Revisión Bibliográfica: Comparación de métodos contra ataques SQL para bases de datos SQL y NoSQL

Mirtha Jiménez, Jessica Urquizo, Byron Bonifaz
ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO SEDE ORELLANA
mirtha.jimenez@espoch.edu.ec, jessica.urquizo@espoch.edu.ec, byron.bonifaz@espoch.edu.ec

Resumen - El internet cada día crece a pasos agigantados y con ello la vulnerabilidad de las aplicaciones web se convierte en una preocupación importante. Es decir, la inyección SQL o más conocido como ataque de inserción SQL es la técnica de inyección de código, la misma que es la responsable de la fuga de seguridad, esta se produce en la base de datos de aplicaciones o páginas web que cuentan con un contenido dinámico. Este documento realizo un análisis de las diversas formas de inyección de código malicioso a bases de datos relacionales y no relacionales con el objetivo de sugerir formas para prevenir dichos ataques para el cual la investigación, análisis y resultados se obtuvieron de 39 artículos todo esto con el fin de recomendar el mejor método para prevenir una inyección de código malicioso.

Palabras Claves – Base de datos, Inyección SQL, Métodos, seguridad, pirateria, vulnevaridad.

I. INTRODUCCIÓN

Debido a los constantes cambios tecnológicos y la importancia que ha tomado la información digital, ha hecho que personas maliciosas quieran sacar beneficio de esto. A causa de esto la seguridad en las aplicaciones web cada día es más desafiante y preocupante ya que la inyección SQL es uno de los principales métodos de la piratería informática.

La explotación de estos ataques SQL aún es menor porque esta vulnerabilidad no solo es relevante en la web, sino que también puede ocurrir en aplicaciones de escritorio que usan software de servidor.

Por esta razon, las inyecciones SQL afectan directamente a las bases de datos relacionales y no relacionales con sus diferentes formas de ataques. La capacidad de detectar estas vulnerabilidades depende de la complejidad de la aplicación asociada.

La mayoría de las veces, las herramientas como antivirus o firewalls no nos alertan sobre fallas de seguridad existentes. [1]

Por ello hemos visto la necesidad de analizar, clasificar, e inferir cual es el mejor método para defenderse ante un ataque (inyección SQL), mediante un estudio de comparación de diferentes fuentes de información. A través de esto queremos recomendar posibles formas o métodos para prevenir o mitigar un ataque de inyección SQL.

II. MARCO TEORICO

A. ¿Qué es la inyección SQL?

La inyección SQL es cuando se rompe en la base de datos o incluso
se bloquea la base de datos causando problemas especialmente en el servicio de los datos conectados.

Scnce JavaScript es ampliamente utilizado, por piratas informáticos y los desarrolladores que practican la implementación, para los procesos de inyección SQL y protección. [2]

B. ¿Qué es seguridad de la información?

La seguridad de los sistemas informáticos es un campo en constante evolución. El objetivo final de la seguridad es permitir que una organización logre con todos sus objetivos negociables o misión, mediante la implementación de sistemas que presten especial atención a los riesgos de las TIC. [3]

C. Inyección de código malicioso a bases de datos SQL

Las aplicaciones web son propensas a ataques de seguridad, entre esos se encuentra el ataque a su base de datos mediante inyección de código malicioso o inyección SQL.

Para detectar fallas de seguridad de un sitio es recomendable hacer pruebas a los puntos vulnerables del mismo. Los resultados nos pueden ayudar a detectar los ataques de inyección más rápidamente y poder actuar ante ellos. [4]

Se recomienda usar sentencias preparadas, declarar el tipo de entrada para formularios, implementación de Chroot, Apparmor, SELinux. [5]

Por otro lado, usar una red neuronal para poder detectar

automáticamente las sentencias NoSQL mediante aprendizaje automático también ayuda a prevenir los ataques. [6]

D. Tipos de inyección SQL

a) Mediante cadenas mal filtradas.

La inyección de SQL basada en cadenas filtradas incorrectamente es causada por la entrada del usuario que no se vacía. Lo que genera que el usuario pueda ingresar una variable que se puede pasar como en una instrucción SQL, lo que hace que el usuario final ingrese a la base de datos.

b) Incorrecta verificación de tipo

Se produce una verificación de tipo incorrecta si la entrada no está marcada con una restricción de tipo. Un ejemplo de esto es un campo de ID numérico, pero no se implementó ningún filtro para verificar que la entrada del usuario sea numérica, por lo que siempre debe usar isnumeric() si se asume explícitamente que el tipo de campo es numérico.

c) Evasión de firma

Es cuando muchas inyecciones SQL son bloqueadas mediante un sistema de detección y prevención de intrusiones que utiliza reglas de detección de firmas. Los programas comunes de detección de inyección SQL son las contramedidas Apache y Snort.

d) Blind SQL injection.

Estas inyecciones se denominan inyecciones SQL ciegas. Entre ellos, se dividen en ciegos parciales y ciegos completos. Una inyección parcialmente ciega puede tener cambios menores en la página resultante, por ejemplo, las inyecciones incorrectas pueden redirigir al atacante a la página de inicio y devolver una página en blanco si tiene éxito. [1]

E. Ataques NoSQL en base de datos no relacionales

Las bases de datos no relacionales cada vez toman más participación debido a que la cantidad de información que se maneja cada vez es mayor y, en muchas ocasiones la información queda distribuida en varios servidores, esto hace que las bases de datos relacionales no puedan cumplir con su propósito.

a) Tautologías Los atacantes hacen uso de sentencias condicionales para inyectar código NoSQL.

b) Consultas gremiales Se usa la técnica de consultas unión para saltarse las páginas de autentificación.

c) Inyecciones de JavaScript Uso de sentencias mediante JavaScript.

Consultas superpuestas Uso de las suposiciones en la interpretación de los caracteres especiales de las secuencias de escape como los caracteres de terminación, etc.

