

Resumen

Este documento presenta veinticinco (25) casos prácticos diseñados para ser ejecutados en el entorno OWASP Juice Shop con fines educativos. Cada caso incluye la identificación de la vulnerabilidad, el objetivo de aprendizaje, los pasos detallados para acceder y resolver el caso en Juice Shop y la evidencia esperada. El propósito es facilitar actividades de aprendizaje práctico sobre las categorías del OWASP Top 10 (2021) y otras buenas prácticas de desarrollo seguro.

Introducción

La enseñanza de la seguridad del software se beneficia significativamente de ejercicios prácticos que permitan a los estudiantes interactuar con vulnerabilidades reales en un entorno controlado. OWASP Juice Shop es una aplicación intencionalmente vulnerable que facilita este tipo de aprendizaje. Los ejercicios descritos en este documento están diseñados para promover la comprensión conceptual y la habilidad técnica en la identificación, explotación y, fundamentalmente, la mitigación de vulnerabilidades.

Instrucciones generales para la actividad

Valoración: 10% del total del curso; cada caso tiene un valor de 1%.

Participación: Todos los miembros del equipo deben participar en la selección de los casos.

Cada miembro del equipo debe subir su documento con la solución de los 2 casos que le correspondieron, en su documento debe indicar cuales son los casos seleccionados por el equipo y cuales le corresponden a cada uno.

Requisitos técnicos: Cada integrante debe tener instalado Juice Shop. Se recomiendan dos opciones de instalación:

1. Con Docker: ejecutar `docker run --rm -p 3000:3000 bkimminich/juice-shop` y acceder a `http://localhost:3000`.
2. Sin Docker: clonar el repositorio oficial `https://github.com/juice-shop/juice-shop` y seguir las instrucciones de instalación (Node.js).

Selección de casos: Cada grupo debe seleccionar **diez (10)** casos de los veinticinco disponibles y resolverlos (2 casos por persona).

Investigación y uso de IA: Para resolver cada caso, es posible que se requiera investigación adicional. Se permite el uso de herramientas de inteligencia artificial siempre y cuando en la entrega se incluya el *prompt* completo utilizado para cada caso.

Entrega de evidencias: En un único documento (formato libre) se deben presentar las capturas de pantalla que muestren el paso a paso utilizado para resolver cada caso y los *prompts* de IA empleados. Además, para cada caso se debe incluir:

- Vulnerabilidad identificada.
- Objetivo de aprendizaje.
- Pasos realizados para entrar a Juice Shop y resolver el caso.
- Evidencia esperada (indicador de logro).

Casos prácticos

A continuación se presentan los veinticinco (25) casos prácticos. Para cada caso se mantiene la siguiente estructura: **Título del caso, Objetivo de aprendizaje, Pasos para entrar a Juice Shop y resolver el caso y Indicador de logro / Evidencia esperada.**

Caso 1: Acceso no autorizado al panel de administración (Control de Acceso Roto)

Objetivo de aprendizaje: Comprender la importancia del control de acceso basado en roles.

Pasos para entrar a Juice Shop y resolver el caso:

- 1) Abre Juice Shop en tu navegador.
- 2) Intenta navegar a /administration o a la ruta 'Administration' si está escondida en el menú.
- 3) Si no estás autenticado, registra un usuario normal o usa uno existente.
- 4) Modifica la URL a /administration o intenta acceder con herramientas de requests.
- 5) Observa si la página muestra funciones administrativas.

Payloads / pasos técnicos: acceder a <http://localhost:3000/#/administration> o usar curl a /rest/admin.

Indicador de logro / Evidencia esperada: Visualizar la página de administración o endpoints administrativos sin credenciales.

Caso 2: Manipulación de ID para acceder a recursos de otros usuarios (IDOR)

Objetivo de aprendizaje: Identificar el impacto de referencias directas inseguras (IDOR).

