COMPILACIÓN DE TIPOS DE ATAQUE A BASES DE DATOS: INYECCIONES SQL

Romero Salas Denisse Andrea[1]

[1]*Facultad de Informática Mazatlán, Universidad Autónoma de Sinaloa (MÉXICO)*

Resumen

Este proyecto da a conocer uno de los ataques a las bases de datos más conocidos, qué es la inyección SQL y algunas formas de prevenirla e identificarla. Con la ayuda de una tabla en la que se clasifican algunos de los tipos de ataque más básicos, así como también se definen conceptos básicos como las bases de datos y se explica el origen de este problema de inyección, para que de esta manera la información presentada pueda ser comprendida por un público no especializado.

Palabras clave: Base de Datos, Inyección SQL.

1 INTRODUCCIÓN

Desde hace mucho tiempo, los seres humanos han tenido la necesidad de guardar información, por lo que, con el paso del tiempo se han creado herramientas para la gestión de esta. Gracias a que la tecnología ha tenido un avance constante, es que estas herramientas han evolucionado a lo que hoy conocemos como una base de datos.

Una base de datos esta definida como "*un sistema formado por un conjunto de datos almacenados en un soporte no volátil lógicamente relacionados entre sí de manera que se controla el almacenamiento de datos redundantes*"[1]. Es decir, las bases de datos son almacenamientos de datos interrelacionados que interactúan con las aplicaciones, que les dan significado a los datos almacenados, pero que, a la vez es independiente de estas.

Debido a que información importante es almacenada en las bases de datos, es que ha surgido interés en personas no éticas por extraer el contenido de estas, por lo que se han creado diferentes ataques para lograr este objetivo. Uno de los ataques más conocidos es la inyección de código SQL, que está definido como "*un método de infiltración de código intruso que se vale de una vulnerabilidad informática presente en una aplicación en el nivel de validación de las entradas para realizar consultas a una base de datos*"[2]. Es decir, este actúa introduciendo instrucciones de código SQL en campos de texto donde se validan datos que proporciona el usuario para tener acceso a la información contenida en la base de datos mediante la aplicación que es la intermediaria entre los usuarios y la base de datos.

La primera vez que se tuvo conocimiento sobre este problema, según el sitio *securitybydefaul*[3], fue en el año 1998, cuando en una e-zine Phrack (una revista distribuida a través de correo electrónico a sus suscriptores), un usuario bajo el pseudónimo de *Rain Forest Puppy*, publicó que había encontrado una vulnerabilidad en un servicio web que interactuaba con una base de datos MS SQL Server 6.5, en la cual, se percató de que era posible ejecutar instrucciones SQL encadenadas.

Desde el descubrimiento de los ataques de inyección, se ha estado combatiendo a este tipo de ataques para evitar el robo de información, mediante actualizaciones o medidas de seguridad para corregir o prevenir problemas que puedan resultar en la infiltración de personas externas para el robo o la modificación del contenido de las bases de datos.

Pero aun con el avance de la tecnología, las inyecciones SQL siguen siendo uno de los principales ataques, pues, a pesar del constante cuidado que se le da a la seguridad de las bases de datos, siguen existiendo diversas maneras para infiltrarse en ellas y poder modificar su contenido, debido a que, este ataque también ha estado cambiando conforme avanza el tiempo. Es por esto que, también han surgido diferentes maneras de ejecutar una inyección.

Según el autor del documento "*Inyección de SQL, tipos de ataques y prevención en asp.net*"[2], estos ataques de inyección son de los más comunes:

- *Introducir una instrucción siempre verdadera:*
  Este tipo de ataque inyecta SQL a la instrucción de una consulta adicional para que al ser ejecutado siempre sea cierto.

- *Lógica incorrecta en las consultas SQL:*
  Cuando una consulta es rechazada por el motor de base de datos, un mensaje de error es devuelto incluyendo la información para que su depuración sea fácil o útil.
- *Unión de consultas:*
  Mediante esta técnica los atacantes unen, una o varias consultas por medio de la palabra "UNION", con la cual se puede obtener datos sobre otras tablas de la aplicación.
- *Ejecutando consultas adicionales:*
  Con este tipo de vulnerabilidad, el atacante explota la base de datos por el delimitados de consulta, tal como ";", al anexar una consulta adicional a la consulta original.

Después de identificar la forma en que actúan las inyecciones de código, se pudo dar con técnicas para resolver este problema, por lo tanto, estas se han trasmitido de diferentes formas para tratar de acabar con los ataques a las bases de datos.

Algunas páginas como *php.net*[4], dan las siguientes precauciones para la prevención de las inyecciones SQL más básicas:

- Nunca se conecte como super-usuario o como propietario de la base de datos. Siempre utilice usuarios personalizados con privilegios muy limitados.
- Emplee sentencias preparadas con variables vinculadas.
- Compruebe si la entrada proporcionada tiene el tipo de datos previsto.