Violación de origen cruzado Los atacantes usan las API REST de HTTP, las cuales son usadas por las bases de datos NoSQL. [7]

III. MATERIALES Y MÉTODOS

Este artículo es una revisión bibliográfica, mediante un método cuantitativo donde se recopilo la información más relevante mediante búsquedas en artículos científicos, limitándonos en que la mayor parte de los documentos correspondan a un cuartil y que estén entre 2006 y 2022 sobre ataques de inyección SQL. Las revisiones bibliográficas son herramientas, que nos permite reunir información de muchas fuentes y hacerlas una sola información, forma parte de lo primordial antes de realizar una investigación ya que, con su ayuda, facilita la justificación del documento, y atreves de su relevancia ayuda a entender nuestras propias metas. Usamos un repositorio de artículos “Recursos Científicos” y una hoja de cálculo para organizar, almacenar y clasificar los artículos investigados (alrededor de 39), después de ello extrajimos la información más relevante o requerida y, por último, los tabulamos para que de esa manera esta revisión bibliográfica sea útil. Con el objetivo de evaluar los diferentes artículos y categorizarlos de manera que el lector pueda decidir cuál implementar y mejorarla para futuras investigaciones.

A. Preguntas de investigación

La inyección SQL y NoSQL ha sido el método de ataque número uno durante cuatro años seguidos, según lo mencionado por SQLiDDS: SQL Injection Detection Using Query Transformation and Document Similarity 2015. Esta es una vulnerabilidad que los hackers comunes aún explotan. [8]

La información publicada anteriormente en esta área cumple con algunos criterios, lo que brinda una comprensión clara de las vulnerabilidades, sus posibles soluciones/métodos, los dominios y las formas de trabajar. Por lo tanto, utilizamos las siguientes preguntas de investigación:

P1: ¿Qué áreas de ataques se han trabajado más?

P2: ¿En qué soluciones y métodos se centran estos artículos?

P3: ¿Cuáles son los tipos de Ataques de inyecciones SQL que más se han estudiado?

P4: ¿Qué tipo de base de datos son las más afectadas? ¿Por qué?

Para responder a P1, analizamos todos los artículos estudiados. Para revisar las soluciones y los métodos propuestos por los artículos, los organizamos y analizamos de una manera exclusiva con el objetivo que se proporcione más adelante. El procedimiento nos condujo a encontrar la respuesta a P2. Para P3, estudiamos los documentos donde profundizan más de cerca los tipos de ataques. Para P4, tratamos de encontrar los tipos de inyecciones SQL más estudiados y, como resultado, categorizamos los contenidos de los artículos sobre esta base.

B. Proceso de búsqueda

También se realizó una búsqueda manual para determinar si se encontraron archivos relevantes. Se buscaron artículos sobre inyección SQL en la mayoría de los repositorios de recursos científicos y en algunas bases de datos de Scopus. Se descargaron alrededor de 39 artículos que coincidían con sus criterios.

- ✓ Utilizamos palabras claves para buscar en la base de datos:
- ✓ Inyección SQL.
- ✓ Tipos de inyección SQL.
- ✓ Clases de inyección SQL.
- ✓ Métodos para inyecciones SQL.
- ✓ Inyecciones SQL en Base de datos.
- ✓ Soluciones para ataques SQL.

C. Proceso de búsqueda

En la revisión bibliográfica, nos enfocamos más en los artículos de investigación que cumplen con nuestros criterios esperados que en aquellos que cubren un tema diferente.

Los documentos que:

a) Se centraron en inyecciones SQL para base de datos relacionales.
b) Abordaron metodologías para la prevención de las inyecciones.
c) Proporcionaron herramientas para los problemas de inyección.

En general, trataron sombre:

a) Inyección SQL como un ataque ante la seguridad de aplicaciones web.
b) Recomendaciones al momento de sufrir un ataque.

D. Evaluación de Calidad

Utilizamos criterios de bases de datos de nuestros resúmenes de revisión de artículos de investigación para responder tres preguntas de evaluación de calidad (QA) en las que confiamos para los criterios utilizados:

QA1: ¿Usaron más de tres formas de inyección SQL para detectar vulnerabilidades del sitio atacado?

QA2: ¿Los métodos o herramientas abarcaron los dos tipos de bases de datos (relacionales-no relacionales)?

QA3: ¿Los métodos empleados redujeron el ataque SQL?

La puntuación de las preguntas anteriores se hizo de la siguiente manera:

QA1: Y (sí) Si implementaron de tres o más métodos para inyectar código SQL.

R (regularmente) Si usaron de dos a tres métodos para inyectar código SQL.

N (no) Si solo usaron una forma de inyectar código SQL.

QA2: Y (sí) Si los métodos o herramientas dan solución para base de datos relacionales y no relacionales.

R(regularmente) si ciertos métodos o herramientas funcionaban en ciertas bases de datos.

N (No) sí los métodos o herramientas son efectivos solo para las bases de datos implementadas.

QA3: Y (sí) si la herramienta o método usado evito la inyección SQL.

R (regularmente) si solo detectaron ciertos tipos de inyecciones.

N (No) si no previno ni detecto ningún tipo de ataque.

Los resultados se muestran en la TABLA II, donde Y = 1 R= 0,5 N=0

QA = Evaluación de calidad.

N° = Numero

TABLA I

REVISIÓN DE CALIDAD DE LOS ARTÍCULOS DE INVESTIGACIÓN.

