Informe de Inyección SQL en DVWA

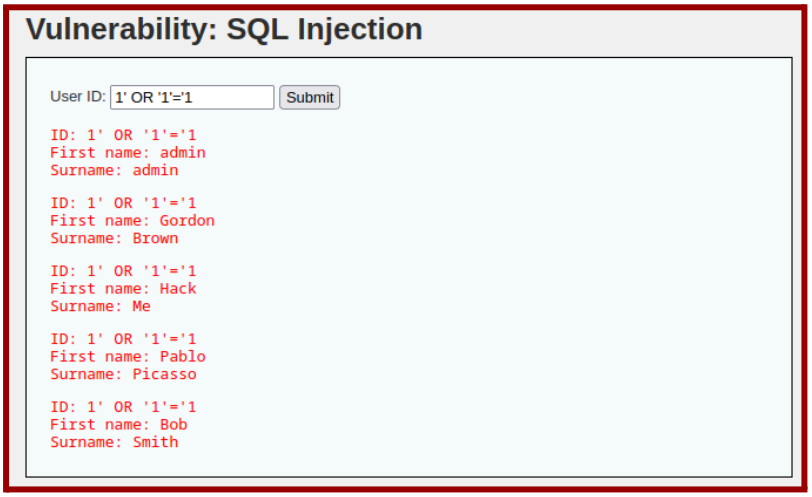
Introducción

En este informe se documenta el incidente de seguridad que ha ocurrido en la aplicación DVWA, en el que se ejecutó un payload del tipo *SQL Injection* detectada y posteriormente reproducida en un entorno seguro para su análisis.

A lo largo de este informe se recogen las características de la incidencia, así como la gravedad de la misma y métodos para prevenir y/o mitigar incidencias de este estilo en lo sucesivo.

Descripción del incidente

Durante la reproducción de la vulnerabilidad, se ejecutó el payload ***1' OR '1' = '1*** en el campo de texto de *user ID* del módulo de *SQL Injection*, lo que resultó en la devolución completa de la tabla de usuarios de la base de datos de la aplicación web:



La explotación de vulnerabilidades de este estilo puede llevar a que personas no autorizadas y/o externas a la organización puedan obtener información sensible de los usuarios sin la correcta autorización.

Proceso de reproducción

Para reproducir una vulnerabilidad de este estilo solo es necesario encontrar campos de texto que estén securizados incorrectamente o directamente sin securizar, de esta forma se podrá extraer la tabla completa a la que se esté realizando la consulta del mismo modo que se ha hecho con la tabla de usuarios en el caso de estudio.

Impacto del incidente

Explotando este tipo de vulnerabilidades los usuarios obtienen acceso no autorizado a datos empresariales sensibles, pudiendo robarlos, así como cambiar y/o borrar datos presentes en la misma base de datos modificando el *Payload* con distintas sentencias de SQL.

Esto representa un grán riesgo para la integridad y confiabilidad de los datos empresariales alojados en este servidor web, así como para todas las aplicaciones que corren en él.

Recomendaciones

Una vez acotada la gravedad e impacto del incidente, se recomiendan las siguientes medidas de seguridad para prevenir ataques que se aprovechen de esta vulnerabilidad:

- Parametrizar las consultas entrantes: Gracias a esto, los estamentos del mismo tipo al usado en esta reproducción no serán efectivos, puesto que la ejecución de la query gestionada por el servidor devolverá una excepción y no el listado de parámetros.

- Validación del formato de los parámetros: De esta forma se puede evitar que los *payload* no puedan contener sub-sentencias SQL, en el caso del exploit que se ha usado aquí, podría programarse una longitud máxima así como carácteres no permitidos, de tal forma que la respuesta del sistema fuera “ID inválido” en caso de no cumplir los requisitos.

- Uso de ORMs: Utilizar herramientas como SQLAlchemy para poder sanitizar todas las consultas entrantes a la base de datos, con esto también se evitan Inyecciones SQL como la que se ha recogido y de complejidad mayor, esto añade una capa de seguridad extra para defenderse de atacantes que dispongan de conocimientos más profundos en bases de datos y aplicaciones web.

- Auditorías regulares: El haber sanitizado las entradas a la base de datos podría no ser suficiente, con objetivo de mantener el nivel de seguridad, es preciso hacer auditorías de forma regular para poder detectar y proteger el sistema frente a nuevos tipos de amenazas.

Conclusión

El descubrimiento y uso de este tipo de vulnerabilidades genera mucha preocupación entre las empresas debido a su alta peligrosidad y amplias consecuencias, por lo que se pone de manifiesto