Proceedings of the Second International Conference on Inventive Systems and Control (ICISC 2018)
IEEE Xplore Compliant - Part Number:CFP18J06-ART, ISBN:978-1-5386-0807-4; DVD Part Number:CFP18J06DVD, ISBN:978-1-5386-0806-7

# *Detection of SQL Injection Attacks by Removing the Parameter Values of SQL Query*

Rajashree A. Katole
SGBAU, Amravati,
Maharashtra, India.

Dr. Swati S. Sherekar
SGBAU, Amravati,
Maharashtra, India

Dr. Vilas M. Thakare
SGBAU, Amravati,
Maharashtra, India

. ***Abstract*— Internet users are increasing day by day. The web services and mobile web applications or desktop web application's demands are also increasing. The chances of a system being hacked are also increasing. All web applications maintain data at the backend database from which results are retrieved. As web applications can be accessed from anywhere all around the world which must be available to all the users of the web application. SQL injection attack is nowadays one of the topmost threats for security of web applications. By using SQL injection attackers can steal confidential information. In this paper, the SQL injection attack detection method by removing the parameter values of the SQL query is discussed and results are presented.**

***Keywords— internet, web services, web applications, database, SQL Injection, SQL Injection attacks, parameter values***

## I. INTRODUCTION

Due to the more digitalization of world, usage of mobile phones, computers, tablets etc. is increasing very fast. With boosting of digitalization and internet the usage of web applications has also increased. Many of the web application has a three- tier construction i.e. Presentation tier, CGI tier, and Database tier. SQL injection attack is also known as SQL insertion attack. Due to advances in internet, most offline services have moved online. These online services use web applications and web services. Most web attacks target the vulnerabilities of web applications. The SQL Injection Attack (SQLIA) does not waste system resources as other attacks do. However, because of its ability to obtain/insert information from/to databases, it is a strong threat to servers like military or banking systems. The web application framework uses filtering methods for data inputted by user [1]. By the development of the information technology, a massive amount of sensitive information is stored in the database. This information is most valuable for the organizations. Database intrusion attacks can occur for stealing the valuable and sensitive information. Database intrusion attacks could be broadly categorized into two types, depending on the access point. In the first type, the malicious user is directly given the permission to access the database to fetch the data. In the second type, the malicious user indirectly accesses the database to obtain the data. The malicious user can access database indirectly by altering the SQL statements. Such attacks are known as SQL injection attacks [2]. SQL injection is the most widespread security issue in the web applications. Code injection attacks consists of SQL injection attacks, in which SQL characters are inserted into the SQL statements using an untrusted access to change the logic or meaning of the intended query. When the SQL statements is constructed using the external input data, the threat of SQL injection is found. The attacker could modify the query statements by modifying or altering the input data. The SQL injection attack occurs due to lack of development time and training, lack of experience and knowledge of potential security issues, developers often misuse these methods which results in SQL injection vulnerabilities (SQLIVs) [3]. Many businesses are conducted over the internet, cyber-security threats increases. One of the most popular cyber-security threats is Structured Query Language injection (SQLi). It is the most common form of vulnerability in web applications. The risk can also be present with the accounting perspective due to the cyber-security threats. SQLi can also be defined as unauthorized access to data, as well as unauthorized inserts, updates, and deletes of data [4]. Web applications have a front-end and back-end. The front-end is available to clients, employees i.e. the screen which can be seen on the device. An Error can be defined as a deviation of an external state of the system from the correct service state. A fault can be defined as the adjudged or hypothesized cause of an error. Vulnerability can be defined as an internal fault that enables an external fault to harm the system and an attack can be defined as a malicious external fault. The security of web applications is a major concern. Most web applications have critical bugs or faults which can affect the security [5]. Another way of providing security to the web applications from SQLIA is use of Intrusion Detection Systems [6]. The prime issue is that a

database and the input field do not have the ability to differentiate between a valid input string and a malicious input string [7]. An SQL Injection Attack is a code injection technique that exploits a security vulnerability occurring in the database layer of an application. The vulnerability occurs if the user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or the user input is not strongly typed [8].

In this paper method of query processing and parameter removal are collectively used to give a method which detects the difference between the original SQL query and the modified i.e. injected SQL query via the parameters. Evolution of approaches from web application generation to web application security is making the web applications more secure and robust. Removing the parameters and comparing them with the original query structure will improve the performance of the system.

