PROTECCIÓN ANTE ATAQUES DE INYECCIÓN SQL EN APLICACIONES WEB

Manuel Alberto López Soto[1], Ana María Delgado Burgueño[2], Manuel Iván Tostado Ramírez[3], Juan Francisco Peraza Garzón[4]

[1,2,3,4] *Facultad De Informática Mazatlán, Universidad Autónoma de Sinaloa (México)*

Abstract

Este es un trabajo que trata sobre medidas o técnicas que se pueden aplicar al momento de estar desarrollando la aplicación web, debido a que en actualidad se han reportado varios casos de ataques hacia las bases de datos de aplicaciones web mediante el uso de la técnica que lleva por nombre inyección SQL, esta tiene como objetivo las bases de datos, esto se debe a que los atacantes se aprovechan de las vulnerabilidades y buscan obtener algún beneficio de los datos obtenidos mediante este ataque, hay que recordar que lo más importante es mantener a salvo la información, porque en caso de caer en manos equivocadas, genera muchos problemas y en muchas ocasiones hasta llegar a las pérdidas monetarias.

Palabras clave: Tecnología, Informática, Vulnerabilidades, Web, SQL.

1 INTRODUCCIÓN

Actualmente el desarrollo de la tecnología informática produce un gran volumen de datos diariamente. Estos datos necesitan ser ordenados y almacenados para posteriormente poder ser usados o analizados, para esto se crearon grandes almacenes de datos llamados bases de datos.

Una base de datos es una aplicación independiente que almacena una colección de información organizada por campos, registros y archivos, de manera que se puedan seleccionar rápidamente los fragmentos de datos que se necesiten. Existen un sinnúmero de sistemas de gestión de base de datos y cada una de ellas posee una forma diferente de manejar sus datos, sin embargo, con el paso de los años, estos se fueron unificando y universalizando para dar paso a mejores técnicas y mejores formas de manejo, de esta manera nació SQL [1].

Los orígenes de SQL comienzan en 1974 con la definición por parte de Donald Chamberlin y de otras personas que trabajaban en los laboratorios de investigación de IBM, de un lenguaje para la especificación de las características de las bases de datos que adoptaban el modelo relacional. El prototipo (System R.), basado en este lenguaje, se adoptó y utilizó internamente en IBM y lo adoptaron algunos de sus clientes elegidos. Gracias al éxito de este sistema, que no estaba todavía comercializado, también otras compañías comenzaron a desarrollar sus productos relacionales basados en SQL. A partir del año 1981, IBM comenzó a entregar sus productos relacionales [2].

En 1986 el ANSI adoptó SQL como estándar de lenguajes relacionales, ahora en su tercera década de existencia, el lenguaje SQL ofrece una gran flexibilidad a los usuarios, SQL se convirtió en la base de una gran cantidad de aplicaciones de bases de datos bien establecidos en internet hoy en día. Sirve tanto para propósitos empresariales como para necesidades académicas y funciona tanto en equipos individuales como en servidores de empresas [2].

Con el avance de la tecnología de bases de datos de aplicaciones basadas en SQL se ha vuelto cada vez más asequible para el usuario normal. Esto se debe a la introducción de diversas soluciones de base de datos SQL de código abierto como MySQL [2].

Como se menciona, este lenguaje de consultas es el estándar al momento de manejar bases de datos, pero al igual que cualquier tecnología, tiene vulnerabilidades y una de ellas es la llamada inyección SQL.

En el año 1999, ya estaba ampliamente extendido el uso de php y el contenido dinámico a nivel de páginas web con MySQL, pero un año antes, el afamado hacker Jeff Forristal, fue el primero en describir,

en un artículo una novedosa técnica, en donde se podían ejecutar a través de conexiones ODBC se podían ejecutar consultas y comandos utilizando el lenguaje SQL [3].

Hoy día, dieciocho años después de su primera divulgación pública, la inyección SQL se encuentra, repetidamente en el puesto número uno de vulnerabilidades en el informe OWASP Top 10, que se publica cada tres años y que controla las peores amenazas a las que se enfrentan los sitios web en la actualidad [3].

La capacidad de tener acceso desde cualquier lugar y tiempo es lo que ha hecho de las aplicaciones web, un elemento indispensable en la vida de todo ser humano hoy en día. Sin embargo, esta misma capacidad representa, en muchos de los casos, una desventaja en la seguridad de los datos en información que contiene una aplicación [4].

Se menciona que uno de los ataques más frecuentes en la web son los de inyección SQL, debido a su capacidad para obtener e insertar información de las bases de datos [5].

En “On automated prepared statement generation to remove SQL injection vulnerabilities” T. Stephen. Menciona que desde el 2002 el 10% de los ataques se ha realizado mediante inyección SQL. Tan solo entre los años 2002 a 2007 este tipo de ataque fue de un 20% en las vulnerabilidades de las validaciones de entrada y un 10% de vulnerabilidades cibernéticas [6].