Pasos para entrar a Juice Shop y resolver el caso:

- 1) Inicia sesión con un usuario A.
- 2) Ve a tu perfil o pedidos y copia la URL que contiene el ID (por ejemplo, /#/order/5).
- 3) Sustituye el ID por otro (por ejemplo 4, 3) y recarga.
- 4) Observa si puedes ver/editar datos que pertenecen a otro usuario.

Indicador de logro / Evidencia esperada: Acceso a detalles o pedidos pertenecientes a otro usuario.

Caso 3: Inyección SQL en el formulario de login (Inyección)

Objetivo de aprendizaje: Comprender cómo la entrada no validada permite manipular consultas SQL.

Pasos para entrar a Juice Shop y resolver el caso: 1) En la página de login intenta usar como usuario: `admin' --` o el campo de password con `` OR 1=1 --`.

2) Alternativamente, prueba payloads clásicos: `` OR '1'='1`.

3) Observa si logras iniciar sesión sin credenciales válidas.

Indicador de logro / Evidencia esperada: Inicio de sesión exitoso usando payload de inyección.

Caso 4: Inyección NoSQL en búsqueda de productos (Inyección NoSQL)

Objetivo de aprendizaje: Reconocer que bases NoSQL también son susceptibles a inyección.

Pasos para entrar a Juice Shop y resolver el caso: 1) Usa la búsqueda avanzada o endpoints que acepten JSON.

2) Intercepta la petición y modifica el cuerpo con payloads NoSQL como `{ "\$ne": null }` en campos de búsqueda.

3) Observa resultados no esperados o filtrado incorrecto.

Indicador de logro / Evidencia esperada: Obtención de resultados no autorizados o bypass de filtros.

Caso 5: XSS reflejado en el buscador (Cross-Site Scripting)

Objetivo de aprendizaje: Identificar vulnerabilidad por entrada no sanitizada en parámetros reflejados.

Pasos para entrar a Juice Shop y resolver el caso: 1) En el cuadro de búsqueda ingresa `` y ejecuta la búsqueda.

2) Observa si el alert se ejecuta.

3) Usa herramientas del navegador para ver la respuesta HTML.

Indicador de logro / Evidencia esperada: Ejecución de script en el navegador (alert o comportamiento visible).

Caso 6: XSS almacenado en comentarios (XSS Persistente)

Objetivo de aprendizaje: Ver impacto persistente de entradas maliciosas que se almacenan en la base de datos.

Pasos para entrar a Juice Shop y resolver el caso: 1) Publica un comentario o review con `

2) Navega a la página donde se listan los comentarios.

3) Observa ejecución del payload al cargar la página.

Indicador de logro / Evidencia esperada: Script se ejecuta para cualquier visitante de la página del comentario.

Caso 7: Modificación del precio en el carrito (Integridad de Datos)

Objetivo de aprendizaje: Comprender la importancia de validar precios en servidor.

Pasos para entrar a Juice Shop y resolver el caso: 1) Agrega un producto al carrito.

2) Abre DevTools > Network > selecciona la petición que añade o actualiza el carrito.

3) Intercepta y modifica el precio en la petición (o modifica el DOM y el hidden input del precio).

4) Continúa con la compra y verifica si el pedido se procesa con el precio manipulado.

Indicador de logro / Evidencia esperada: Confirmación de compra con precio menor/manipulado.

Caso 8: Compra de producto sin completar pago (Lógica de Negocio)

Objetivo de aprendizaje: Identificar bypass en la lógica de pago que permite confirmar pedidos sin transacción válida.

Pasos para entrar a Juice Shop y resolver el caso: 1) Inicia una compra y en la etapa de pago interrumpe o manipula la confirmación (por ejemplo, modifica el estado en la petición final).

2) Observa si el pedido se registra como 'pagado'.

3) Prueba con distintos flujos (simular callbacks del gateway con curl).

Indicador de logro / Evidencia esperada: Pedido marcado como completado sin pago real.

Caso 9: Subida de archivos sin validación (RCE/Desbordamiento)

Objetivo de aprendizaje: Detectar falta de validación en archivos subidos y sus riesgos.

