A Thorough Study On Sql Injection Attack-Detection And Prevention Techniques And Research Issues

Shobana R.
Govt. Arts College for men, Nandanam
3 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Dr M Suriakala
Government Arts College for Men chennai
47 PUBLICATIONS **37** CITATIONS
SEE PROFILE

A Thorough Study On Sql Injection Attack- Detection And Prevention Techniques And Research Issues

R. Shobana[1],*MCA.,M.Phil.*, and **Dr. M. Suriakala**[2],*M.Sc.,M.Phil.,Ph.D.,*

[1] Part time Research Scholar, University of Madras,
Assistant Professor, Department of Computer Science and Applications,
D.K.M. College for Women, Vellore- 1
[2] Assistant Professor, Department of Computer science,
Government Arts College for Men, Nandanam, Chennai-35

Abstract

SQL Injection is one of the big and dangerous threat which can be done by the hackers by stealing electronic records at the backend of the web application and hence the intruders will steal any kind of sensitive data at the anytime by injecting malicious statement to SQL queries into the database. It can be prevented by using more techniques but still the hackers find holes to carry out the SQL injection like inserting, altering and deleting the record by unrestricted access to the database. The vulnerability of SQL injection leads to loss of confidentiality and integrity. This paper comes out with a review of different types of SQL injection attack, their detection and prevention techniques.

Keywords: *SQL Injection, Database, detection, prevention.*

1. Introduction

Structured Query Language Injection Attack is the expansion of SQLIA. The main usage of SQL is to work with data in the database so that the data can be manipulated by interacting SQL into the database. So that the intruders use malicious SQL queries to steal sensitive information from the database server which is executed over web based application. Most of the database like Banking, Finance sector, Health care and Employee details were the common vulnerable web based application attack. The different kinds of attacks are: String SQL Injection, Numeric SQL Injection, Comments attack, Blind SQL injection, Timing attacks, Command union SQL Injection.

In three tire web applications, the user provides query specification as input fields in predefined input form. These input values are used to construct SQL queries by the application server in the middle tire. Common web applications include web mail, online retail sales, online auctions, online banking, and many other functional applications. There are two types of web applications:

Presentation-oriented: A presentation-oriented web application generates interactive web pages in various types of markup language (HTML, XML, and so on) and containing dynamic content in response to requests. Web applications are popular due to the ubiquity of web browsers, and the convenience of using a web browsers as a client, sometimes called a thin client. The ability to update and maintain web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity.

Service-oriented: A service-oriented web application implements the endpoint of a web service. Presentation-oriented applications are often clients of service-oriented web applications.

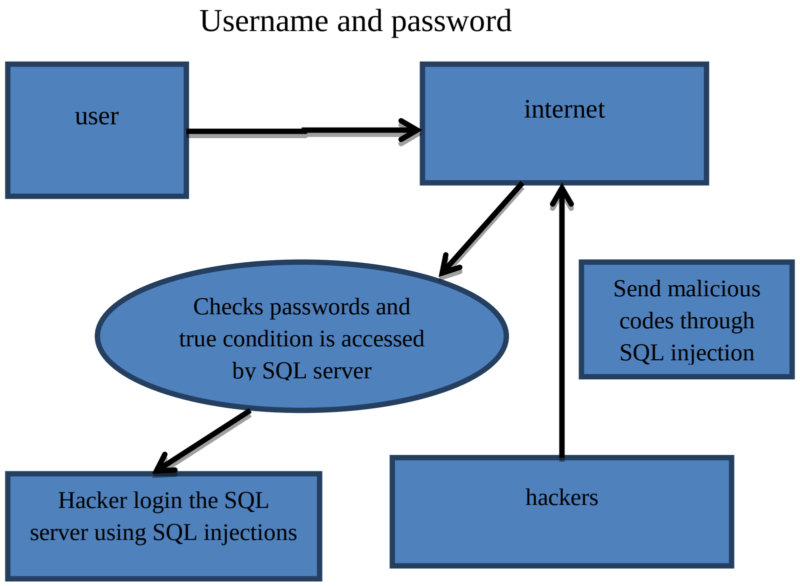
Username and password
user
internet
Checks passwords and true condition is accessed by SQL server
Send malicious codes through SQL injection
Hacker login the SQL server using SQL injections
hackers