## II. RELATED WORK

The web applications are improving day by day with respect to many parameters such as security, storage space, browser compatibility etc. But still there are some chances of attacks which damage the secure web application. Many methodologies are present to detect, prevent and recover from these types of attacks; some of such methodologies are as follows:

*A. A combination of SQL query parameter removal and combined static and dynamic analysis method:* A very simple and effective method to accurately detect SQL injection attacks by using a combination of SQL query parameter removal and combined static and dynamic analysis methods. The effectiveness of this method has been tested and validated using web applications. This method uses a combination of SQL query parameter removal and combined static and dynamic analysis methods. The web application uses three tiers. The data processing is carried out at the CGI tier where input from the presentation tier is collected, the processing is done with the help of the storing and retrieving of data from the database tier and the result is displayed at the presentation tier. The effectiveness of this method has been tested and validated using web applications. On comparing the method with other detection methods and showed which makes it independent of the DBMS. Complex operations such as parse trees or particular libraries are not needed in the proposed method. Future work is needed for not only SQL injection attacks but also for other web application attacks such as XSS, based on the proposed method and machine learning algorithms [1].

*B. The query tree approach:* A method to detect SQLIA efficiently and accurately at the database level. To detect the SQLIA the query tree approach is used, which is an internal representation of an SQL statement written in database logs. The malicious query trees are separated from normal query trees. The data-mining based method to detect SQLIA is beneficial in detecting unknown attacks with high accuracy against the rapid emergence of various forms of attack. A novel method is proposed to convert the query tree into an n-dimensional feature vector by using a multi-dimensional sequence as an intermediate representation. A method is proposed to transform string feature values into numeric feature values, combining multiple statistical models. The combined model maps one string value to one numeric value by containing the multiple characteristic of each string value. For evaluating the performance of data-mining based classifiers, k-fold cross validation is usually used. A large amount of time is consumed to generate a multi-dimensional sequence from a query tree [2].

*C. comparison of different techniques for SQLIA detection:* This study helps to find new solutions and new techniques that defeats the SQL injection attacks which is a serious issue. By finding new solutions for this issue the precision and effectiveness is improved. Paper to investigate the security called SQL injection attack different tools are proposed by the researchers. These tools are studied and analysed one by one. Comparison of tools is made to check which tool is precise and more efficient to detect and defeat the SQL injection attacks. By combining these techniques several new solutions could be obtained for solving the security issue. Runtime prevention approaches require dynamic monitoring systems but it could prevent all attacks. Dynamic monitoring system is required for runtime prevention approaches. Vulnerability detection approaches identifies all vulnerabilities but it can generate many false alarms. Developers must be provided with adequate training to raise security awareness and get familiar with existing state-of-the- art defense techniques and tools. Researchers must find simple ways to effectively combine existing defense methods for overcoming the limitations of each individual method rather than finding new engineering methods to address SQL injection. Too often, researchers' state-of-the-art techniques are either not available for use or difficult to be adopted. The readily available tools and the easy adoption of combined defense methods would motivate developers to deal with security and finally defeat SQL injection. [3].

*D. the cyber-security threat- SQL injection:* This considers both the parameters: a Microsoft Access database and also the web- based environment. The cyber-security threat i.e. SQL injection is presented as a case which explores how the SQL injection operates. The questions are answered like how cyber-security threats works and what are the actions that are to be

taken to reduce the risk. A Microsoft Access database and the web-based environment both the parameters are considered. The new models which can perform better must be found for reducing the risk of SQLi. The models must be studied properly [4].

*E. The characteristics of source code defects:* The characteristics of source code defects generating major web application vulnerabilities are studied. The software faults which leads to vulnerabilities in the web application developed using different programming languages. The developer should avoid common mistakes while developing the web applications. A field study on two of the most widely spread and critical web application vulnerabilities i.e. SQL Injection and XSS. The source code of security patches of widely used web applications is analysed which is written in weak and strong typed languages. Many typical software faults are behind the majority of web application vulnerabilities, with respect to different programming languages. A realistic attack injector is built. When not all vulnerabilities can be fixed in due time, the data can be used to select those that should be addressed first [5].

