Informe Vulnerabilidad SQL Injection

En cumplimiento con ISO/IEC 27001

Introducción

Este informe documenta el hallazgo y explotación de una vulnerabilidad de inyección SQL (SQL Injection) en la aplicación Damn Vulnerable Web Application (DVWA), utilizada como entorno controlado para prácticas de hacking ético y análisis de seguridad web. La actividad se llevó a cabo con el objetivo de comprender los riesgos asociados a las inyecciones SQL, así como reforzar las buenas prácticas de desarrollo seguro.

Descripción del Incidente

Durante el análisis del módulo "SQL Injection" en DVWA, se identificó una vulnerabilidad crítica que permite a un atacante insertar código SQL malicioso en campos de entrada del usuario. Esta vulnerabilidad puede derivar en el acceso no autorizado a información sensible, manipulación de la base de datos, y filtración de datos críticos.

En niveles de seguridad bajo y medio, se logró manipular la lógica interna de las consultas SQL, analizaremos también el paso por el nivel de seguridad alto con pruebas más avanzadas como la Blind SQL Injection, donde la información se obtiene sin retroalimentación directa visible.

Proceso de Reproducción

Se utilizaron múltiples técnicas para explotar la vulnerabilidad, incluyendo inyecciones básicas y ataques por fuerza bruta basados en rangos de caracteres ASCII para descubrir el nombre de usuario de la base de datos.

Inyección SQL básica:
1' OR '1'='1

Este payload fuerza la condición a verdadera, devolviendo múltiples registros en lugar del esperado, lo que evidencia una falla de control.

Obteniendo de salida:

ID: 1
First Name: admin
Surname: admin

ID: 2

First Name: Gordon
Surname: Brown

ID: 3
First Name: Pablo
Surname: Picasso

Inyección SQL ciega con fuerza bruta:

1' AND ASCII(SUBSTRING((SELECT user FROM users LIMIT 1),2,1)) > 95 -- -
1' AND ASCII(SUBSTRING((SELECT user FROM users LIMIT 1),5,1)) > 95 -- -
1' AND ASCII(SUBSTRING((SELECT user FROM users LIMIT 1),4,1)) = 105 -- -
Estas inyecciones preguntan: ¿El primer carácter del nombre de usuario está entre la "a" (97) y la "z" (122)?
La respuesta se interpreta a partir del comportamiento del servidor (respuesta diferente = condición verdadera).

Se repitió el proceso modificando los rangos y posiciones hasta descubrir el nombre completo del usuario, carácter por carácter. Dado que este proceso es más largo se probó para sacar los primeros caracteres de cara a demostrar la inyección. Jugando con el rango obtuvimos que los primeros caracteres eran el 97 y el 100, a y d respectivamente.

Impacto del Incidente

Explotar este tipo de vulnerabilidad permite:
- Acceso no autorizado a información crítica almacenada en la base de datos.
- Modificación o eliminación de datos, afectando la integridad del sistema.
- Pérdida de confianza del usuario al percibir un fallo grave en la seguridad.
- Consecuencias legales y normativas, por incumplimiento de normativas como el RGPD o la ISO 27001.

Recomendaciones

1. Validación de Entrada
2. Consultas Preparadas (PDO en PHP)
3. Cortafuegos de Aplicación Web (WAF)
4. Auditorías de Seguridad Recurrentes
5. Educación y Concienciación
6. Principio de Mínimos Privilegios
7. Manejo Seguro de Errores

Conclusión

El ejercicio con DVWA ha demostrado cómo un SQL Injection, incluso en su forma más sencilla, puede comprometer seriamente la seguridad de una aplicación web. A través de técnicas avanzadas como la inyección ciega con fuerza bruta de rangos ASCII, fue posible exfiltrar información crítica sin una salida directa visible.

Implementando controles adecuados como la validación estricta de entradas, consultas parametrizadas y medidas defensivas como WAF y educación continua, las organizaciones pueden reducir significativamente el riesgo de este tipo de ataques.