

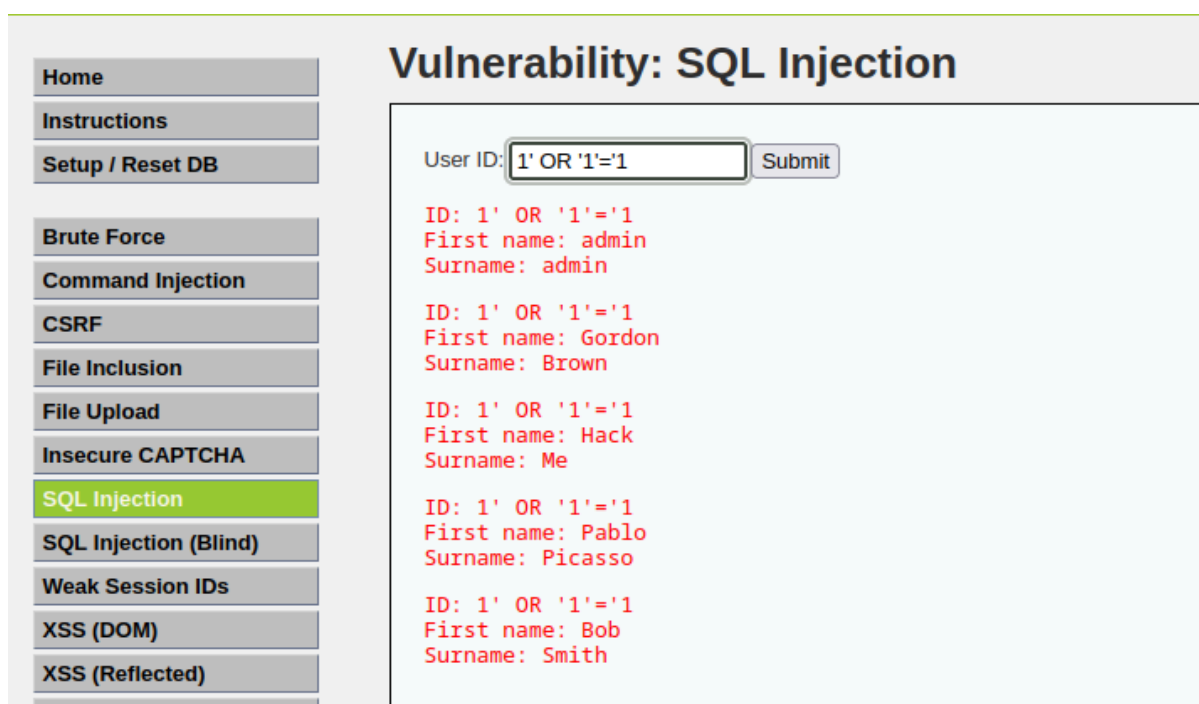
# Ataque SQL Injection en DVWA

## Introducción

Este informe documenta la ejecución de una prueba de inyección SQL en la aplicación Damn Vulnerable Web Application (DVWA), configurada en nivel de seguridad bajo. Siguiendo las normas ISO 27001 para la gestión de incidentes de seguridad de la información.

## Descripción del incidente

Durante la prueba, se identificó una vulnerabilidad en el apartado de **USER ID**, accesible después de iniciar sesión en la aplicación. Mediante una inyección SQL simple, fue posible extraer nombres de usuario y contraseñas almacenadas en la base de datos, lo que representa un riesgo significativo de seguridad.



The screenshot shows the DVWA interface with the 'SQL Injection' vulnerability selected. The 'User ID' input field contains the payload '1' OR '1'='1', and the 'Submit' button is visible. The output displays five rows of user data extracted from the database.

User ID	First name	Surname
1' OR '1'='1	admin	admin
1' OR '1'='1	Gordon	Brown
1' OR '1'='1	Hack	Me
1' OR '1'='1	Pablo	Picasso
1' OR '1'='1	Bob	Smith

## Proceso de reproducción

1. Se accedió a DVWA con la configuración de seguridad en nivel bajo.
2. Se identificó el campo **USER ID** como un posible punto vulnerable.
3. Se ingresó la siguiente sentencia SQL maliciosa para obtener todos los usuarios y contraseñas: (1' OR '1'='1)
4. Al ejecutar la consulta, la aplicación devuelve una lista completa de nombres de usuario y contraseñas.

## Impacto del incidente

En un entorno real, esta vulnerabilidad podría tener consecuencias graves, tales como:

- **Acceso no autorizado:** Un atacante podría obtener credenciales de administrador y comprometer toda la aplicación.
- **Exposición de información sensible:** Se podrían filtrar datos confidenciales de los usuarios.
- **Escalamiento de privilegios:** Un atacante podría utilizar las credenciales robadas para obtener permisos elevados y ejecutar comandos maliciosos.

## Recomendaciones

Para mitigar esta vulnerabilidad, se recomienda implementar las siguientes medidas de seguridad:

- **Uso de consultas preparadas:** Evitar la concatenación de entradas del usuario en las consultas SQL.
- **Validación y sanitización de entradas:** Asegurar que los datos ingresados por los usuarios sean verificados antes de procesarlos.
- **Uso de principios de mínimos privilegios:** Restringir el acceso a la base de datos solo a lo estrictamente necesario.
- **Implementación de WAF (Web Application Firewall):** Detectar y bloquear intentos de inyección SQL.
- **Registro y monitoreo de accesos:** Analizar patrones de comportamiento sospechoso en los intentos de autenticación.

## Conclusión

Este ejercicio demostró cómo una inyección SQL puede ser explotada para obtener información sensible de una base de datos vulnerable. Además, resaltó la importancia de aplicar buenas prácticas de seguridad, como el uso de consultas preparadas y la validación de entradas, para prevenir ataques similares. La seguridad en bases de datos es fundamental para proteger la integridad y confidencialidad de la información almacenada, evitando así posibles brechas que podrían comprometer a una organización.