Web application and Web application security are most fundamental services of today's digital era. Most of system acquires web application security mechanism most of them successfully uses that but still there is threat of SQL injection attack in web applications. For detecting and preventing many techniques are there but due to vulnerabilities of problem definition, need of new effective SQL injection detection and prevention mechanism is expected. SQL injection from an input SQL query is challenging and most effective mechanism for making web application more robust, as if only SQL query does not produce a different result as compared to expected result will help to improve accuracy and effectiveness of proposed method.

| Method Name | Advantages | Disadvantages |
|---|---|---|
| A novel method for SQL injection attack detection based on removing SQL query attribute values [1] | The proposed method cannot only be implemented on web applications but it can also be used on any applications connected to databases. It can be used for SQL query profiling, SQL query listing and modularization of detection programs. | It is independent of the DBMS when compared with other SQLIA detection methods. |
| Data-mining based SQL injection attack detection using internal query Trees [2] | The proposed method decreases the computation time and increases the probability of correctly detection of SQLIA. | A large amount of time is consumed to generate a multi-dimensional sequence from a query tree. |
| Defeating SQL Injection [3] | Defensive coding practices are labor-intensive. | Runtime prevention approaches require dynamic monitoring systems but it could prevent all attacks. |
| SQL Injection: A Demonstration and Implications for Accounting Students [4] | Identifying and understanding the risks of SQL injection and its impact on financial processes. | Identifying and understanding the risks of SQL injection and its impact on financial processes. |
| Analysis of Field Data on Web Security Vulnerabilities [5] | Applications written in strong typed languages have a smaller number of vulnerabilities and exploits. | Vulnerabilities are not present in the source data analyzed. |

**TABLE 1: COMPARISON BETWEEN SQL INJECTION ATTACK DETECTION TECHNIQUES**

## III. PROPOSED WORK

Many programming languages are currently used to develop web applications. Proprietary languages such as C#, VB and open-source languages such as PHP, CGI, Perl, and Java are also used to develop web application. Programming languages can be classified on parameters such as the programming paradigm, the type system, the execution mode etc. By using

strong typed languages the developed software is more robust software because a value of one type cannot be treated as another type. Given method works primarily to check the SQL queries from given web application and checking it by removing the parameter values. Goal of proposed method is to give a simple and robust approach for SQL injection detecting and checking its originality.

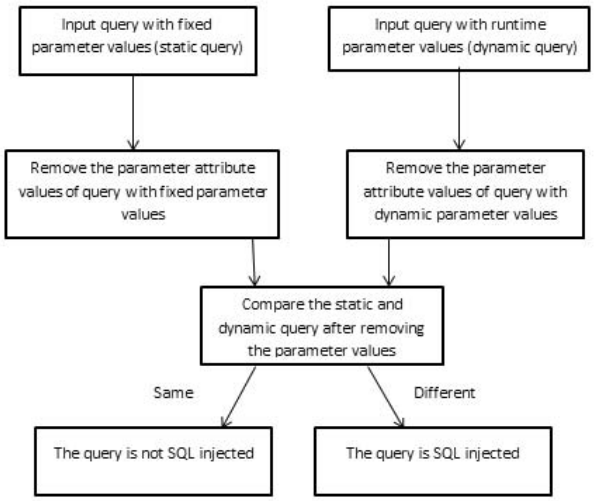


Fig.1: Framework of proposed methodology.

The proposed method detects the injection attacks by using a combined static and dynamic analysis. The parameter values are removed at the dynamic i.e. runtime time from the SQL query and compared with the SQL queries analyzed in fixed or static i.e. in advance. The proposed method is explained with example:

Consider a query:

FXQ: SELECT * FROM STUDENT WHERE RNO='$rollno' AND NAME='$name'; ..... (1)

A function delete () is used to delete the parameter values [1]. The parameter values in fixed SQL queries i.e. static and the SQL queries generated at dynamic time are deleted.