		QA			
N°	Título de las revistas	1	2	3	T
AR 1	SQL injection attack detection: Profiling of web application parameter using the sequence pairwise alignment	Y	Y	Y	3
AR 2	SDriver: Location-specific signatures prevent SQL injection attacks	R	N	R	1
AR 3	SQL Injection Defense Mechanisms for IIS plus ASP plus MSSQL Web Applications	Y	Y	Y	3
AR 4	Combinatorial methods for dynamic gray-box SQL injection testing.	Y	R	R	2
AR 5	Idea: Using System Level Testing for Revealing SQL Injection-Related Error Message Information Leaks	R	N	Y	1.5
AR 6	A novel method for SQL injection attack detection based on removing SQL query attribute values.	Y	R	Y	2,5
AR 7	Web Anomaly Misuse Intrusion Detection Framework for SQL Injection Detection	Y	N	Y	2
AR 8	Object oriented approach to SQL injection preventer	R	Y	Y	2.5
AR 9	Integrated approach to prevent SQL injection attack and reflected cross site scripting attack	N	R	Y	1,5
AR10	SQL injection attacks with the AMPA suite	Y	R	Y	2.5
AR11	Lethality of SQL injection against current and future internet technologies.	R	N	Y	1,5
AR12	MAC based solution for SQL injection.	Y	R	Y	2,5
AR13	SQLPIL: SQL injection prevention by input labeling.	R	R	R	1,5
AR14	SQL shield: Preventing SQL Injection Attacks by Modifying User Input Data	R	R	R	1,5
AR15	SQLiDDS: SQL Injection Detection Using Query Transformation and Document Similarity	Y	N	Y	2
AR16	A Top Web Security Vulnerability SQL Injection attack – Survey.	Y	Y	Y	3
AR17	MongoDB NoSQL Injection Analysis and Detection	N	N	Y	1
AR18	Analysis and Mitigation of NoSQL Injections	Y	N	Y	2
AR19	Neutralizing SQL Injection Attack Using Server-Side Code Modification in Web Applications.	Y	R	Y	2.5
AR20	Towards Analyzing MongoDB NoSQL Security and Designing Injection Defense Solution.	Y	N	Y	2
AR21	A Mutation Approach of Detecting SQL Injection Vulnerabilities.	R	R	R	1,5
AR22	Research on the Technology of Detecting the SQL Injection Attack and Non-Intrusive Prevention in WEB System.	Y	Y	Y	3

AR23	SQL Injection Attack classification through the feature extraction of SQL query strings using a Gap-Weighted String Subsequence Kerne.	Y	R	Y	2,5
AR24	Basic NoSQL injection analysis and detection on mongo dB.	Y	Y	Y	3
AR25	SQL Injection Attack Principles and Preventive Techniques for PHP Site.	Y	R	Y	2,5
AR26	Securing SQL Injection flaw	Y	N	Y	2
AR27	A SQL Injection Detection Method Based on Adaptive Deep Fores	Y	R	Y	2.5
AR28	LsSQLIDP : Literature survey on SQL injection detection and prevention techniques	Y	R	R	2
AR29	Defeating SQL injection attack in authentication security: an experimental study.	Y	N	R	1,5
AR30	CODDLE: Code-Injection Detection with Deep Learning	Y	N	Y	2
AR31	ART4SQLi: The ART of SQL Injection Vulnerability Discovery	R	N	Y	1.5
AR32	Automatic Detection of NoSQL Injection Using Supervised Learning	Y	N	Y	2
AR33	SQL INJECTION - PREVENTION AND DEFENSE	R	R	Y	2
AR34	Detection of SQL injection based on artificial neural network	Y	Y	Y	3
AR35	Detection of SQL Injection Vulnerability in Embedded SQL	Y	N	Y	2
AR36	DIAVA: A Traffic-Based Framework for Detection of SQL Injection Attacks and Vulnerability Analysis of Leaked Data	N	R	Y	1,5
AR37	A systematic review of detection and prevention techniques of SQL injection attacks	Y	Y	Y	3
AR38	The Importance of Developing Preventive Techniques for SQL Injection Attacks	N	N	R	0,5
AR39	Detecting SQL Injection Vulnerabilities Using Nature-inspired Algorithms	Y	Y	Y	3

A. Descripción de la tabla

Como se muestra en la Tabla I, proporciona los resultados basados en la investigación, más de la mitad de los artículos examinados y analizados tuvieron una puntuación superior a 2, y entre los artículos examinados, también hubo los siguientes resultados: 11 artículos de 0 a 1,5. Hay 28 artículos entre 2 y 3.

IV. MATERIALES Y MÉTODOS

Los resultados compilados e investigados en la sección anterior se analizan para dar respuesta a las preguntas. Para ello se muestra el número de trabajos publicados desde 2006 como se muestran en la Fig. 1, dando como respuesta que en el año 2019 fue en el que más nos reflejó artículos basados al tema.

Fig. 1. Artículos clasificados por año.

Centrándonos en base a la respuesta de QA1, la Fig. 3, muestra los siguientes resultados en la cual el 67% muestra que Y (sí) Si implementaron de tres o más métodos para inyectar código SQL. EL 23 % responde a la occión de R (regularmente) Si usaron de dos a tres métodos para inyectar código SQL. Y el 10% restante corresponde a la occion de respuesta de N (no) Si solo usaron una forma de inyectar código SQL.

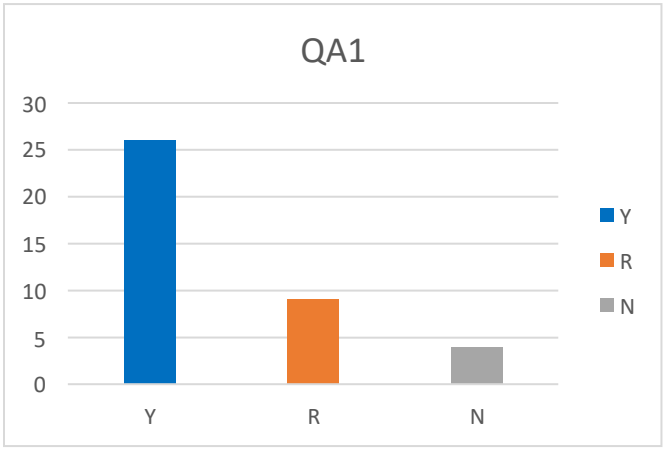

Fig. 2. Resultados de las respuestas dentro de la pregunta uno.

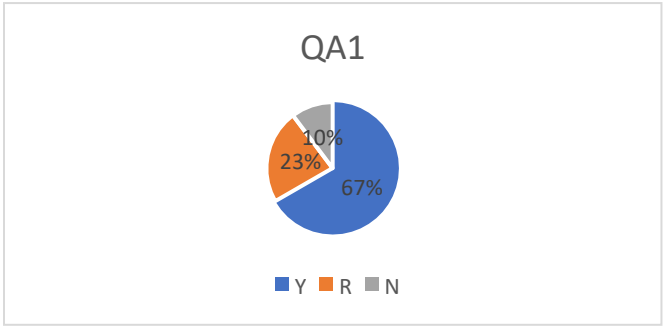

Fig. 3. Porcentaje de las respuestas de la pregunta uno.