La capacidad de acceder a la web en cualquier lugar y en cualquier momento es una gran ventaja; sin embargo, a medida que la web se vuelve más popular, los ataques web van en aumento. La mayoría de los ataques web se dirigen a las vulnerabilidades de las aplicaciones web, que se han investigado y analizado en OWASP [7].

De acuerdo a lo anterior mencionado, diversos investigadores han estudiado este tipo de ataques utilizando diversos métodos para su detección y prevención.

2 PLANTEAMIENTO DEL PROBLEMA

A lo largo del tiempo, hemos visto que se han empleado bases de datos en aplicaciones web, esto es muy interesante, debido a que las aplicaciones web les permiten a los usuarios almacenar grandes cantidades de datos y administrar la información.

A lo mencionado anteriormente, es un gran apoyo a la administración de la información en las empresas de hoy en día, en la actualidad existen diversos ataques hacia estas aplicaciones debido a la información que manejan, pero el problema ocurre, cuando un usuario experto desea obtener esa información de la base de datos de la aplicación web, donde el usuario ingresa código SQL adicional al ya programado en la barra de direcciones o en algún campo de texto donde se realicen consultas a la base de datos, haciendo que el código SQL realice una consulta general de todos los datos contenidos en la base de datos.

En la actualidad muchas empresas dicen haber sufrido ataques de robo de información, entre ellos se encuentra la inyección SQL, la cual consiste en atacar a las bases de datos para realizar ciertas acciones que nos permite el lenguaje SQL tales como, Insertar registros falsos, borrar datos de la base de datos, consultar toda la información contenida en la base de datos o modificar ciertos registros.

Se sabe que la mayoría de este tipo de ataques es debido al o los desarrolladores de la aplicación web, por no haber previsto o tomado en cuenta este tipo de amenazas.

En la actualidad este tipo de ataques se han hecho más comunes en empresas donde se manejan grandes cantidades de datos, los atacantes aprovechan las vulnerabilidades de la aplicación para sacar algún provecho, muchos de estos ataques se enfocan en corromper la información contenida en la base de datos y hasta vender los datos que fueron obtenidos por medio del ataque de inyección SQL.

Este tipo de amenazas las cuales se aprovechan de las vulnerabilidades con las que cuentan algunas aplicaciones web, son problemas que se pudieron evitar desde las etapas de desarrollo de la aplicación, seguido de la etapa de testeo donde se identifican posibles vulnerabilidades y tomar medidas a tiempo.

Estos ataques en principio no eran tan frecuentes debido a que el lenguaje SQL no era un estándar y apenas se empezaban los desarrolladores a familiarizarse con el lenguaje, pero al paso de los años este

lenguaje se volvió un estándar para consultas a bases de datos por lo tanto todo desarrollador al hacer una aplicación web con base de datos se veía obligado a utilizar este lenguaje para poder interactuar con la base de datos.

En la actualidad, al ser un estándar el lenguaje SQL, muchos usuarios que ya tenían experiencia en el manejo de base de datos, comenzaron a pensar en una forma de poder alterar el funcionamiento original de la aplicación web y poder sacar provecho de esta vulnerabilidad. En la mayoría de los casos los desarrolladores pasan por alto este tipo de vulnerabilidades y no implementan métodos de validación de campos para evitar que se ingresen caracteres en común con el lenguaje SQL.

Como se mencionó este tipo de ataques siguen ocurriendo a día de hoy, a pesar de ser un ataque que se viene dando desde tiempo atrás.

3 JUSTIFICACIÓN DEL TEMA

La siguiente investigación se está realizando, debido a que en la actualidad se han reportado varios casos de ataques a aplicaciones web, mediante la técnica de inyección SQL y se considera importante el saber identificar vulnerabilidades en nuestra aplicación web, que pudieran permitir este tipo de ataques, por lo tanto, es importante el poder recomendar ciertas técnicas que en la actualidad nos brindan tanto los gestores de bases de datos y algunos lenguajes de programación.

Los ataques de inyección SQL se han utilizado desde hace tiempo, los cuales han evolucionado en la manera en la que atacan a las aplicaciones web.

Como se ha mencionado con anterioridad, los ataques de inyección SQL son de los más populares y usados, por lo cual, estos han hecho grandes daños (informáticos) a las personas y hasta empresas que han sido víctimas de ellos.

La mayoría de las veces se han investigado este tipo de ataques después de que ocurrieron, pero no nos hacen mención de cómo prevenir este tipo de ataques o que técnicas debemos utilizar, por lo tanto, debemos de contar con conocimiento acerca de este tipo de ataques y saber, más que nada, si existen herramientas que nos puedan ayudar.

En la actualidad se sabe que los lenguajes de programación del lado del servidor, cuentan con funciones que podemos implementar al momento de estar desarrollando la aplicación, ya que se sabe que la mayoría de estos ataques se pudieron haber prevenido desde la etapa de desarrollo de la aplicación, lo cual nos hace mención que la mayoría de los ataques son a causa de los desarrolladores al no tener en cuenta este tipo de ataques.