Según la página *Digital Guide*[5], estos son otros métodos de prevención:

- Supervisar las modificaciones automáticas en las aplicaciones
- Proporcionar una protección completa al servidor.
- Blindar la base de datos y utilizar códigos más seguros.

Para *Tovar*, en su proyecto "*Inyección de SQL, tipos de ataques y prevención en asp.net*": *"Relacionar la base de datos",* es uno de los métodos que se pueden utilizar para la prevención de estos ataques, ya que, al relacionar todas las tablas de la base de datos, se podría evitar el perder información por la sentencia DELETE, puesto que, se necesita confirmación para eliminar las relaciones entre las tablas.

Ya vistos los problemas técnicos, se debe de tener una idea de las consecuencias que se generan al ser víctima de un ataque de inyección a la base de datos.

Las consecuencias causadas por este ataque, según *Supra Networks*[6], son las siguientes:

- ***Omisión de autenticación:*** *Si la forma de autenticación de la aplicación es vulnerable a la inyección de SQL, el usuario puede iniciar sesión en la aplicación sin proporcionar las credenciales adecuadas.*
- ***Obtener acceso a datos no autorizados:*** *A través de la inyección de SQL, un usuario puede obtener acceso a datos que solo personal autorizado debería.*
- ***Manipulación de datos no autorizada****: La inyección de SQL también puede permitir que un usuario de la aplicación inserte, modifique o elimine datos que no esté autorizado. Esto hace que la integridad de los datos se vea comprometida.*
- ***Obtenga privilegios administrativos:*** *la inyección de SQL podría permitir a un atacante o un usuario malintencionado obtener privilegios administrativos en la base de datos o en el servidor de la base de datos y, en última instancia, realizar acciones como cerrar la base de datos. Esto afecta su disponibilidad y, en consecuencia, la falta de disponibilidad de la aplicación.*

Muchos no tienen conocimiento de lo perjudicial que puede llegar a ser un problema de inyección, por este motivo, la presente investigación busca analizar lo que es la Inyección SQL y explicar maneras de solucionar o evitar este tipo de infiltración para mantener una base de datos segura de personas externas a ella.

2 METODOLOGÍA

El objetivo de esta investigación es el análisis de las inyecciones SQL para hacer ver a las personas lo que puede suceder en el caso de no cuidar adecuadamente una base de datos, por lo que, se inició buscando la definición de inyección SQL. Lo siguiente fue buscar las diferentes formas en las que se puede presentar una inyección, para tener una idea más clara sobre cómo se ejecuta el ataque, aunque de las maneras más básicas. Para esto, se tuvieron que plantear las bases de lo que es, para conocer y entender las consecuencias que ocasionan estos ataques en las bases de datos, por lo que, se

investigaron los conceptos de bases de datos y códigos de SQL, que se utilizan para la creación y modificación de las bases de datos.

Una vez que se tienen los conceptos básicos, se procedió a buscar formas para prevenir que este problema suceda, después de identificar algunas formas para prevenir las inyecciones de código SQL, se buscaron las consecuencias que ocasionan en la base de datos, estas instrucciones de código antes mencionadas.

3 RESULTADOS

A continuación, se presentará una tabla en la que se explican algunos de los problemas más simples al ejecutar la inyección de código SQL ordenados por categoría, para tener una idea de cómo se ejecutan estos, en las aplicaciones que están conectadas a las bases de datos.

Tabla 1. Categorías de los ataques más básicos de inyección SQL.

Tipos de ataque		Descripción	Ejemplos de errores
Inyección por error		Este tipo de error, utiliza los errores que manda la aplicación y va mostrando los errores que da la base de datos cuando se va introduciendo código de inyección SQL. Con este error es muy fácil conseguir cualquier cosa de la base de datos (estructura, tablas, nombres de campos e incluso datos).	•Lógica incorrecta en las consultas SQL: ***1)*** *Url con parámetros:* *http://www.ejemplo.com/usuarios.aspx?id=888* ***2)*** *Url con inyección de delimitador " ' ":* *http://www.ejemplo.com/usuarios.aspx?id=888'* ***3)*** *Mensaje de error por la consulta:* *SELECT * FROM Usuarios WHERE id=8864\';* •Introducir una instrucción siempre verdadera: *' or 1=1 –*
Inyección por unión		Consiste en añadir una consulta que empiece con UNION, revelando información sensible.	•Ejecutar consultas adicionales: *SELECT FROM Usuarios WHERE Usuarios = 'admin' AND Contrasena = 'xc#12Ns'; DROP TABLE Usuarios;*
Inyección a ciegas	Basada en contenido (o booleana)	Consiste en introducir expresiones booleanas para verificar si la aplicación es vulnerable a inyecciones de código SQL. Si hay cambios después de la inyección, significa que la aplicación si es vulnerable[7].	•Cambios en la aplicación con una de estas consultas: *www.sitioweb.com/index.php?id=3 AND 1=1 // verdadero* *o* *www.sitioweb.com/index.php?id=3 AND 1=0 // falso*
	Basada en tiempo	Se basa en pausar la base de datos por un tiempo especificado, para que posteriormente devuelva los resultados, indicando si la consulta triunfo o no[7].	•Consultas para pausar la base de datos: *www.sitioweb.com/index.php?id=3 ' AND SLEEP(10)=' //MYSQL* *www.sitioweb.com/index.php?id=3 WAITFOR DELAY '0:0:5' //MSSQL*