Analizando cada uno de los artículos investigados con referencia en la respuesta de la QA2: ¿Los métodos o herramientas abarcaron los dos tipos de bases de datos (relacionales-no relacionales)? Se muestra en la Fig.5, dejando un resultado del 39% correspondiente a Y (sí) Si los métodos o herramientas dan solución para base de datos relacionales y no relacionales. El 38% respentivo a la occion de respuesta R(regularmente) si ciertos métodos o herramientas funcionaban en ciertas bases de datos. Y el 23% restante a N (No) sí los métodos o herramientas son efectivos solo para las bases de datos implementadas.

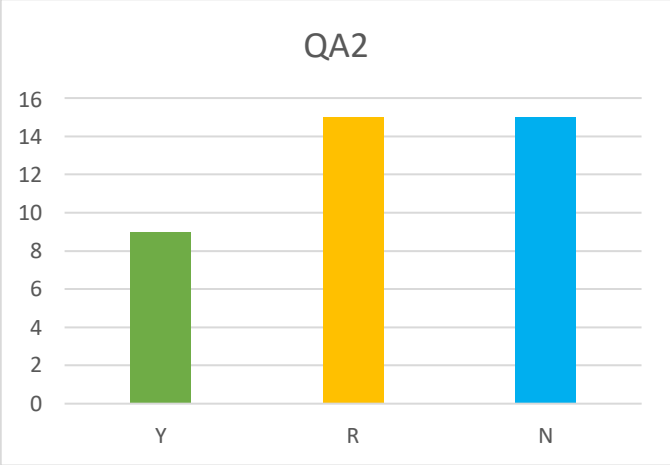

Fig. 4. Resultados de las respuestas dentro de la pregunta dos.

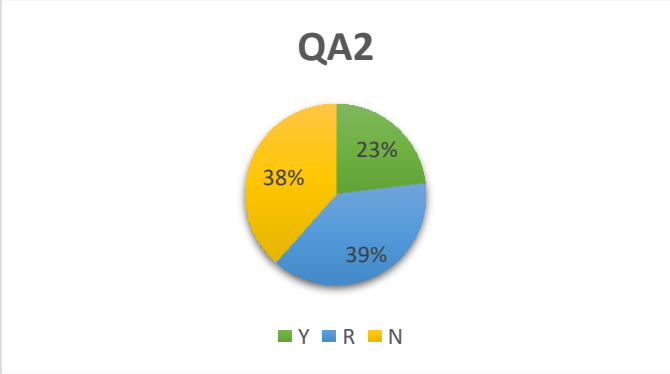

Fig. 5. Porcentaje de las respuestas de la pregunta dos.

La Fig.7, muestra los resultados obtenidos en el cual nos detallan que el 79% equivale a la respuesta; Y (sí) si la herramienta o método usado evito la inyección SQL. El 21% relativo a la occion R (regularmente) si solo detectaron ciertos tipos de inyecciones. Y el 0% corresponde al N (No) si no previno ni detecto ningún tipo de ataque.

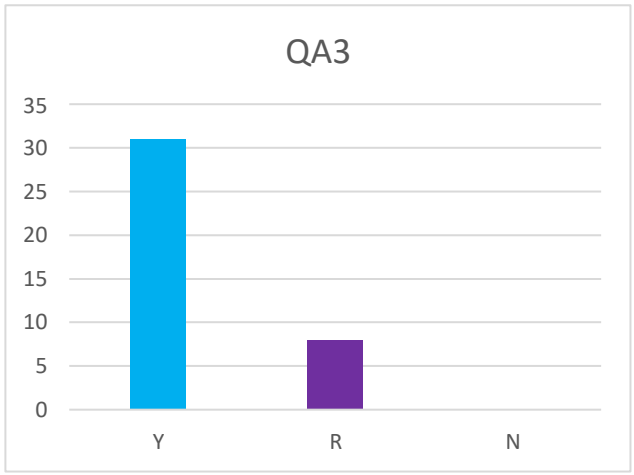

Fig. 6. Resultados de las respuestas dentro de la pregunta tres.

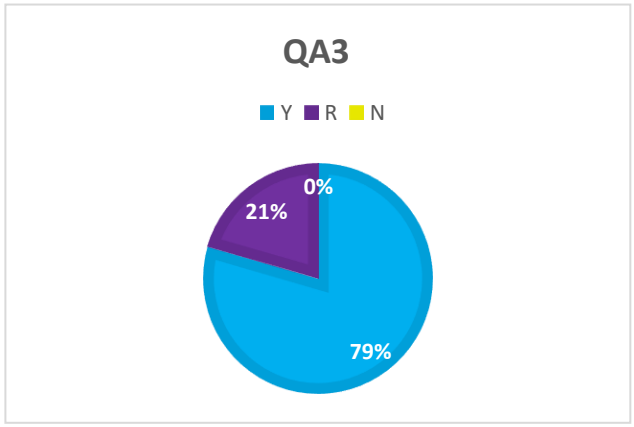

Fig. 7. Porcentaje de las respuestas de la pregunta tres.

V. DISCUSIÓN

Los métodos de inyecciones SQL estudiados en estos artículos fueron métodos para Analizarlos, Usarlos y Solucionar.

En esta sección, se discute con mayor presión la información se recopilada. De los 39 artículos. Su rendimiento y clasificación se muestra en la Tabla 2.