Por lo antes mencionado, en esta investigación se hará mención de técnicas que nos ayuden a prevenir este tipo de ataques, por lo tanto, las recomendaciones que se darán tendrán como fin, el tener un conocimiento sobre este tipo de ataque y saber cómo actuar o prevenir el mismo, de esta manera se intentará disminuir las vulnerabilidades en las aplicaciones web.

4 OBJETIVOS GENERAL Y ESPECÍFICOS

General: Identificar y recomendar ciertas técnicas que se puedan utilizar o implementar para prevenir o disminuir los ataques de inyección SQL en aplicaciones web.

Específicos: Investigar técnicas que se puedan implementar en la etapa de desarrollo de la aplicación web, así como también recomendar funciones que nos ofrece el gestor de base de datos, recomendar funciones que nos puedan brindar el lenguaje de programación para prevenir el ataque y recomendar técnicas para validar campos de texto evitando que se realicen este ataque.

5 VARIABLES INDEPENDIENTE Y DEPENDIENTE

Independiente: Validación de los datos de entrada del usuario.

Dependiente: Estabilidad de la base de datos.

6 METODOLOGÍA

Tipo de investigación: El tipo de investigación que se va a realizar, será de tipo cuantitativo, debido a que se pretende obtener datos que puedan ser de fácil medición.

Técnicas de investigación: Durante el desarrollo de esta investigación, la técnica de investigación ha sido de tipo documental, debido a que se ha investigado y complementado por medio de la revisión bibliográfica de documentos como lo son: tesis, tesinas, artículos y libros acerca del tema.

Objetos de recolección de datos: En cuanto a los objetos de recolección de datos se utilizará la encuesta, ya que a través de las preguntas formuladas se obtendrá específicamente los datos necesarios, además que se tiene un mayor control de la información recopilada de quienes participaran en la encuesta.

La encuesta se pretende aplicar a los alumnos de la Facultad de Informática Mazatlán de la Universidad Autónoma de Sinaloa, donde por medio de los resultados obtenidos mediante la encuesta, sabremos el nivel de conocimiento que tienen acerca de este ataque y si conocen sobre algunas técnicas que se puedan implementar.

7 HIPÓTESIS

La hipótesis es la siguiente: sí se validan de manera correcta los campos de texto en los formularios de la aplicación web, se disminuirán las intrusiones a la base de datos significativamente.

8 CONCLUSIONES

Se ha llegado a la conclusión de que los ataques de inyección SQL, son causados en mayor parte por falta de control de los datos que se ingresan en los campos de texto de los formularios en aplicaciones web, logrando insertar código malicioso mediante el uso del lenguaje SQL, alterando de esa manera la estabilidad de la base de datos que recibe el ataque, por tal motivo el validar correctamente los campos de texto logrará disminuir de manera considerable los ataques de inyección SQL.

En la actualidad se deben implementar métodos al momento de desarrollar la aplicación web, los cuales nos brindan ciertos frameworks de algunos lenguajes de programación del lado del servidor.

Lo anterior mencionado es algo que siempre se debe implementar en sistemas o aplicaciones web, ya que lo más importante es proteger y mantener un control sobre la información que se maneja a diario y que es gestionada en bases de datos, de esa manera evitando ser víctima de ataque de inyección SQL.

REFERENCIAS

[1] Carlos Eduardo Plascencia Prado, *¿Qué es y por qué aprender SQL?,* recuperado de: https://devcode.la/blog/que-es-sql/.

[2] Universidad Internacional de Valencia, *Lenguaje SQL, historia y conceptos básicos*, 2018, recuperado de: https://www.universidadviu.com/lenguaje-sql-historia-conceptos-basicos/

[3] Jonathan Préstamo Rodríguez, *Así fue descrita la primera inyección SQL de la historia*, 2016, recuperado de: https://www.teknoplof.com/2016/12/01/asi-fue-descrita-la-primera-inyeccion-sql-la-historia/

[4] Meléndez, S. Diseño e implementación de un algoritmo para detección de inyección sql en sitios web para dispositivos móviles [Tesis Maestría]. México, D.F: Instituto Politécnico Nacional; 2014. 116 p.

[5] I. Lee, S. Jeong, S. Yeo y J. Moon, “A novel method for SQL injection attack detection based on removing SQL Query attribute values”, ELSEVIER, vol. 55, no. 1, pp. 58-68, 2012.

[6] S. Thomas, L. Williams y T. Xie,” On automated prepared statement generation to remove SQL injection vulnerabilities”, ELSEVIER, vol. 51, no. 3, pp. 589 – 598, 2009.

[7] R. Latha y E. Ramaraj, “SQL Injection Detection Based On Replacing The SQL Query Parameter Values”, ijecs, vol. 4, no. 8, pp. 13786 – 13790, 2015.