4 CONCLUSIONES

En la presente investigación, se presentaron conceptos básicos para la mejor compresión del tema de este artículo, que es la inyección SQL, para así, tener mayor entendimiento del funcionamiento de este ataque y las consecuencias que puede presentar en las bases de datos. Como resultado, se obtuvo una tabla en la que se clasificaron algunos de los ataques que se realizan más comúnmente, mostrando algunos ejemplos con el fin de poder identificar con mayor facilidad el tipo de ataque y dar con una posible solución de una manera más rápida.

Con este proyecto, se llegó a la conclusión que, a pesar de que este ataque es uno de los más comunes, aún sigue habiendo ignorancia sobre él, debido a que no todas las personas sienten curiosidad por lo que sucede con sus datos después de ser capturados por las aplicaciones. Por esto, uno de los objetivos presentados anteriormente, fue presentar las consecuencias que tiene en las bases de datos, sufrir algún tipo de inyección. De esta forma, se puede hacer ver a las personas lo que sucede con la información.

5 REFERENCIAS

[1] *Pérez Sandoval, Jessica, Las bases de datos, su seguridad y auditoria. El caso de MySQL. Leganés, 2011.*

[2] *Tovar Valencia, Orlando, Inyección de SQL, tipos de ataques y prevención en asp.net. Bogotá, S.F.*

[3] *Guasch, José, securitybydefault, 6 de noviembre de 2019. Recuperado de http://www.securitybydefault.com/2013/11/quien-descubrio-las-inyecciones-de.html*

[4] *Php.net, consultado el 6 de noviembre de 2019. Recuperado de https://www.php.net/manual/es/security.database.sql-injection.php*

[5] *Digital Guide, consultado el 6 de noviembre de 2019. Recuperado de https://www.ionos.mx/digitalguide/servidores/seguridad/inyeccion-sql-principios-y-precauciones/*

[6] *Supra Networks, consultado el 6 de noviembre de 2019. Recuperado de https://www.supra.com.pe/blog/ataques-de-inyeccion-sql-que-son-y-como-protegerse/*

[7] *Backtrackacademy, consultado el 7 de noviembre de 2019. Recuperado de https://backtrackacademy.com/articulo/inyeccion-sql-definicion-y-ejemplos*

[8] *Acens, Bases de datos y sus vulnerabilidades más comunes, consultado el 5 de noviembre de 2019. Recuperado de https://www.acens.com/comunicacion/white-papers/backtrackacademy.com/articulo/inyeccion-sql-definicion-y-ejemplos*

[9] *Marcelo Racciatti, Hernán, Técnicas de SQL Injection: Un Repaso. N.D., 2002. [versión 1.5] recuperado de https://www.redeszone.net/content/uploads/Tecnicas-de-SQL-Injection.pdf.*

[10] *Minero Guardado, Jesús, Identificación y Clasificación de las Mejores Prácticas para Evitar la Inyección SQL en Aplicaciones Desarrolladas en PHP y PostgreSQL. Zacatecas, 2011.*

[11] *Chicaiza, Giovanny, Ponce, Luis, Velázquez Campos, Gabriela, Inyección de SQL, caso de estudio OWASP. Sangolquí, S.F.*

[12] *Anónimo, Ataques de Inyección SQL. Qué son y Cómo protegerse. Bilbao, S.F.*

[13] *Losada Regos, Diego, Seguridad en aplicaciones web. S.C., 2015.*

[14] *Gómez, Ivan Camilo, Diseño de metodología para verificar la seguridad en aplicaciones web contra inyecciones SQL. Bogotá, 2011.*

[15] *López Soto, Manuel Alberto, Peraza Garzón, Juan Francisco, Protección ante ataques de inyección SQL en aplicaciones web. Mazatlán, S.F.*

[16] *Peris Cortés, Jorge, Introducción a ataques de tipo inyección: Inyección SQL. Valencia, S.F.*

[17] *Meléndez César, Sergio, Diseño e implementación de un algoritmo para detección de inyección SQL en sitios web para dispositivos móviles. México, 2014.*

[18] *Coronado, Steven, Desarrollo de una guía metodológica basada en análisis SQL Injection y formas de protección a las bases de datos. Quito, 2014.*

[19] *Domínguez Chávez, Jorge, Principios básicos de seguridad en bases de datos. Aragua, S.F.*

[20] *Alonso Cebrián, José, Guzmán Sacristán, Antonio, Laguna Durán, Pedro, Bailón, Alejandro Martín, Ataques a BB. DD., SQL Injection. Cataluña, S.F.*