Analizaron: A
Uso: U
Solución: S
Total: T

TABLA II

TIPOS DE INYECCIONES SQL DENTRO DE LOS ARTÍCULOS ANALIZADOS

Inyección SQL	A	U	S
Cadenas mal filtradas	AR26 AR37 AR39 AR8 AR17 AR 28 AR5 AR6 AR9 AR11 AR12 AR13 AR14 AR15 AR21 AR22 AR23 AR24 AR25 AR27 AR29 AR31 AR33 AR32 AR35 AR36	AR37 AR39 AR8 AR17 AR 28 AR5 AR6 AR9 AR11 AR12 AR13 AR14 AR15 AR21 AR22 AR23 AR24 AR25 AR29 AR31 AR32 AR36	AR26 AR37 AR39 AR8 AR17 AR 28 AR5 AR6 AR9 AR11 AR12 AR13 AR14 AR15 AR21 AR22 AR23 AR24 AR25 AR27 AR29 AR31 AR33 AR32 AR35 AR36
Incorrecta manipulación de tipo	AR17 AR 28 AR18 AR24 AR25 AR27 AR30 AR31 AR33 AR35 AR38	A17 AR 28 AR18 AR24 AR25 AR30 AR31 AR32 AR38	A17 AR 28 AR18 AR24 AR25 AR27 AR30 AR31 AR33 AR35 AR38
Evasión de Firma	AR 10 AR37 AR16 AR2 AR1 AR25 AR27 AR33	AR 10 AR37 AR16 AR2 AR1 AR25 AR	AR 10 AR37 AR16 AR2 AR1 AR25 AR27 AR33
Blind SQL Injection	AR3 AR4 AR7 AR14 AR20 AR25 AR27 AR31 AR33 AR34 AR36	AR3 AR4 AR7 AR20 AR25 AR31 AR34 AR36	AR3 AR7 AR20 AR25 AR27 AR31 AR33 AR34 AR36

A. Descripción de la tabla

Los resultados obtenidos en la Tabla II, arrojan como resultado, que más de la mitad de los artículos apuntan que el ataque de inyección SQL más común es Cadenas mal filtradas con una sumatoria de 26 artículos, seguido del tipo de inyección Incorrecta manipulación de tipo el mismo que va a la par con Blind SQL Injection los cuales se sitúan con 11 artículos y finalmente el tipo de inyección Evasión de Firma con 8 artículos.

B. Análisis de datos

Los resultados obtenidos y seguidamente tabulados se estudian con el fin de dar a conocer acerca de los tipos de inyecciones SQL, demuestran la cantidad de artículos que mediante el tipo de inyección la cadena mal filtrada analizó 26, se usaron 24 y finalmente dio solución 26 artículos, mediante el tipo de incorrecta manipulación de tipo analizaron 11, se usaron 9 y se solucionó a 11 artículos. Mediante el tipo de inyección Evasión de Firma se analizó 8, se usaron 7 da como solución a 8 artículos, y finalmente para el tipo de Blind SQL Injection se analizó en 11, se usó 8 y dan solución 11 artículos. Como se muestran en las Fig.8, y Fig.9.

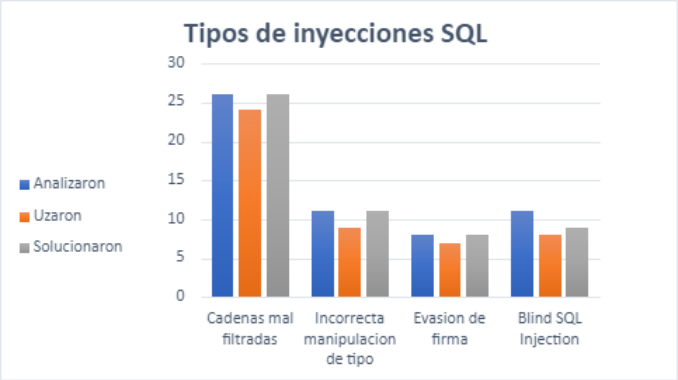

Fig. 8. Tipos de inyecciones dentro de los artículos.

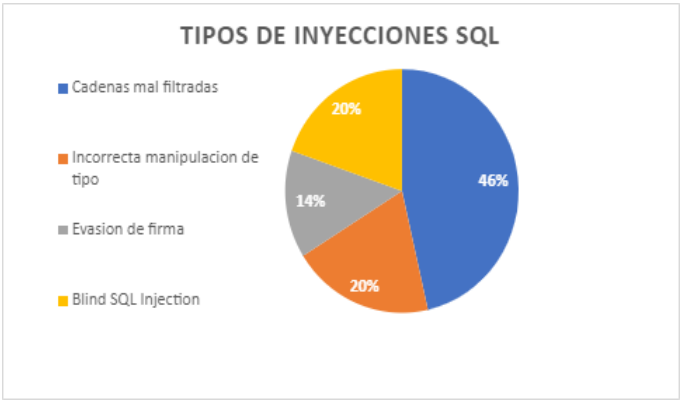

Fig. 9. Resultados dependientes del modo empleando los tipos de artículos dentro los artículos(porcentaje).

Para mayor entendimiento acerca de la investigación colocamos un breve resumen sobre qué es lo que hacen dentro de cada uno de los artículos investigados.

Para ello, realizamos una tabla con las revistas ordenadas según su año y su respectiva referencia.