By applying the function delete () to the query FXQ. The result is:

DFXQ=delete (FXQ) = SELECT * FROM STUDENT WHERE RNO= ' ' AND NAME= ' '; …. (2)

At the runtime the query can be in the form,

RTQ1= SELECT * FROM STUDENT WHERE RNO='1001' AND NAME='AJAY'; ….. (3)

By applying the function delete () to the query RTQ1. The result is:

DFXQ1= delete (RTQ1) = SELECT * FROM STUDENT WHERE RNO=' ' AND NAME=' '; …..(4)

Again a query at runtime could be as

RTQ2= SELECT * FROM STUDENT WHERE RNO='1' or'1=1'—'AND NAME='AJAY'; …..(5)

By applying the function delete () to the query RTQ2. The result is:

DFXQ2= delete (RTQ2) = SELECT * FROM STUDENT WHERE RNO=' ' or ' '—' 'AJAY'; ….(6)

When the static and dynamic query after removing the parameter values are compared, if both are same then the query is normal. If there is some difference then the query is not normal, i.e. code injection is present in the query.

## IV. RESULT ANALYSIS

The mechanism is implemented for real web application to detect and prevent the SQL injection attacks. It detects all SQL injection attack types.

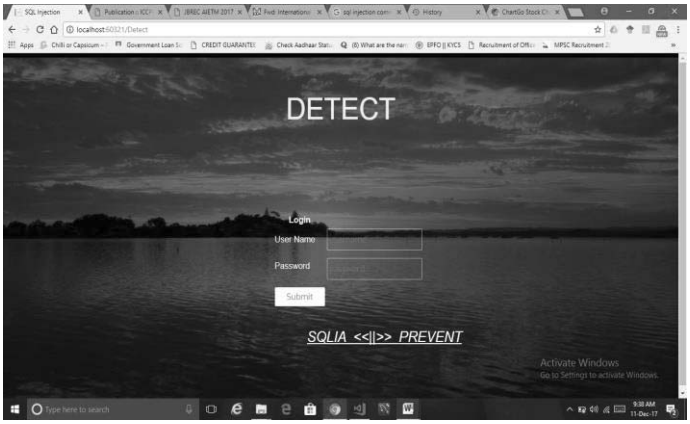


Fig. Snapshot of the detection page

The evaluation for results are done on the basis of Response time, Performance accuracy rate, Computing time, Comparison with combined static and dynamic analysis methods, Detection Rate.

- **Response time:** It is the response time of the system in handling the raised errors. The time to detect the SQL injection is considered for particular attack type.

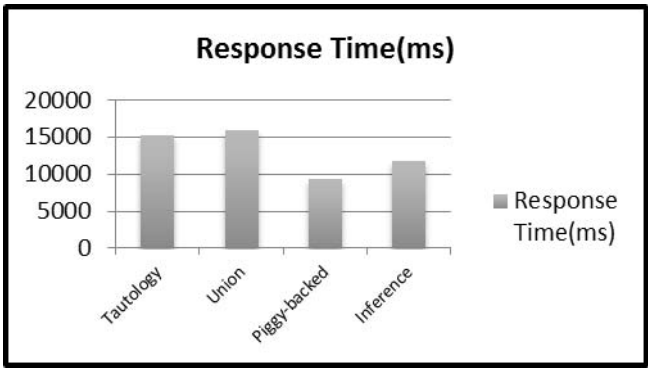


Fig. Detection time of various attack types

- **Performance accuracy rate**: It is the rate to get the accurate result so that the injected code is detected. The detection of number of SQLI represents the accuracy of the SQLI prevention.

- **Computing time:** It is the time required to compute the result of the normal query with respect to the injected query.

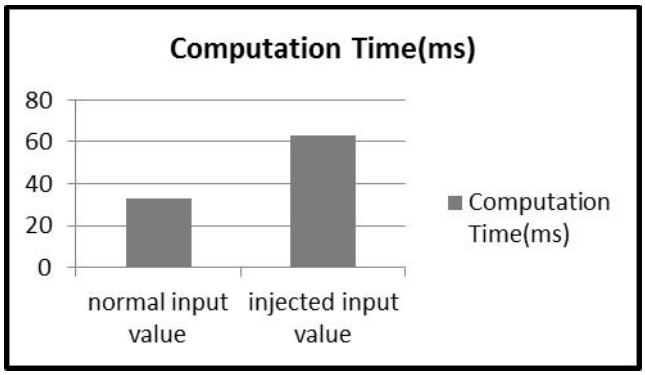


Fig. Computation time for normal query and injected query

- **Comparison with combined static and dynamic analysis methods:** On Comparing different methods which are using combined static and dynamic analysis techniques. The comparison based on parameters such as source code adjustment, static analysis, dynamic analysis, applicable type of web applications and detected type of attacks [31]. In the proposed mechanism the source code adjustment is not needed; static and dynamic analysis is done on SQL queries.

