



Nombre Taller:9

SQL Injection via Cookie

Detalles Prácticos:

Descripción: Explotar vulnerabilidad de **Inyeccion SQL en la cookie**

Nivel(Básico/Intermedio o avanzado)

Instructores:Rierdipe

Tabla de contenidos:

Introducción	Contexto del taller
	Objetivos generales y específicos
Materiales Necesarios	Software requeridos
	Recursos Adicionales
Metodología	Desglose paso a paso del proceso
	Practicas recomendadas
Ejercicios Prácticos	Captura de solicitudes
	Análisis y pruebas
Resultados y Evaluación	Resultados esperados de las actividades
	Criterios de evaluación
Conclusión	Resumen de aprendizajes
	Preguntas y próximos pasos

Objetivo:

Acceder al panel de usuario sin proporcionar credenciales válidas, explotando una vulnerabilidad de **inyección SQL**.

Contexto:

La aplicación web gestiona el inicio de sesión mediante una cookie llamada session. Esta cookie no está debidamente validada, es posible manipularla con una inyeccion SQL para eludir la autenticación.

Vector de ataque:

Se modificó manualmente la cookie session usando el payload:

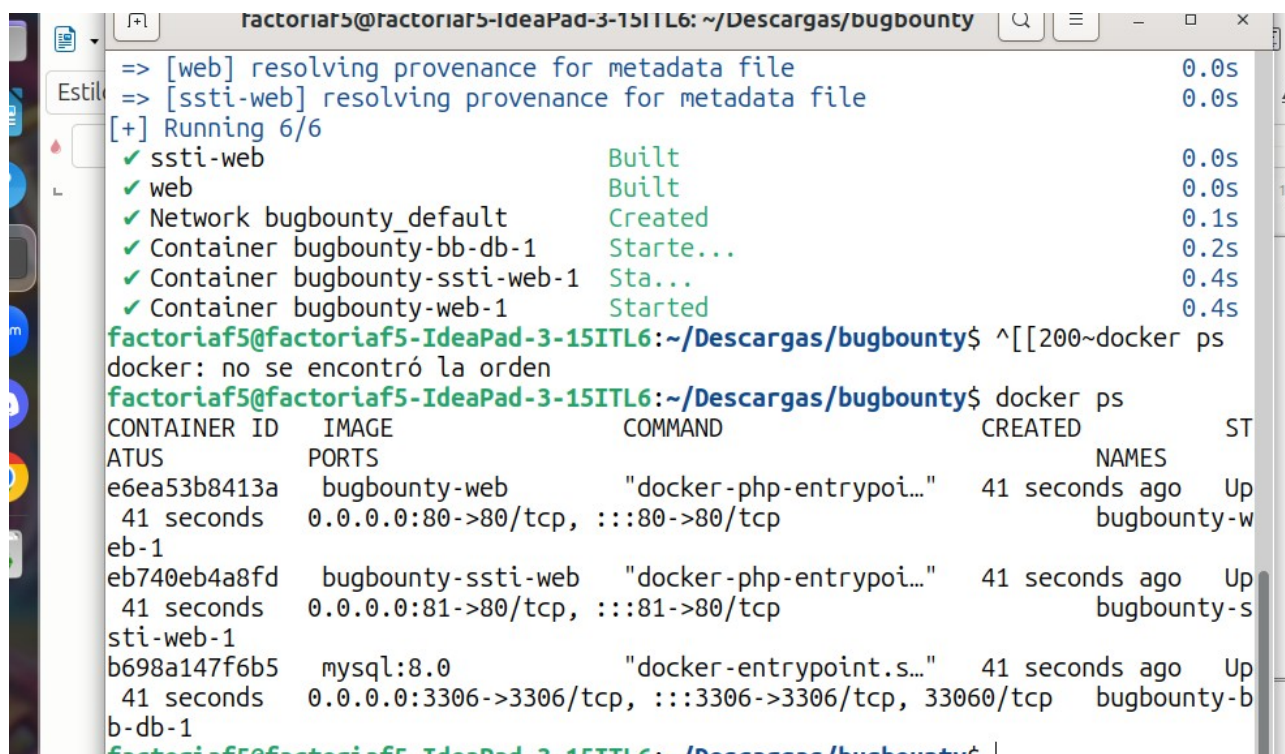
' OR 1=1#

Esto fuerza la condición SQL a ser siempre **verdadera**, permitiendo el acceso sin necesidad de usuario o contraseña.

