Nombre Taller:Laboratorio 14

Laboratorio XSS 0x02 – Stored XSS

Detalles Prácticos:

Descripción: Obtenermos las cookies de sesion del usuario Víctima
Nivel(Básico/Intermedio o avanzado)
Instructores: Rieradipe

Tabla de contenidos:

Objetivo:

Explotar una vulnerabilidad Stored XSS para ejecutar código JS malicioso que se almacena en el servidor y se ejecuta cuando otro usuario accede a la página afectada. En este caso, el objetivo es obtener las **cookies de sesion** del usuario víctima.

Análisis del comportamiento de la aplicación:

1. La aplicación permite añadir ideas o comentarios que se almacenan y luego se reflejan en pantalla.
2. No existe validación ni sanitización del contenido HTML introducido por el usuario
3. Al introducir etiquetas <script> estas se almacenan y se ejecutan al ser visualizadas por otros usuarios

Ejecución del ataque:

Preparación: Simulación de dos usuarios

Se utilizan extensiones de FIREFOX para simular dos usuarios distintos

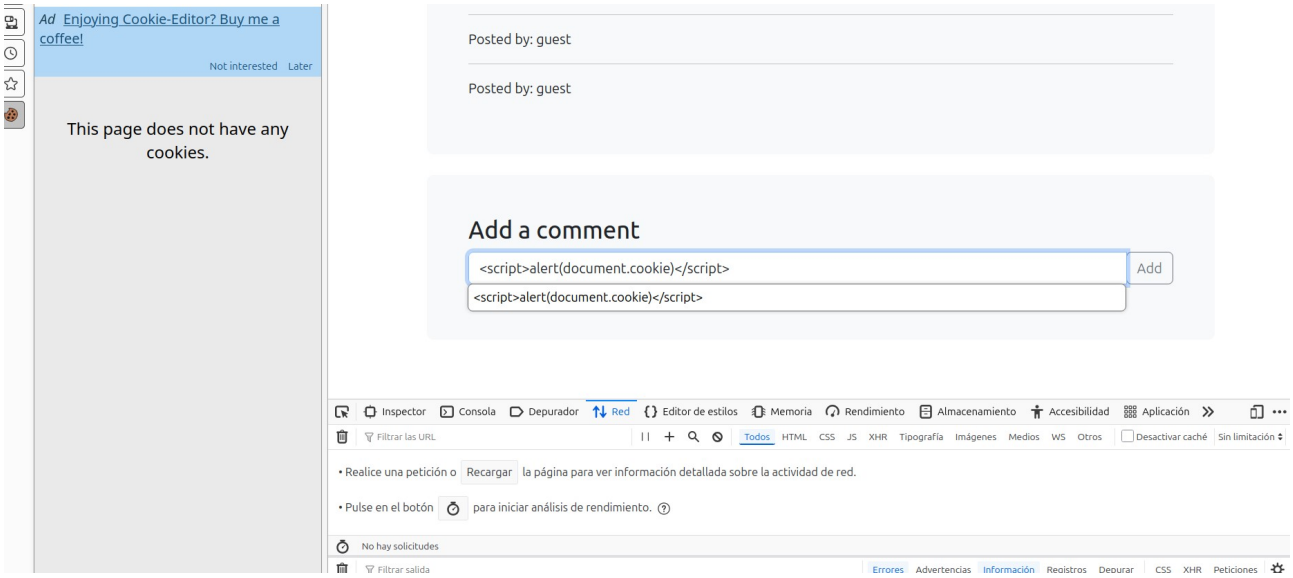
- Cookie Editor: para insertar manualmente una cookie de sesión
- Multi-Account Containers: para aislar la sesión del atacante y la victima

Contenedor Trabajo (atacante)

- Se abrio el laboratorio desde un contenedor aislado
- En el campo de comentarios, se incluye esta payload malicioso

```
<script>alert(document.cookie)</script>
```