Figure 1- Overview of SQL injection attack

The overviews of SQL injection are given in Figure 1. Where user enters the database by using authenticated username and password. But the intruders will enter into the database by injecting malicious query. The query is then checked with the threshold value (Assuming threshold as 50) if the string value is lesser than the threshold value, then it allows the username name and password enter into the database and if the string value is greater than the threshold value, it generates the alarm to the admin .So that, the condition used by the hackers will always produces the output true through SQL injection if the threshold value is greater. This vulnerability results in loss of confidentiality and integrity of data. Most detection and prevention techniques help to secure the database over the web application but still hackers have some loop holes to perform SQL injection. This study helps us to provide a background of detection and prevention method for future work and hence it enables the research to identify the SQLIA.

The various types of SQL injection attacks are

i. Tautology attack
ii. Union Queries attack
iii. Piggy- Backed attack
iv. Timing attack
v. Blind SQL Injection attack

vi. Logically Incorrect Queries
vii. Alternate Encodings

i. Tautology attack

SQLIA can be done by inserting malicious string/ numeric values into the conditional query statement. This type of attack always brings conditions true while the SQL statements are injected into the queries.

e.g., Select * from user where username=’1’ or ‘1=1’—‘and password=’1234’;

The above query makes the application insecure. The username and password used by the hacker would always return the condition true, it does not return any error message and the login parameter is vulnerable to the injection and hence it allows the hacker to enter into the database. Here, the input used can be either numeric values or string to the conditional query statement.

ii. Union Queries

This type of attacks occurs by injecting the malicious query with the safe query by using the keyword UNION to get useful information from the data base

e.g., Select * from user where username=’1111’ UNION Select * from member where member name=’admin’—‘and password=’1234’;

The above query concatenates the safe query with the malicious query by using the keyword UNION and makes the statement vulnerable. In the first scenario, query contains the username and in the second scenario, query has both username and password. Select statement with UNION keyword will fill the missing column “password” and manipulate the output in order to retrieve the desired values.

iii. Piggy- Backed attack

This type of attack is used to inject additional query statements to the original query by using a query delimiter (;). Hence, the first query is original and the subsequent query is considered to be an injected query.

e.g., Select ID from user where username=’user1’ and password=0; drop user

The above query combines the original query which contains “username” and “password” with the malicious query by using the delimiter (;). Hence the second injected query makes the statement vulnerable to the database and drops the user table.

iv. Timing attack

This type of attack occurs by using if condition statement and WAITFOR along the branches and hence an attackers collects the information from the database and by observing the time delays in the database responses. The attackers who use WAITFOR in the conditional statements to delay its response for a specified time.

WAITFOR delay ‘0:0:5’ will make the database to stop for five seconds. Hence the vulnerable code delay in response time if the condition is true.

V. Blind SQL Injection attack

This type of attack occurs by asking logical questions through SQL Statements if the database applications are insecure and hence the error message will allow the intruders to compromise the database. The web application displays error message from the database when the attacker asks true or false questions to the database.

e.g., Select ID from user where username=’user’ and 1=0—AND pass= AND pin=0

Select info from user where username=’user’ and 1 =1—AND pass= AND pass=0

If both the queries have strong input validation, then the application would be strong enough. If the queries have weak input validation, first it checks the first query, it always returns the error message because of value “1=0” which is not true. Then it checks the second query, it returns no error message because of value “1=1”, then the attackers search the field vulnerable to the injection.