Cuartil: Q

TABLA III
INFORMACIÓN RELEVANTE DE LOS ARTÍCULOS

INVESTIGADOS

N°	Año	Resumen	Referencia	Q
AR 1	2006	Los experimentos muestran que los métodos detectan inyecciones SQL y ataques.	[9]	Q4
AR 2	2009	Para detectar ataques, el controlador usa consultas SQL truncadas y seguimientos de pila para crear firmas de declaraciones SQL.	[10]	Q1
AR 3	2010	Presenta los métodos disponibles para proteger las aplicaciones web IIS ASP MSSQL contra ataques.	[11]	Q3
AR 4	2010	Simulan ataques SQLi contra bases de datos compatibles con MySQL.	[12]	Q3
AR 5	2010	Detectar fugas de mensajes de error y vulnerabilidades de inyección de SQL.	[13]	Q4
AR 6	2012	Utiliza una combinación de análisis estático y dinámico.	[14]	Q2
AR 7	2012	El marco crea un perfil del comportamiento legítimo de la base de datos derivado de las reglas de asociación de la aplicación y lo coloca en un archivo XML.	[15]	Q4
AR 8	2012	Se basa en detectar la entrada de texto confiable en vez de una cadena de texto maliciosa.	[16]	SN
AR 9	2012	Presenta un modelo para prevenir ataques de inyección de código SQL y secuencias de comandos entre sitios.	[17]	Q2
AR10	2013	Se ve implementación de Chroot, Apparmor, SELinux.	[18]	Q4
AR11	2014	Representan todas las posibles vulnerabilidades de consultas en tiempo de ejecución.	[19]	Q4
AR12	2015	Evita ciertos atributos en las consultas.	[20]	Q2
AR13	2015	Implementaron el modelo de Bloqueo de Detección contra ataques de inyección SQL.	[21]	Q3
AR14	2015	Evaluaron la implementación Java de SQLPIL utilizando un punto de referencia que incluye cinco aplicaciones comerciales JSP.	[22]	Q4
AR15	2015	Se ve un esquema SQLshield para prevenir ataques de inyección SQL en aplicaciones web que utiliza técnicas de aleatorización.	[8]	Q4
AR16	2015	Una de las más eficientes formas de evitarlos es evitando ciertos atributos en las consultas...	[23]	SN
AR17	2016	Las bases de datos no relacionales también son propensas a una inyección SQL.	[24]	SN
AR18	2016	Analizaron diferentes formas de inyección NoSQL mediante java script, identificaron diferentes formas de Código malicioso.	[25]	Q3
AR19	2017	En este artículo, proponen un nuevo método para prevenir ataques de inyección SQL. Donde han utilizado varias herramientas comunes de ataque de inyección SQL y conjuntos de datos de seguridad de aplicaciones web.	[26]	Q3
AR20	2017	Se aplica en MongoDB ejemplos de inyección y los enfoques de defensa.	[2]	SN
AR21	2017	Presenta un enfoque eficiente para probar operadores MOSA y las mutaciones basadas en él	[27]	Q4
AR22	2017	Muestra un método para detectar ataques de inyección SQL en sistemas operativos, IIS, bases de datos, etc.	[28]	SN
AR23	2018	Da una solución para clasificar consultas SQL con propiedades de la cadena de consulta original.	[29]	Q1
AR24	2018	Ataques básicos de inyección no SQL en MongoDB	[30]	SN
AR25	2018	Proporciona una mirada en profundidad a los métodos comunes de ataque de inyección de SQL.	[31]	SN
AR26	2018	Utilizaron técnicas de coincidencia de patrones y técnicas de procesamiento de tokens.	[32]	SN
AR27	2019	Se analiza como la inyección SQL causa mucho daño al sistema de red, provocando fugas de datos y parálisis del sitio web.	[33]	Q2
AR28	2019	Analiza todos los tipos de ataques y prevención de SQLinAs.	[34]	Q3
AR29	2019	Da a conocer que una aplicación web que ejecuta sentencias SQL dinámicas puede sufrir inyección de SQL.	[35]	Q2
AR30	2019	Para combatir los ataques de inyección de código como SQLInjection y Cross-Site Scripting (XSS), CODDLE implica el uso de redes neuronales profundas convolucionales.	[36]	Q2
AR31	2019	SQLi es una de las más vulnerables bases de datos, es muy propensa a inyecciones SQL.	[4]	Q1
AR32	2019	Las bases de datos no relacionales son propensas a una inyección SQL.	[6]	SN
AR33	2020	Describe los diferentes tipos de ataques de inyección SQL y las diversas herramientas y técnicas que pueden prevenir estos ataques.	[37]	Q4
AR34	2020	Propone una red neuronal artificial para la detección de ataques de inyección SQL.	[38]	Q1

AR35	2020	El SQL incorporado coloca sentencias SQL en el lenguaje de programación host y las ejecuta, inyección SQL en el lenguaje de programación C/C++ del host.	[39]	Q4
AR36	2020	Presenta una nueva plataforma de análisis y detección de vulnerabilidades basada en el tráfico de SQLIA llamada DIAVA que puede enviar alertas rápidamente a los inquilinos.	[40]	Q1
AR37	2021	Revisa búsquedas bibliográficas existentes y métodos de filtrado basados en bases de datos académicas (IEEE, ScienceDirect y Springer) .	[41]	Q3
AR38	2022	Solucionar los ataques contra una aplicación web de prueba de Acunetix construida con el lenguaje de programación PHP y una base de datos relacional MySQL.	[42]	Q3
AR39	2022	Lograron detectar en diferentes aplicaciones diferentes tipos de vulnerabilidades Mediante la implementación de algoritmos basados en la naturaleza.	[43]	Q4

VI. Conclusión

La seguridad de las bases de datos se ve comprometida por las aplicaciones web que interactúan con ella, si bien los métodos de seguridad han avanzado, siguen existiendo formas para vulnerar dicha seguridad. En el presente artículo se realizó un análisis de los diferentes métodos que se usan para vulnerar los sitios web, notando que existen cuatro tipos de que usan los hackers.

El método más común de inyección SQL y NoSQL se dan mediante cadenas te texto maliciosas debido a que ciertos aplicaciones web no se protegen ante la entrada de texto del usuario.

Otra de las formas más aplicadas para atacar al sitio es usar variables que no definen su tipo, es decir, usar una sentencia con una variable de tipo int sin identificarla se puede usar para cambiar e insertar código con datos de tipo string.

Mediante esta revisión bibliográfica emos concluidos que los mejores métodos para la prevención ante estos ataques son:

Declarar los tipos de datos que un usuario puede ingresar, verificar las URL, establecer los tipos de variables usadas en las sentencias y, otro de los métodos seria usar sentencias preparadas.

Existen más métodos para evitar la inyección de código malicioso como el uso de redes neuronales o implementación de aplicaciones que detecten este tipo de hackeos, pero nos emos centrado en los métodos más accesibles y recomendados.

