

Nombre Taller: lab 11

Inyección de Segunda Orden

Detalles Prácticos:

Descripción: Explotar una vulnerabilidad de tipo SQL Injection segunda orden

Nivel(<u>Básico</u>/Intermedio o avanzado)

Instructores:Rieradipe

Tabla de contenidos:

S

Objetivo:

Explotar una vulnerabilidad de tipo **SQL injection de segundo orden,** en el cual el payload no se ejecuta el el momento de la inyección, sino que queda almacenado en la bbdd y se evalúa en una acción posterior(como el login)

1. Análisis inicial de la aplicación

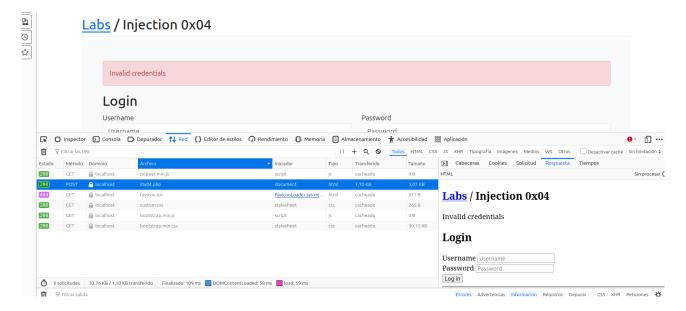
La aplicación tiene dos formularios:

- Sign Up: registra un nuevo usuario
- Login: iniciar sesión con un usuario ya registrado Ambos envían datos mediante método POST

2. Pimera prueba de login

Se intenta acceder con un payload clásico

username: 'OR 1=1 password: (daigual)



Esto indica que los datos enviados por el formulario de login probablemente **si están escapados o protegidos.**

3. <u>Inyección de la operación de registro(Sign up)</u>

Se prueba la inyección al registrar un nuevo usuario. En el campo username, se inserta el siguiente payload

testuser' OR 1=1---

Enviado como newusername=testuser' OR 1=1---

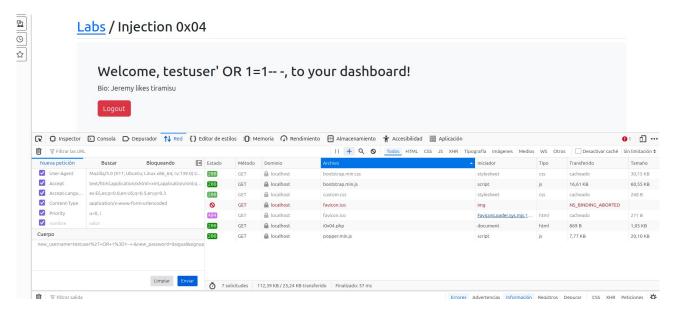
Passwor se registra como: daigual

El sistema acepta el registro, almacenando el valor tal cual en la bbdd

4. Explotación del ataque de segunda orden

Luego se intenta haer login con ese usuario que contiene el payload almacenado

username: testuser password: daigual



Resultado: Acceso exitoso.

5. Conclusión

Este laboratorio demuestra cómo un imput malicioso, si no se sanitiza ni calida al ser almacenado, puede aprovecharse en otra etapa del flujo de la app, permitiendo evadir controles o manipular datos

6. Medidas de Mitigación recomendadas

Uso de prepared statements o sentencias parametizadas
Validación y escpe de datos antes de almacenarlos
Principio de menor privilegio en consultas a la bbdd
Núnca reutilizar directamente valores almacenados sin volver a validarlos
Configurar mensajes de error genéricos