| Detection/ Prevention Method | Source code adjustment | Static analysis | Applicable type of web application | Detected type of attacks |
|---|---|---|---|---|
| AMNESIA | Not needed | String analysis | Runtime monitoring | All but except stored procedures |
| Framework and database firewall method | Not needed | W3af | Database firewall | All |
| Proposed method | Not needed | String analysis | Query string with dynamic parameter. | All |

- **Detection Rate:** The performance of mechanism by issuing malicious attack data and normal input data is measured. These are issued at injection points that are identified at static analysis. A total of 120 queries are issued for each page out of which 45 are malicious and 75 are normal.

| Web page | Attacks detected/ Malicious input | Attacks detected/ Normal input | Detection rate (%) |
|---|---|---|---|
| SQLIA page | 40/45 | 75/75 | 96 |
| Detect Page | 45/45 | 75/75 | 100 |
| Prevent page | 45/45 | 75/75 | 100 |

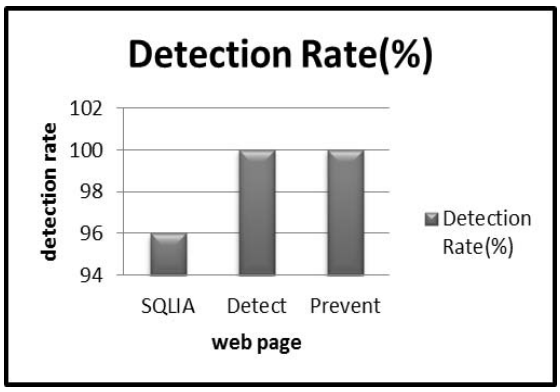


Fig. Analysis of detection rate

## V. CONCLUSION

This paper presents the SQL injection detection mechanism and protecting web applications from SQL injection attacks. Digital era of technology expects attack free systems making more secure sharing of data. Proposed method makes web applications with ability to detect the code injection (SQL injection) attacks before losing any data makes systems more secure. Combining existing SQL injection detection mechanisms to develop more strong mechanism to make web application more robust is best with result as it's expected from this proposed method.

Future work should work on efficient methods for detecting the SQL injection attack and methods to prevent it. A less time must be consumed to detect the SQL injections, for this more new and robust methods are to be developed. The impact on the businesses must be understood to reduce the risk of SQL injection attacks. More research is needed for accurately detection of SQL injection attacks.

## REFERENCES

[1] Inyong Lee, Soonki Jeong, Sangsoo Yeo, Jongsub Moon," A novel method for SQL injection attack detection based on removing SQL query attribute values", Mathematical and Computer Modelling (Elsevier), Volume: 55, Issue: 1-2, PP. 58-68, January 2012.

[2] Mi-Yeon Kim, Dong Hoon Lee," Data-mining based SQL injection attack detection using internal query Trees," Expert Systems with Applications (Elsevier), Vol. 41, Issue 11, PP. 5416–5430, September 2014.

[3] Lwin Khin Shar, Hee Beng Kuan Tan," Defeating SQL Injection," Computer: the flagship publication of the IEEE Computer Society (IEEE), Volume: 46, Issue: 3, PP. 69 - 77, March 2013.

[4] David Henderson, Michael Lapke and Christopher Garcia," SQL Injection: A Demonstration and Implications for Accounting Students," AIS Educator Journal, Vol. 11, Issue 1, PP. 1-8, 2016.

[5] Jose Fonseca, Nuno Seixas, Marco Vieira, and Henrique Madeira," Analysis of Field Data on Web Security Vulnerabilities," IEEE Transactions on Dependable and Secure Computing, Vol. 11, Issue 2, PP. 89-100, March/April 2014.

[6] Anjali Sardana and et. al., " Protecting Web Applications from SQL Injection Attacks by using Framework and Database Firewall", ACM, 2012.

[7] Ashwin Ramesh et. al., "An Authentication Mechanism to Prevent SQL Injection by Syntactic Analysis", IEEE, 2015.

[8] Indrani Balasundarama and et. al., "An Efficient Technique for Detection and Prevention of SQL Injection Attack using ASCII Based String Matching", International Conference on Communication Technology and System Design, 2011.