- Este payload se almacena en la bbdd

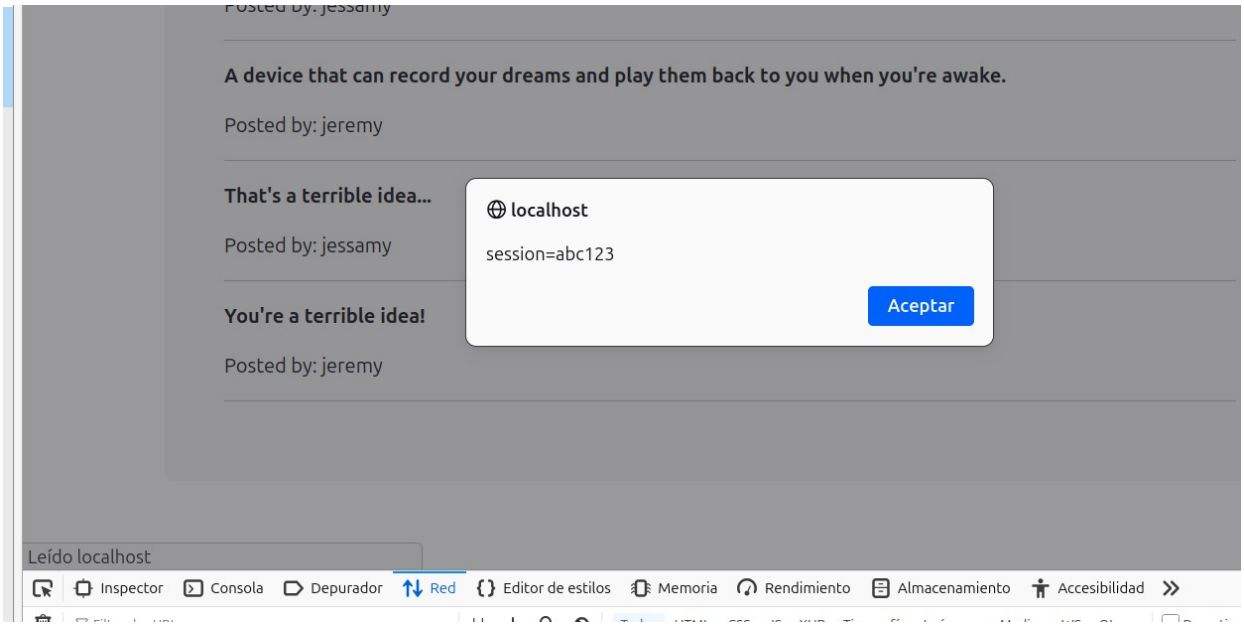


Contenedor Casa (víctima)

- Se abre el mismo laboratorio desde otro contenedor con la cookie se de sesion configurada

Nombre: session	Valor: abc123

- Al visitar la página, el script se ejecuta automáticamente y muestra el contenido de la cookie en un alert
-



Resultado:

El ataque fue exitoso: se almacena un payload de XSS persistente, y al ser visitado desde otra sesión, el navegador ejecuta el script malicioso que accede a las cookies del usuario.

Recomendaciones de mitigación

1. Marcar cookies sensibles con la bandera HttpOnly

   Previene que document.cookie pueda leerlas desde JS

2. Sanitizar entradas del usuarios

   Evita guardar contenido sin filtrar en la bbdd
   Escapar o eliminar etiquetas peligrosas(<script> onerror, onload)

3. Escapar contenido al renderizar

   Usa funciones específicas por lenguaje

   * PHP: htmlspecialchars()
   * Python: html.escape()
   * Java: StringEscapeUtils.escapeHTML4()

Conclusión:

Este laboratorio demuestra cómo una vulnerabilidad Stored puede comprometer la seguridad de las sesiones si la aplicación no valida ni protege las entradas del usuario. Un atacante puede robar cookies, suplantar identidad o escalar privilegios solo con almacenar un comentario malicioso