🛠️ Ejercicio Práctico

📌 **Título:** Identificación y prueba manual de una vulnerabilidad de tipo XSS en DVWA

🎯 Objetivo:

Realizar una prueba controlada de **Cross-Site Scripting (XSS)** en un campo de entrada de la aplicación DVWA, utilizando un payload simple para verificar si el sistema refleja o ejecuta código malicioso en el navegador.

🧱 Escenario:

Estás trabajando en un entorno de pruebas local con la aplicación **DVWA** (Damn Vulnerable Web Application) configurada en nivel de seguridad **Low**. Se te pide probar si el módulo de XSS refleja código sin validación.

🔍 Actividades:

✅ Paso 1 – Accede al entorno

1. Inicia sesión en **DVWA** con credenciales predeterminadas (`admin / password`).
2. Asegúrate de que el **Security Level** esté configurado en **Low**.

✅ Paso 2 – Navega al módulo vulnerable

1. En el menú lateral izquierdo, selecciona **"XSS (Reflected)"**.
2. Observa el campo de texto destinado a recibir datos del usuario.

✅ Paso 3 – Ejecuta una prueba básica

1. En el campo de entrada, ingresa el siguiente código:

```
<script>alert('XSS')</script>
```

2. Presiona “Submit” o “Enviar”.

✅ Paso 4 – Observa el comportamiento

- Si el navegador muestra una ventana emergente (alert), entonces la vulnerabilidad existe.
- Si el código aparece como texto sin ejecutarse, puede que esté protegido o mal interpretado.

📋 Entregables:

1. Descripción del comportamiento observado.
2. Captura de pantalla del resultado.
3. Explicación de qué ocurrió y por qué es una falla de seguridad.
4. Recomendación básica para mitigar este tipo de vulnerabilidad.

🧠 Reflexión Final:

¿Por qué el navegador ejecutó el código en lugar de mostrarlo como texto?
¿Qué pasaría si ese script robara cookies o redirigiera al usuario a otro sitio?
¿Qué medidas puede implementar el desarrollador para evitar que esto suceda?

✅ Solución Modelo – Ejercicio Práctico

📌 Identificación y prueba manual de una vulnerabilidad de tipo XSS en DVWA

🧪 Paso 1 – Acceso y configuración del entorno

- Se accedió correctamente a la aplicación DVWA mediante la URL local `http://localhost/dvwa`.
- El usuario inició sesión con las credenciales:
    - **Usuario:** admin
    - **Contraseña:** password
- En la opción **DVWA Security**, el nivel fue configurado en **Low**.

🔍 Paso 2 – Exploración del módulo vulnerable

- En el menú de DVWA, se seleccionó la sección **XSS (Reflected)**.
- Se identificó un campo de entrada y un botón de envío que ejecuta una respuesta en la misma página, basada en los datos ingresados por el usuario.

🧪 Paso 3 – Prueba de vulnerabilidad

- Se ingresó el siguiente payload en el campo de entrada:

```
<script>alert('XSS')</script>
```

- Tras presionar el botón **Submit**, el navegador mostró una **ventana emergente** (alerta con el mensaje “XSS”).

🗒️ Observaciones:

- El sistema reflejó el contenido ingresado **sin ninguna sanitización o validación**.
- El navegador interpretó el código como HTML válido, lo que confirma la **presencia de una vulnerabilidad de tipo XSS reflejado**.

📸 Captura de pantalla:

(Aquí se insertaría una imagen de la alerta emergente con el mensaje "XSS")

Simulación: [ ]

🛡️ Recomendación para mitigación:

Para evitar este tipo de vulnerabilidad, se deben aplicar las siguientes medidas:

1. **Escapar o codificar caracteres especiales** (`<`, `>`, `"` y `'`) antes de reflejar contenido en la interfaz.
2. Utilizar funciones de sanitización de entrada y salida (ej. `htmlspecialchars()` en PHP).
3. Implementar una **Política de Seguridad de Contenidos (Content-Security-Policy)** que limite la ejecución de scripts no autorizados.
4. Validar entradas tanto en el **frontend** como en el **backend**.

🧠 Reflexión final del ejercicio:

Esta actividad demostró cómo una simple omisión en el tratamiento de entradas del usuario puede permitir la ejecución de código no deseado en el navegador. Aunque este ejemplo se limita a una alerta, un atacante real podría haber robado cookies, redirigido al usuario a un sitio falso o comprometido sesiones. La validación de entradas no es solo una buena práctica: **es una necesidad de seguridad crítica**.