Pasos para entrar a Juice Shop y resolver el caso: 1) Ve a la funcionalidad que permita subir avatar o archivos.

2) Intenta subir un archivo con doble extensión `shell.php.jpg` o un archivo .js renombrado.

3) Observa si el archivo es aceptado y si se puede acceder/executar desde la URL

generada.

Indicador de logro / Evidencia esperada: Archivo subido accesible o ejecutable desde el servidor.

Caso 10: Exposición de claves o tokens en recursos públicos (Datos Sensibles)

Objetivo de aprendizaje: Detectar exposición de información sensible en recursos accesibles.

Pasos para entrar a Juice Shop y resolver el caso: 1) Explora rutas públicas, archivos de prueba o endpoints de documentación (como /ftp, /assets, /api-docs).
2) Busca configuraciones, tokens o credenciales visibles.

Indicador de logro / Evidencia esperada: Encontrar credenciales, tokens o claves expuestas.

Caso 11: Enumeración de usuarios por mensajes (Identificación y Autenticación)

Objetivo de aprendizaje: Reconocer filtración de existencia de usuarios a través de mensajes de error.

Pasos para entrar a Juice Shop y resolver el caso: 1) En el formulario de login o recuperación de contraseña prueba varios nombres de usuario.
2) Registra diferencias en respuestas (tiempo, mensajes 'usuario no encontrado' vs 'contraseña incorrecta').

Indicador de logro / Evidencia esperada: Determinar si un usuario existe por la respuesta del sistema.

Caso 12: Sesión persistente después del logout (Gestión de Sesiones)

Objetivo de aprendizaje: Evaluar controles de sesión y revocación de tokens.

Pasos para entrar a Juice Shop y resolver el caso: 1) Inicia sesión, copia el token o cookie de sesión.
2) Cierra sesión desde la UI.
3) Intenta usar el token/cookie copiado para acceder a recursos protegidos.

Indicador de logro / Evidencia esperada: Acceso autorizado usando token después del logout.

Caso 13: Falla en límites de intentos de autenticación (Brute Force)

Objetivo de aprendizaje: Comprender la importancia del rate limiting y bloqueo de intentos.

Pasos para entrar a Juice Shop y resolver el caso: 1) Intenta múltiples intentos de

login con credenciales incorrectas.

2) Observa si el sistema impone retrasos o bloqueos.

3) Puedes automatizar con un script simple para probar resistencia.

Indicador de logro / Evidencia esperada: Sistema permite numerosos intentos sin mitigación.

Caso 14: Error detallado del servidor (Información Sensible)

Objetivo de aprendizaje: Comprender por qué no se deben exponer stack traces al usuario final.

Pasos para entrar a Juice Shop y resolver el caso: 1) Provoca un error enviando caracteres inválidos o peticiones malformadas.

2) Revisa la respuesta para contenido técnico (stack trace, rutas, versiones).

Indicador de logro / Evidencia esperada: Visualizar stack trace o detalles internos en la respuesta.

Caso 15: Uso de bibliotecas inseguras (Dependencias con CVE)

Objetivo de aprendizaje: Aprender a identificar dependencias con vulnerabilidades conocidas.

Pasos para entrar a Juice Shop y resolver el caso: 1) Revisa archivos expuestos que contengan versiones (package.json / package-lock.json si están accesibles).

2) Identifica versiones antiguas y compara con bases de datos de CVE (fuera de Juice Shop).

Indicador de logro / Evidencia esperada: Identificar una dependencia vulnerable con CVE conocida.

Caso 16: SSRF al consultar URLs externas (SSRF)

Objetivo de aprendizaje: Entender SSRF y su impacto en la red interna.

Pasos para entrar a Juice Shop y resolver el caso: 1) Busca funcionalidades que acepten URL (import, thumbnail generation).

2) Introduce `http://127.0.0.1:80` o `http://localhost` o IP de la red interna.

3) Observa si la aplicación realiza la solicitud y muestra la respuesta.