Vi. Logically Incorrect Queries

The error message displayed by the database often contains useful information will help the attacker to find out the vulnerable parameter in an application

e.g., Select * from user where username=’1111’ and password=’1234’ and convert (char,no)

The above query makes the database vulnerable and return the false value because, the type of the field varies and hence the query injected makes the hacker to get the useful information returned by the error message.

vii. Alternate Encodings

This type of attack occurs when the attacker modify the injected query using Hexadecimal, ASCII and UNI code. Hence the developer should be familiar with the types of attack to identify the defence coding to prevent it from attacks

e.g., Select accounts from user where login=” AND pin=0; exec (char (7564f73ex0))

The above query contains hexadecimal, ASCII and UNI code. The hacker alter the query by injecting different types of code and makes the statement vulnerable to the database.

Survey on various detection and prevention techniques

Frantisek Franek et al., (2017) proposed hybridization of Knuth–Morris–Pratt (KMP) and Boyer–Moore (BM) algorithm which guaranteed both independence from alphabet size and worst-caseexecution time linear in the pattern length. Experiments indicate that in practice the new algorithm is among the fastest exact pattern-matching algorithms discovered to date, apparently dominant for alphabet size above 15–20.

TehFaradilla Abdul Rahman et al., (2015) proposed a detection model to scan SQL injection on the web environment, based on the defined and identified criteria using the Boyer-Moore String Matching Algorithm. From several tests that had been done, the results showed that the proposed model is able to detect vulnerable web applications with the defined criteria of the SQL Injection. In conclusion, this proposed model can be used by web application developer and system admin to secure the application from being attacked and compromised.

SudhaSenthilkumar et al.,(2017)proposed techniques bruto force pattern matching checks the vulnerability of SQL Injection and accordingly takes action. If the vulnerability at one level is less, it skips the rest of the checks and hence making the process efficient. By incorporating the best features of stated techniques, an algorithm is successfully proposed for tackling SQL Injection.

Jalel Rejeb et al., (2014) proposed an extension to the Aho-Corasick algorithm to detect injection of characters introduced by a malicious attacker. They showed that how this is achieved without significantly increasing the size of the finite-state pattern matching machine of the Aho-Corasick algorithm. Moreover, the machine would detect a match only if the number of stuffed characters is within a specified limit so that the number of false positives remains low. A comparison of the CPU time consumption between Aho-Corasick algorithm and the proposed algorithm is provided. It was found the proposed algorithm can outperform the Aho-Corasick, while ignoring the stuffed characters and detecting a valid match.

SaqibHakak et al., (2018) proposed a novel idea for exact string matching to achieve both time and space efficiency regardless of query pattern length, dataset size and scripts. The proposed algorithm split given query pattern length into two halves and then it considers right halve for searching in a text. Once the match is found for right halve, the proposed algorithm uses left halve directly from the matched reference. This process helps in reducing the number of computation especially comparisons at the same time it consumes less memory due to no pre-processing involved as compared to existing exact matching algorithms.

Renu et al.,(2017) proposed an effective string matching algorithm called String Count. Protection of data within and outside of the organization is a huge risk. Inorder to overcome this dicey situation organizations use different data protection techniques and policies. Organize data in a planned and productive way will helps to implement privacy effectively. String count can be calculated by adding ascii equivalent of the alternate alphabets of the string .

Tatiana et al. (2015) proposed novel online approach to exact string matching and filtering of large database. Initially, a self-organizing map retrieves the cluster of database strings that are most similar to the query string; subsequently, a harmony theory network compares the retrieved strings with the query string and determines whether an exact match exists. The similarity measure is configured to the specific characteristics of the database so as to expose overall string similarity rather than character coincidence at homologous string locations. The experimental results demonstrate foolproof, fast, and practically database-size independent operation that is especially robust to database modifications.