VII. DESAFÍOS FUTUROS

Finalmente, En el desafío futuro, nos centraremos más en las bases de datos NoSQL, ya que no hemos encontrado tanta información requerida. Analizaremos varios tipos de ataques maliciosos a bases de datos NoSQL. Para lograr desarrollar un esquema de detección para prevenirlos y mantener una base de datos de información más segura contra las inyecciones.

VIII. Referencias

[1] M. Mauricio, 2013. [En línea]. Available: http://www.revistasbolivianas.ciencia.bo/scielo.php?pid=S1997-40442013000100017&script=sci_arttext. [Último acceso: 29 Noviembre 2022].

[2] B. Hou, Y. Shi, K. Qian y L. Tao , «Towards Analyzing MongoDB NoSQL Security and Designing Injection Defense Solution,» *IEEE ACCESS*, Enero 2017.

[3] J. A. Berlolin, Seguridad de Información Redes, informática y sistemas de información, M. J. Raso, Ed., Madrid: Paraninfo, 2009.

[4] L. Zhang, D. Zhang, C. Wang, J. Zhao y Z. Zhang, «ART4SQLi: The ART of SQL Injection Vulnerability Discovery,» *IEEE TRANSACTIONS ON RELIABILITY* , vol. 68, nº 4, Diciembre 2019.

[5] S. Cecchini y D. Gan, «SQL injection attacks with the AMPA suite,» *INTERNATIONAL JOURNAL OF ELECTRONIC SECURITY AND DIGITAL FORENSICS*, vol. 5, nº 2, Enero 2013.

[6] M. Ul Islam, M. Islam, Z. Ahmed, A. Iqbal y R. Shahriyar, «Automatic Detection of NoSQL Injection Using Supervised Learning,» *IEEE 43RD ANNUAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE (COMPSAC)*, vol. 1, Junio 2019.

[7] A. Ron, A. Shulman-Peleg y A. Puzanov, «Analysis and Mitigation of NoSQL Injections,» *IEEE SECURITY & PRIVACY* , vol. 14, nº 2, Marzo 2016.

[8] P. S. y. S. S. Kar Debabrata, «SQLiDDS: SQL Injection Detection Using Query Transformation and Document Similarity,» *LECTURE NOTES IN ARTIFICIAL INTELLIGENCE,* vol. 8956, Junio 2015.

[9] J.-C. &. N. B. N. Parque, «SQL injection attack detection:Profiling of web application parameter using the

sequence pairwise alignment,» *LECTURE NOTES IN ARTIFICIAL INTELLIGENCE*, vol. 4298, Enero 2006.

[10] M. D. y. S. Diomidis., «SDriver: Location-specific signatures prevent SQL injection attacks,» *COMPUTERS & SECURITY*, vol. 28, nº 3-4, Mayo 2009.

[11] W. Beihua , «SQL Injection Defense Mechanisms for IIS plus ASP plus MSSQL Web Applications,» *CHINA COMMUNICATIONS*, vol. 7, nº 6, Diciembre 2010.

[12] D. E. Simos, «Combinatorial methods for dynamic gray-box SQL injection testing,» *SOFTWARE TESTING VERIFICATION & RELIABILITY*, vol. 32, nº 6, Septiembre 2022.

[13] W. L. y. A. A. Smith Ben, «Idea: Using System Level Testing for Revealing SQL Injection-Related Error Message Information Leaks,» *LECTURE NOTES IN ARTIFICIAL INTELLIGENCE*, vol. 5965 , Enero 2010.

[14] I. Lee, S. Jeong y S. &. L. J. Yeo, «A novel method for SQL injection attack detection based on removing SQL query attribute values,» *MATHEMATICAL AND COMPUTER MODELLING*, vol. 55, nº 1-2, Enero 2012.

[15] M. M. I. E.-F. L. y. H. Y. K. Salama Shaimaa Ezzat, «Web Anomaly Misuse Intrusion Detection Framework for SQL Injection Detection,» *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, vol. 3, nº 3, Marzo 2012.

[16] D. Giri, S. Kumar, L. Prasannakumar y R. Murthy, «OBJECT ORIENTED APPROACH TO SQL INJECTION PREVENTER,» *THIRD INTERNATIONAL CONFERENCE ON COMPUTING COMMUNICATION & NETWORKING TECHNOLOGIES (ICCCNT)*, Julio 2012.

[17] P. Sharma, R. Johari y S. S. Sarma, «Integrated approach to prevent SQL injection attack and reflected cross site scripting attack,» *INTERNATIONAL JOURNAL OF SYSTEM ASSURANCE ENGINEERING AND MANAGEMENT*, vol. 3, nº 4, Diciembre 2012.

[18] S. Cecchini y D. Gan, «SQL injection attacks with the AMPA suite,» *INTERNATIONAL JOURNAL OF ELECTRONIC SECURITY AND DIGITAL FORENSICS* , vol. 5, nº 2, Enero 2013.

[19] P. A.-S. K. y. K. D. Abdoulaye, «Lethality of SQL injection against current and future internet technologies,» *INTERNATIONAL JOURNAL OF COMPUTATIONAL SCIENCE AND ENGINEERING*, vol. 9, nº 4, 2014.

[20] D. G. Kumar y M. & Chatterjee, «MAC based solution for SQL injection,» *JOURNAL OF COMPUTER VIROLOGY AND HACKING TECHNIQUES*, vol. 11, nº 1, Agosto 2015.

[21] D. G. Kumar y M. & Chatterjee, «SQLPIL: SQL injection prevention by input labeling,» *SECURITY AND COMMUNICATION NETWORKS*, vol. 8, nº 15, Septiembre 2015.

[22] J. S. y. M. L. D. Punit Mehta, «SQLshield: Preventing SQL Injection Attacks by Modifying User Input Data,» *LECTURE NOTES IN ARTIFICIAL INTELLIGENCE*, vol. 9478, Junio 2015.

[23] J. Abirami, R. Devakunchari y C. Valliyammai, «A Top Web Security Vulnerability SQL Injection attack - Survey,» *SEVENTH INTERNATIONAL CONFERENCE ON ADVANCED COMPUTING (ICOAC)*, Enero 2015.