Indicador de logro / Evidencia esperada: Servidor realiza peticiones a recursos internos y devuelve información.

Caso 17: Validación solo en cliente (Bypass de validaciones)

Objetivo de aprendizaje: Mostrar por qué las validaciones deben implementarse en servidor.

Pasos para entrar a Juice Shop y resolver el caso: 1) Desactiva JavaScript en el navegador o modifica los atributos de formulario con DevTools.
2) Envía datos que el cliente debería bloquear (campos largos, caracteres inválidos).
3) Observa si el servidor acepta los datos.

Indicador de logro / Evidencia esperada: Aceptación de entradas no válidas cuando JS está deshabilitado.

Caso 18: Formularios sin HTTPS (Transporte inseguro)

Objetivo de aprendizaje: Evaluar exfiltración de credenciales por tráfico no cifrado.

Pasos para entrar a Juice Shop y resolver el caso: 1) Asegura que tu instancia use HTTP (solo pruebas locales en red controlada).
2) Observa tráfico con proxy/wireshark para ver si credenciales van en claro.

Indicador de logro / Evidencia esperada: Credenciales visibles en texto plano en el tráfico.

Caso 19: Escalada de privilegios mediante modificación de token (Control de Privilegios)

Objetivo de aprendizaje: Detectar falta de verificación de roles y checks en servidor.

Pasos para entrar a Juice Shop y resolver el caso: 1) Inspecciona el token JWT si existe (DevTools > Storage > cookies/localStorage).
2) Modifica el campo `role` o `isAdmin` en el token y reenvía la petición (si no está firmado o verificado correctamente).
3) Observa si obtienes capacidades administrativas.

Indicador de logro / Evidencia esperada: Acceso a funciones administrativas tras modificar token.

Caso 20: Endpoint oculto descubierto por fuzzing o /robots.txt (Rutas ocultas)

Objetivo de aprendizaje: Comprender riesgos de endpoints no documentados o expuestos accidentalmente.

Pasos para entrar a Juice Shop y resolver el caso: 1) Revisa /robots.txt y sitemap si están disponibles.

2) Usa una herramienta de fuzzing ligero o wordlist sobre rutas comunes (/backup, /admin, /old).

3) Accede a endpoints descubiertos y evalúa su seguridad.

Indicador de logro / Evidencia esperada: Encontrar endpoint funcional sin autenticación o con baja protección.

Caso 21: Logs accesibles públicamente (Exposición de registros)

Objetivo de aprendizaje: Identificar riesgos de tener logs accesibles desde la web.

Pasos para entrar a Juice Shop y resolver el caso: 1) Busca archivos con extensión .log o rutas comunes (/logs, /var/logs) accesibles.

2) Revisa contenido por datos sensibles (stack traces, tokens).

Indicador de logro / Evidencia esperada: Encontrar logs con información sensible expuesta.

Caso 22: Manipulación de parámetros API (API Abuse)

Objetivo de aprendizaje: Revisar la validez y robustez de endpoints API frente a datos maliciosos.

Pasos para entrar a Juice Shop y resolver el caso: 1) Intercepta llamadas API con Burp o DevTools.

2) Modifica parámetros (IDs, cantidades, precios) y reenvía.

3) Observa respuesta del servidor y si acepta cambios no autorizados.

Indicador de logro / Evidencia esperada: Servidor acepta parámetros manipulados y ejecuta acciones no autorizadas.

Caso 23: Fuerza bruta en endpoints expuestos (Automatización de ataques)

Objetivo de aprendizaje: Evaluar cómo un endpoint abierto puede ser atacado a escala.

Pasos para entrar a Juice Shop y resolver el caso: 1) Identifica endpoint vulnerable (login, reset-password, token endpoint).

2) Usa una herramienta simple de fuerza bruta (burp intruder, hydra, script) con una wordlist.

3) Observa si el endpoint no tiene mitigaciones.

Indicador de logro / Evidencia esperada: Éxito al adivinar credenciales o abusar del endpoint sin limitaciones.