Zhuang Chen et al., (2018) proposed a SQL detection method based on machine learning, word vector text is the representation method. It combines word2evc http request processing with support vector machine for classification and produces the final classification results. It overcomes the existing rule matching method and the advantage of SQL detection methods are security detection, overcomes SQL injection mutation. The future work of this detection method needs required memory for large amount of dataset and this high quality data sets need known SQL injection samples and requested samples.

Zainab S. Alwan et al., (2017) analysis SQLIA is one of the vulnerable attack against mobile, web and desktop applications. This survey paper presents different types of detection and prevention techniques, its method and the types of attack. This paper compares different detection method and analyzed some techniques should be improved to overcome SQLIA.

Payal Zade et al., (2017) proposed a certification and shrinking method for SQL injection attack using Aho- Corasick pattern matching algorithm. This method is used to detect and prevent SQLIA by using string pattern matching technique. It can be extended to modify the pattern matching algorithm to double step process and hence provide security to database against SQLIA.

Dr. Ahmad Ghafarian et al.,(2017) proposed a hybrid technique for the detection and prevention of SQLIA. This method consists of three phases namely design, implementation and CGI. The advantage of this method is, it can handle any type of queries and it is platform independent. This research can be extended at theoretical stage and can test the performance. The algorithm can be extended to other types of SQLIA.

Nency Patel (2015) presents a method based on modified Aho-corasick algorithm which is used to detect and prevent SQLIA. This method reduces the memory space by storing the repeated word only once into the database. Here, the SQLIA can be detected and prevented by using SQLMAP tool and AIIDA-SQL technique by checking whether the user generated queries are injected or not and the static pattern matching algorithm is used to check the user generated queries. The future work of this research is to extend the pattern matching algorithm to check whether the user generated queries are injected to overcome the SQLIA tools

Rhythm Dubey et al., (2016) proposed a method based on hash value and proxy server which is used to secure the database against SQL injection attack. Another two phase technique will be used for two check authentication to prevent SQLIA. The future work of this research can be extended to technique and hence save the time, memory effective and cost efficient.

Vamshi Krishna Gudipati et al., (2016) presented a technique based on defensive mechanism using kali linux which does not allow the hacker’s SQL malicious code to the website without penetration testing and hence the database cannot be breached by the hackers and advanced methods are used to prevent SQL injection attack. The paper can be enhanced by providing penetration test to all the websites consciously

Benfano Soewito et al., (2018) proposed a technique based on escape string and URL parameters. Injection can be prevented by identifying the query and SQL type of URL parameter and user column type by grouping their types. The vulnerable attack can be carried out by the URL parameter with text type. This can be prevented by filtering, validating and processing the parameter before sending it to the database. This paper concluded with the effective prevent system and exploit the SQL injection attack.

Table 1. Comparative analysis of SQL attack detection and prevention techniques

Author	**Techniques/Methods used**	**Advantages**	**Future Work**	**Area of Focus**
Frantisek Franek	hybridization of Knuth–Morris–Pratt (KMP) and Boyer–Moore (BM) algorithm	Independent from both alphabet size and execution time	Pattern matching algorithm can be improved by carrying out the alphabet size >20	SQL injection prevention
TehFaradilla	Boyer-Moore String Matching Algorithm	Admin can secure the application from vulnerable attack	Different attacks should be prevented over different databases	SQL injection detection and prevention
Sudha Senthilkumar	bruto force pattern matching	Process efficient and skips the remaining check if the injection is lesser	It can be carried out for number of attacks and more number of databases	SQL injection detection and prevention
Jalel Rejeb	Aho-Corasick pattern matching algorithm	Ignore the stuffed character and find the exact match	It can be extended to check stuffed string too	SQL injection prevention
SaqibHakak	Query pattern matching algorithm	Split the query into two halves and find the match for one half will reduce the time and memory	Matching of one half of the query can result sometime to injection and hence should improve the memory efficiency	SQL injection detection and prevention
Renu	String count algorithm	Algorithm used to count the string in terms of ascii	String count can be carried out in future by using hash value	SQL injection prevention
Tatiana	String matching and filtering	Similarity of existing string to the original string find match only for string s not for characters	String matching should be carried out for character too	SQL injection detection and prevention
Zhuang Chen	Machine learning algorithm used.	Very good classification results	Required High quality datasets, more memory space depends on data sets	SQL injection detection

