Año 2025

Desarrollo Seguro de Aplicaciones
Práctica 3 - Inyecciones

Fecha de entrega: 29/06/2025 a las 17:59

Términos relacionados:

- OWASP top 10 (**2021**): **A03 (inyección)**
- Inyección SQL (SQLI), Ejecución remota de comandos (RCE), Server side template injection (SSTI), Cross Site Scripting (XSS)

Guía para hacer la práctica:

- Continuaremos utilizando el repositorio grupal de código en GitHub y Docker con docker-compose.
- Ejecute **git pull origin main** en la carpeta del repositorio para bajar los últimos cambios.
- Ejecute los ejercicios accediendo a la carpeta **practica3** y corriendo el comando ***./run.sh***
- Para acceder vía web a cada reto deberá ingresar al puerto que corresponda, por ejemplo, para el reto *python_sqli,* http://localhost:13001

Ejercicio 1 - Explotación y fix

Para cada ejercicio explote la vulnerabilidad y genere un parche con el código que arregle la vulnerabilidad editando el código fuente de cada ejercicio en la carpeta *www*.
Realizar un push en el branch "**fix-practica3**"

Ejercicios Repositorio:
Reto *python_sqli* (http://localhost:13001)
Reto *python_rce* (http://localhost:13002)
Reto *python_ssti* (http://localhost:13003)
Reto *python_xss* (http://localhost:13004)

Nota: Evitar soluciones basadas en la confianza ciega de los datos proporcionados por el usuario sin una validación adecuada. Su solución no debe bloquear o restringir ciertos inputs de usuario. No depender únicamente de medidas de seguridad en el front-end, sino también implementar controles de seguridad en el backend.

Ejercicio 2 - Entrega de Informe

Desarrolle un informe en el que explique cómo logró explotar cada una de las vulnerabilidades del Ejercicio 1. Entregue la tarea en https://catedras.linti.unlp.edu.ar en formato PDF.

Ejercicio 3 - CTF - SQL Injection

Encuentre la flag escondida en la columna “passwd” de alguno de los usuarios, y envíela a la plataforma CTF.

a) Reto **SQLi 03 A**: https://retosql1.dsa.linti.unlp.edu.ar/index.php?name=admin

Pista: fragmento del código fuente

```
<?php
  require_once('header.php');
  require_once('db.php');
        $sql = "SELECT * FROM users where name='";
        $sql .= $_GET["name"]."'";
        $result = mysqli_query($db,$sql);
```

b) Reto **SQLi 03 B**: https://retosql2.dsa.linti.unlp.edu.ar/?name=admin

Pista: no podes poner espacios

```
if (preg_match('/ /', $_GET["name"])) {
    echo 'No podes poner espacios';
}else{
    $sql = "SELECT * FROM users where name='";
    $sql .= $_GET["name"]."'";
    $result = mysqli_query($db,$sql);
```

c) Reto **SQLi 03 C**: https://retosql3.dsa.linti.unlp.edu.ar/?id=1

```
<?php
  require_once('header.php');
        require_once('db.php');
        $sql = "SELECT * FROM users where id=";
        $sql .= mysqli_real_escape_string($db,$_GET["id"]);
        $result = mysqli_query($db,$sql);
```

d) Reto **SQLi 03 D**: https://retosql4.dsa.linti.unlp.edu.ar/?id=1

```
if (!preg_match('/^[0-9]+/', $_GET["id"])) {
                die("ERROR INTEGER REQUIRED");
        }
```

```
$sql = "SELECT * FROM users where id=";
$sql .= mysqli_real_escape_string($db,$_GET["id"]);
$result = mysqli_query($db,$sql);
```

Ejercicio 4 - CTF - XSS

a) **Basic XSS**: https://xss.dsa.linti.unlp.edu.ar/
   La flag está dividida en tres partes. Resolver /easy, /medium y /high.
b) (Opcional) **Esoteric Language**: https://xss2.dsa.linti.unlp.edu.ar/
c) (Opcional) **mutation XSS** https://xss3.dsa.linti.unlp.edu.ar