas de usuario en formularios y parámetros GET.

***Nivel de riesgo general***: Alto

**3. Alcance de la Evaluación**

***Tipo de aplicación:***  
E-commerce (carrito de compras, gestión de usuarios, pasarela de pago).

***Entorno evaluado***:  
Preproducción – versión 2.3.1

***Tecnologías involucradas***:

* Frontend: React 18, Bootstrap 5
* Backend: Node.js 16, Express
* Servidor web: Nginx 1.22
* Base de datos: PostgreSQL 13

***Roles evaluados***:

* Usuario anónimo
* Usuario autenticado (cliente)
* Usuario administrador

***Restricciones o limitaciones***:

* No se evaluaron integraciones con terceros (pasarelas de pago externas).
* Horario de pruebas: 09:00–12:00 (lunes a viernes).

**4. Metodología**

***Estándares y marcos utilizados***:

* OWASP Testing Guide v4
* OWASP Top 10 – 2021
* NIST SP 800-115 (Technical Guide to Information Security Testing)
* GPEN / PTES

***Técnicas aplicadas***:

* Pruebas dinámicas (DAST) con OWASP ZAP
* Pruebas manuales de lógica de negocio
* Revisión rápida de configuraciones de Nginx
* Análisis de respuestas HTTP

***Herramientas utilizadas***:

* OWASP ZAP 2.11.1
* Burp Suite Community Edition 2025.1
* Nmap 7.93
* sqlmap 1.6

**5. Hallazgos de Seguridad**

***5.1 VULN-001: Inyección SQL en /api/products***

* ***Identificador***: VULN-001
* ***Descripción***: El parámetro category\_id no está parametrizado, permitiendo inyección de SQL.
* ***Clasificación***: OWASP A03:2021 – Injection / CWE-89 / CVSS v3.1: 9.1 (Crítico)
* ***Impacto potencial***: Exposición de datos sensibles, modificación o eliminación de registros.
* ***Evidencia técnica***:
  + Payload: ?category\_id=1%20OR%201=1;--
  + Respuesta HTTP 200 con listado completo de productos de todas las categorías.
* ***Reproducción paso a paso***:
  + Enviar solicitud GET a /api/products?category\_id=1 OR 1=1;--.
  + Observar respuesta con catálogo completo.
* ***Recomendación de mitigación***:
  + Utilizar consultas parametrizadas en el ORM (por ejemplo, pg o Sequelize).
  + Validar y sanitizar entradas numéricas.
* ***Nivel de riesgo***: Crítico

***5.2 VULN-002: Cross-Site Scripting (XSS) reflejado en búsqueda***

* ***Identificador***: VULN-002
* ***Descripción***: El parámetro q de /search se refleja sin escape en el HTML.
* ***Clasificación***: OWASP A07:2021 – XSS / CWE-79 / CVSS v3.1: 7.4 (Alto)
* ***Impacto potencial***: Robo de cookies, ejecución de scripts maliciosos en el navegador.
* ***Evidencia técnica***:
  + *Payload en barra de búsqueda*: <script>alert('XSS')</script>
  + Al cargar resultados, el script se ejecuta.
* Reproducción paso a paso:
  + Acceder a /search?q=<script>alert('XSS')</script>.
  + Verificar ejecución del alert en la vista de resultados.
* ***Recomendación de mitigación***:
  + Escapar caracteres <, > y comillas en la salida HTML.
  + Implementar una librería de sanitización como DOMPurify.
* ***Nivel de riesgo***: Alto

**6. Evaluación de Riesgos**

***6.1 Matriz de Riesgo***

| **Riesgo** | **Probabilidad** | **Impacto** | **Nivel de Riesgo** | **Recomendación Prioritaria** |
| --- | --- | --- | --- | --- |
| VULN-001 | Alta | Alta | Crítico | Parametrizar consultas inmediatamente |
| VULN-002 | Media | Alta | Alto | Sanitizar entradas de búsqueda |

***6.2 Resumen por Categoría OWASP***

| **Categoría OWASP** | **Vulnerabilidades Detectadas** | **Severidad Promedio** |
| --- | --- | --- |
| A03:2021 Injection | 1 | Crítico |
| A07:2021 Cross-Site Scripting (XSS) | 1 | Alto |

**7. Recomendaciones Generales**

* Implementar un **WAF** (Web Application Firewall) para filtrar patrones de inyección.
* Configurar **Content Security Policy (CSP)** para mitigar XSS.
* Forzar HTTPS con HSTS en todas las rutas de producción.
* Realizar **análisis de dependencias** periódicamente (npm audit, Snyk).
* Capacitación continua en seguridad para desarrolladores (DevSecOps).

**8. Conclusiones**

La aplicación ShopSecure presenta vulnerabilidades críticas y de alto riesgo que podrían comprometer datos de usuarios y la integridad del sistema. La priorización de correcciones debe centrarse en la inyección SQL y XSS reflejado. Una vez mitigadas, se recomienda revalidar con pruebas automáticas y manuales, y establecer un ciclo de pruebas continuas.

**9. Anexos**

***Anexo A – Detalles de herramientas utilizadas***  
Listado de versiones, parámetros y configuración de escaneo.

***Anexo B – Resultados de escaneos automáticos***  
Archivos JSON/HTML exportados desde OWASP ZAP y Burp Suite.

***Anexo C – Payloads y scripts de prueba***  
Scripts de sqlmap y ejemplos de payloads para XSS.

***Anexo D – Evidencias gráficas***  
Capturas de pantalla de respuestas HTTP y alertas de XSS.

***Anexo E – Glosario de términos***  
Definiciones de CVE, CWE, CVSS, DAST, SAST, etc.

***Anexo F – Referencias normativas***

* OWASP Testing Guide v4
* NIST SP 800-115
* ISO/IEC 27001