Informe de Incidente — Inyección SQL (SQLi) en DVWA

Autor: Alex Winter
Fecha: 2025-10-19
Entorno: Máquina virtual Debian (VirtualBox). Apache2 + PHP 8.2 + MariaDB. Damn Vulnerable Web Application (DVWA) — Security: Low.
URL de prueba:** http://localhost/DVWA/vulnerabilities/sqli/

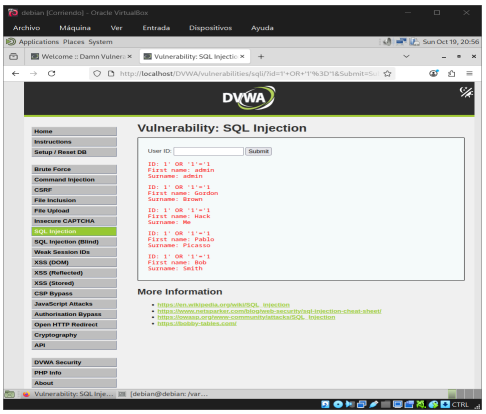
Inyección SQL (SQLi) en el parámetro “User ID” del módulo SQL Injection de DVWA

2. Introducción
Durante una práctica controlada en una instancia local de DVWA se realizó una prueba de seguridad orientada a identificar vulnerabilidades de inyección SQL. El objetivo de este informe es documentar la vulnerabilidad encontrada, detallar cómo se reprodujo, mostrar la evidencia recogida, evaluar su impacto y proponer medidas de mitigación priorizadas.

3. Descripción del Incidente
El módulo “SQL Injection” de DVWA presenta una vulnerabilidad de inyección SQL clásica: el parámetro `id` (etiquetado como *User ID*acepta entrada de usuario que se concreta directamente en la consulta SQL sin validación ni uso de prepared statements. Debido a esto, es posible manipular la lógica de la consulta y forzar la devolución de múltiples registros de la base de datos, incluyendo datos de usuarios.

4. Proceso de Reproducción (Paso a Paso / PoC)
“Requisitos previos:” Acceso local a la VM y DVWA en Security: Low.
Credenciales usadas: Usuario `admin` / Contraseña `password`.

1. Acceder a DVWA e iniciar sesión con las credenciales por defecto (`admin` / `password`).
2. Abrir el módulo “SQL Injection” (`http://localhost/DVWA/vulnerabilities/sqli/`).



3. En el campo “User ID” introducir el siguiente payload:

1´ OR ‘1’= ‘1

4. Hacer clic en “Submit”.

5. Resultado observable: la página devuelve múltiples filas de la tabla de usuarios, mostrando nombres (First name / Surname) repetidos para la condición forzada por el payload. (Ver evidencias adjuntas).

"http://localhost/DVWA/vulnerabilities/sqli/?id=1'%20OR%20'1'='1'%20--%20&Submit=Submit" -o /home/debian/evidencia_sqli.html

Impacto del Incidente

Exposición de datos: extracción de registros de usuarios (nombres, apellidos y potencialmente otros campos).

Severidad: Alta en un entorno real, porque la exfiltración de credenciales o hashes podría permitir acceso no autorizado, movimiento lateral o escalado de privilegios.

Condiciones agravantes: si las contraseñas extraídas están hasheadas con algoritmos débiles o si los usuarios utilizan credenciales, el riesgo aumenta significativamente.

Recomendaciones

Implementar consultas preparadas (prepared statements) con parámetros enlazados (PDO o MySQLi) y eliminar la concatenación directa de variables en las consultas SQL.

Principio de mínimo privilegio: asegurar que el usuario de la base de datos tenga únicamente los permisos necesarios (por ejemplo, SELECT sobre tablas requeridas) y no privilegios administrativos.

Registro y monitorización: auditar accesos a endpoints sensibles y configurar alertas por patrones de inyección (p. ej. presencia de comillas y comentarios en parámetros numéricos).

Pruebas posteriores: después de aplicar mitigaciones, repetir la prueba en Security: Medium/High y realizar análisis forense de logs para confirmar que no hubo exfiltración previa.

Conclusión

La prueba evidencia una vulnerabilidad de inyección SQL trivialmente explotable en el parámetro `UserID` del módulo SQL Injection de DVWA en modo **Low**. La mitigación prioritaria es reemplazar las consultas construidas por concatenación por consultas preparadas y validar/filtrar estrictamente las entradas. Tras aplicar las medidas sugeridas, se recomienda volver a realizar las pruebas para verificar la remediación y activar controles de detección para prevenir explotación en entornos productivos.