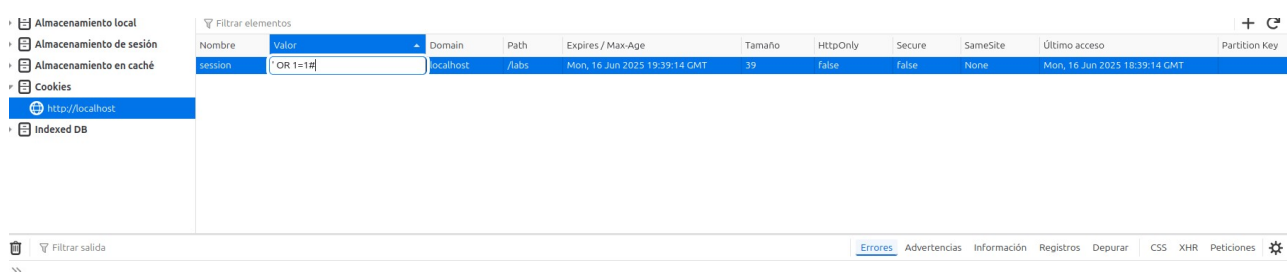
Pasos realizados:

1. Se lanzó en entorno del lab usando Docker.
2. Se accede al laboratorio
3. Desde las **herramientas de desarrollo de navegador (F12)** se accede a la pestaña **Almacenamiento** → **Cookies** → <http://localhost>
4. Se modifica el valor de la cookie session y se introduce el payload 'OR 1=1#
5. Al recargar la página, la aplicación nos redirige directamente al panel de bienvenida del dashboard, confirmando acceso no autorizado.

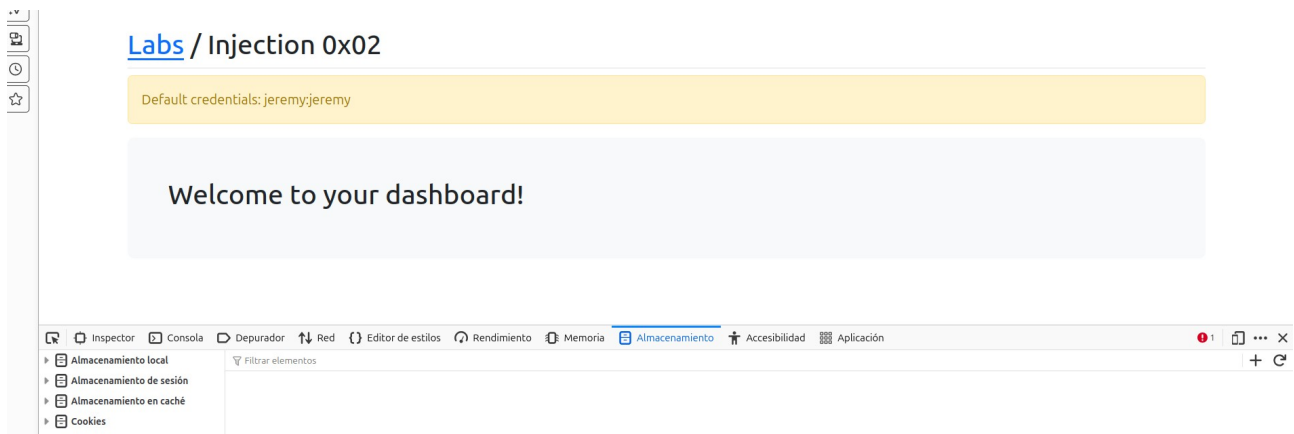
Evidencias:



```
factoriaf5@factoriaf5-IdeaPad-3-15ITL6: ~/Descargas/bugbounty
=> [web] resolving provenance for metadata file 0.0s
=> [ssti-web] resolving provenance for metadata file 0.0s
[+] Running 6/6
  ✓ ssti-web Built 0.0s
  ✓ web Built 0.0s
  ✓ Network bugbounty_default Created 0.1s
  ✓ Container bugbounty-bb-db-1 Starte... 0.2s
  ✓ Container bugbounty-ssti-web-1 Sta... 0.4s
  ✓ Container bugbounty-web-1 Started 0.4s
factoriaf5@factoriaf5-IdeaPad-3-15ITL6:~/Descargas/bugbounty$ ^[[200~docker ps
docker: no se encontró la orden
factoriaf5@factoriaf5-IdeaPad-3-15ITL6:~/Descargas/bugbounty$ docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS
ATUS          e6ea53b8413a        bugbounty-web           "docker-php-entryp... 41 seconds ago Up
41 seconds    0.0.0.0:80->80/tcp, :::80->80/tcp
eb-1          eb740eb4a8fd        bugbounty-ssti-web      "docker-php-entryp... 41 seconds ago Up
41 seconds    0.0.0.0:81->80/tcp, :::81->80/tcp
sti-web-1     b698a147f6b5        mysql:8.0               "docker-entrypoint.s... 41 seconds ago Up
41 seconds    0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp
bugbounty-bb-db-1
```



Nombre	Valor	Domain	Path	Expires / Max-Age	Tamaño	HttpOnly	Secure	SameSite	Último acceso	Partition Key
session	'OR 1=1#	localhost	/labs	Mon, 16 Jun 2025 19:39:14 GMT	39	false	false	None	Mon, 16 Jun 2025 18:39:14 GMT	



Impacto:

- Acceso total sin autenticación
- Backend confía en los datos del cliente(cookies) sin validarlos
- Podría permitir movimientos laterales o acceso a usuarios con mayores privilegios

Mitigación:

Nunca usar valores de cookies directamente en sentencias SQL

Usar consultas preparadas. **Prepared statements o OEMs**

Validar y sanitizar todas las entradas de usuario, incluso los input