[24] B. Hou, K. Qian, L. Li, Y. Shi, L. Tao y J. Liu, «MongoDB NoSQL Injection Analysis and Detection,» *IEEE 3RD INTERNATIONAL CONFERENCE ON CYBER SECURITY AND CLOUD COMPUTING (CSCLOUD)*, Diciembre 2016.

[25] A. Ron, A. Shulman-Peleg y A. Puzanov, «Analysis and Mitigation of NoSQL Injections,» *Analysis and Mitigation of NoSQL Injections*, vol. 14, nº 2, Marzo 2016.

[26] A. K. Dalai y S. K. & Jena, «Neutralizing SQL Injection Attack Using Server Side Code Modification in Web Applications,» *SECURITY AND COMMUNICATION NETWORKS*, Mayo 2017.

[27] Y. Huang, C. Fu, X. Chen, H. Guo, X. Él, J. Li y Z. Liu, «A Mutation Approach of Detecting SQL Injection Vulnerabilities,» *IEEE 3RD INTERNATIONAL CONFERENCE ON BIG DATA SECURITY ON CLOUD (BIGDATASECURITY, IEEE 3RD INTERNATIONAL CONFERENCE ON HIGH PERFORMANCE AND SMART COMPUTING, (HPSC) AND 2ND IEEE INTERNATIONAL CONFERENCE ON INTELLIGENT DATA AND SECURITY (IDS)*, Enero 2017.

[28] H. Hu, «Research on the Technology of Detecting the SQL Injection Attack and Non-Intrusive Prevention in WEB System,» *MATERIALS SCIENCE, ENERGY TECHNOLOGY, AND POWER ENGINEERING I*, vol. 1839, Diciembre 2017.

[29] P. R. K. K. &. S. Q. McWhirter, «SQL Injection Attack classification through the feature extraction of SQL query strings using a Gap-Weighted String Subsequence

Kerne,» *JOURNAL OF INFORMATION SECURITY AND APPLICATIONS*, vol. 40, Junio 2018.

[30] V. Sachdeva y S. Gupta, «BASIC NOSQL INJECTION ANALYSIS AND DETECTION ON MONGODB,» *INTERNATIONAL CONFERENCE ON ADVANCED COMPUTATION AND TELECOMMUNICATION (ICACAT)*, Enero 2018.

[31] H. y. Z. X. Zhang, «SQL Injection Attack Principles and Preventive Techniques for PHP Site,» *Maquinaria Assoc Comp*, Octubre 2018.

[32] R. B. V. A. B. .. N. A. J. Rajadurai, «Securing SQL Injection flaw,» *BIOSCIENCE BIOTECHNOLOGY RESEARCH COMMUNICATIONS*, vol. 11, nº 1, Enero 2018.

[33] Q. Li, W. Li y J. &. C. M. Wang, «A SQL Injection Detection Method Based on Adaptive Deep Forest,» *IEEE ACCESS*, vol. 7, Diciembre 2019.

[34] K. y. U. R. L. Varshney, «LsSQLIDP : Literature survey on SQL injection detection and prevention techniques,» *JOURNAL OF STATISTICS AND MANAGEMENT SYSTEMS*, vol. 22, nº 2, Febrero 2019.

[35] S. U. B. D. K. Das Debasish, «Defeating SQL injection attack in authentication security: an experimental study,» *INTERNATIONAL JOURNAL OF INFORMATION SECURITY*, vol. 18, nº 1, Febrero 2019.

[36] A. S. y. B. Giuseppe., «CODDLE: Code-Injection Detection With Deep Learning,» *IEEE ACCESS*, vol. 7, Octubre 2019.

[37] N. Ljubicic y D. &. P. P. Jaksic, «SQL INJECTION - PREVENTION AND DEFENSE,» *ZBORNIK VELEUCILISTA U RIJECI-JOURNAL OF THE POLYTECHNICS OF RIJEKA*, vol. 8, nº 1, Junio 2020.

[38] P. Tang, W. Qiu, Z. Huang y H. &. L. G. Lian, «Detection of SQL injection based on artificial neural network,» *KNOWLEDGE-BASED SYSTEMS*, vol. 190, nº 105528, Febrero 2020.

[39] J. Young-Su, «Detection of SQL Injection Vulnerability in Embedded SQL,» *TRANSACCIONES IEICE SOBRE INFORMACIÓN Y SISTEMAS*, vol. E103D, nº 5, Mayo 2020.

[40] H. Gu, J. Zhang, T. Liu, M. Hu, J. Zhou, T. Wei y M. Chen, «DIAVA: A TraffiDIAVA: A Traffic-Based Framework for Detection of SQL Injection Attacks and Vulnerability Analysis of Leaked Data,» *TRANSACCIONES IEEE SOBRE CONFIABILIDAD*, vol. 69, nº 1, Marzo 2020.

[41] A. A. Q. M.-Q. R. NasereddinMohammed, «A systematic review of detection and prevention techniques of SQL injection attacks,» *INFORMATION SECURITY JOURNAL*, Octubre 2021.

[42] H. L. H. T. C. Bedekovic Nenad, «The Importance of Developing Preventive Techniques for SQL Injection Attacks,» *TEHNICKI GLASNIK-TECHNICAL JOURNAL*, vol. 16, nº 4, Septiembre 2022.

[43] K. Baptista, A. Bernardino y E. Bernardino, «Detecting SQL Injection Vulnerabilities Using Nature-inspired Algorithms,» *LECTURE NOTES IN ARTIFICIAL INTELLIGENCE*, vol. 5, Diciembre 2022.

IX. Agradecimientos

Para el desarrollo del presente articulo agradeceremos de manera especial a la ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO-SEDE ORELLLANA, alma máster de la ciencia y la tecnología, a la facultad Informática y Electrónica. De igual manera a mis queridos docentes en especial al PHD. WILSON GUSTAVO CHANGO SAILEMA, docente de la materia Base de datos avanzados.

De igual forma un agradecimiento especial a vuestros padres, hermanos, compañeros y amigos por estar a nuestro lado y por ser parte de este gratificante y arduo trabajo.