			and known SQL samples	
Zainab S. Alwan	Different existing tools and techniques like ML, SQL mao, Noxes and ardilla tool	Different types of attacks can be prevented by using different types of techniques like ML, SQL mao, Noxes and ardilla tool	Efficiency of some techniques should be improved against SQLIA	SQL injection detection and prevention
Payal Zade	Aho- Corasick pattern matching algorithm	Various databases are used, certification and shrinking methods are used	Required static and dynamic together	SQL injection detection
Dr. Ahmad Ghafarian	Static and dynamic method	Handle any type of queries and platform independent	The future work can be extended to test the performance, tested against existing methods and can include other types of SQLIAs	SQL injection detection and prevention
Nency Patel	Modified Aho-Corasick pattern matching algorithm	SQLMAP tool, ANN scheme and search algorithm is used to prevent SQL injection attack	Lack of memory consumption should be improved	SQL injection detection
Rhythm Dubey	Integration of Proxy filtering and SQLIPA technique	It will double check the login process	Proxy filtering should be implemented and make it as more effective, user friendly and cost efficient	SQL injection prevention
Vamshi Krishna Gudipati	Kali linux	Penetration testing prevents SQL injection attack	Penetration test should be carried out for all the websites	SQL injection prevention
Benfano Soewito	Regular expression and escape string	Validation, filtering, processing over URL parameters secure databases against SQLIA	Validate handling solution should be more efficient and consistent	SQL injection prevention

Conclusion

This paper analysed the various types of SQLIA, and its related literature study in detail. From the extensive literature study, it is observed that the Researchers have developed many techniques to detect and prevent this vulnerability of the web application through SQLIA. There is no appropriate solution that can prevent all types of SQL injection attacks. The techniques are summarized and their parameters are also discussed. To overcome some of the limitations from above mentioned techniques, a novel technique has to be developed by improving the existing techniques by the considering future work mentioned.

Reference

1. *Renu, S., and SH Krishna Veni. "An Enhanced CIA tree Using String Matching Algorithm." International Journal of Applied Engineering Research 12.16 (2017): 6123-6126.*
2. *TehFaradilla Abdul Rahman ,AlyaGeogianaBuja, KamarularifinAbd. Jalil, FakariahMohd Al," SQL Injection Attack Scanner Using Boyer-Moore String Matching Algorithm" Journal of Computers, Volume 12, Number 2, March 2017*
3. *Frantisek Franek, Christopher G. Jenningsb,W.F.Smyth" A simple fast hybrid pattern-matching algorithm" Journal of Discrete Algorithms (2017), pp. 135–144*
4. *B. Jakub, P. Buciak, and P. Sapiecha. "Building dependable intrusion prevention systems." Dependability of Computer Systems, International Conference on. IEEE, 2016.*
5. *SudhaSenthilkumarand Krishna TejaReddy,"Preventing SQL Injection Attack Using Pattern Matching, Parse Tree Validation and Cryptography Algorithm",Journal of Environmental Science, Computer Science and Engineering & Technology" Vol.6. No.4, 246-253,Nov-2017*
6. *Jalel RejebMahalakshmi Srinivasan," Extension of Aho-Corasick Algorithm to Detect Injection Attacks", Advances in Computer and Information Sciences and Engineering, pp 207-212,June 2015*
7. *Zainab S, and Manal F. Younis. "Detection and Prevention of SQL Injection Attack: A Survey." International Journal of Computer Science and Mobile Computing (2017): 5 - 17.*