www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 4 Issue 8 Aug 2015, Page No. 13786-13790

SQL Injection Detection Based On Replacing The SQL Query Parameter Values

R.Latha[1], Dr.E. Ramaraj[2],
[1]M.phil Research Scholar, Department of Computer Science &Engineering,
Alagappa University, Karaikudi
lathasaru10@yahoo.com

[2]Professor, Department of Computer Science &Engineering,
Alagappa University, Karaikudi
eramaraj@rediffmail.com

Abstract: ***Information is converted to digitalized format and then flows through the network medium. Security mechanisms are mostly used to protect information from unauthorized intruders on the network. Secure communication between the medium as well as between the communicating entities is an essential part. There exist many types of attacks in which the SQL Injection is considered for the proposed work. This paper proposed a novel method for the detection and proper replacement to the affected queries. SQL Injection is one of the major attacks which will leaks the valuable information to the intruders. SQL Injection attacks target databases that are accessible through frontend structure of the website, and made flaws in the input validation logic of its components. Therefore, a strong method is needed to overcome the dispute. This paper proposed an efficient method for detecting the SQL injection by manipulating input attributes of the SQL query and measuring the distance of query strings. It satisfies the both query analysis for both the static and dynamic manipulation of user queries.***

Keywords: Information, Security, SQL Injection, Web Components, Databases.

1. Introduction

Now – a- days, most of the services are offered as online services. Therefore, security of the data is remained as the major problem. The ability to access the web anywhere and anytime is a great advantage; however, as the web becomes more popular, web attacks are increasing. Most web attacks target the vulnerabilities of web applications, which have been researched and analyzed at OWASP [16]. A large number of web applications, especially those deployed by companies for e-business operations involve high reliability, efficiency and confidentiality [11]. SQL injection attack or SQL insertion attack is one of the code injection technique which makes use of the vulnerabilities arises in the security occurring in the database. This is most often found within web pages and its content when it is concerned as dynamic. SQL Injection is stronger than other threats in the mean that it does not destroy the records instead they deploy various malicious attacks and prevent the logic from working status. Many researchers have been studying a number of methods to detect and prevent SQL injection attacks, and the most preferred techniques are web framework, static analysis, dynamic analysis, combined static and dynamic analysis, and machine learning techniques [12]. In digital world, every user has to handle the information that is available in various types and format in world wide. Web application is one of the main sources for accessing the data from anywhere and web application is used to retrieve the data from database and render in web pages. The intruders attack the system to steal the information through various type attacking models.

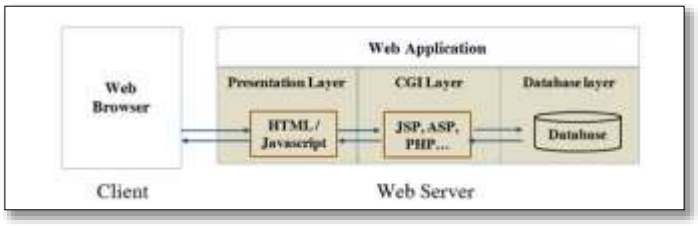

Figure 1: Web Application Architecture

SQL commands are injected from the web form into the database of an application. The Injected SQL commands then change the database content or dump the database information like credit card or passwords to the attacker side. SQL injection is one of the types of attack which drip the information without extra usage of system resources [10]. In this paper we proposed a new method for detecting SQL injection by string replacement algorithm and levenshtein distance method. The rest of the paper is as follows. Section 1 shows the basics of the intrusion and attacks taken place at communication medium, SQL Injection attack. Section 2 consists of the related works to the method. Section 3 proposes the method to detect and replace the SQL Injection in the specific database. Section 4 provides the results available from the experimental analysis. Section 5 brings the conclusion to the proposed method.

2. Related Works

Many researchers have proposed various methods for detecting SQL injection. Developers design the code to find the keywords of the input parameters which presents malicious code. Stephen W. Boyd and Angelos D.Keromytis[1] have proposed method which concatenate the random keywords with legitimate query to avoid the injection. For this method, proxy server is needed and developer knowledge must be

essential. So it's difficult to implement in web application. William Halfond [3] has presented the detection technique based on the concept of depth first traversal of the non deterministic finite automata. To Traces the hot spot and parse the sequence of strings of the SQL query by NDFA. This security model satisfies both static query analysis and dynamic query dynamic query analysis except stored procedure and runtime SQL query generation. Y.-W. Huang [4] have provided a method that satisfy both analysis and dynamic analysis. It performs sanitizing the task and in inserts the sanitizing routine in vulnerable section of code. The pre conditions should be stored for sanitizing to trusted output from tainted input. This method is not generic for all web application. Because the preconditions may change in various architecture design based upon web application. Ke Wei[13] has provided a solution for preventing SQL injection in Stored procedures. This solution satisfies the static and run time validation. Define the query dependencies by graph and stored in table. Check the query dependencies while scanning the query execution. If satisfy the flow graph (DFA). This solution is not efficient because the data of query dependencies should be stored. Angelo ciampa. Et. Al., [14] have applied the heuristic based approach. The authors generate the attacks by crawling the web application and stored the attack pattern in database. The injection detects based on the stored attack patter. This method is not user friendly and does not satisfy the runtime query generation. R. A. McClure et.al., [5] has provided the solution based on DLL. The DLL will views on the SQL Syntax which is eliminate the possibility of SQL syntax, misspelling and data type mismatch problems. The information of each table and query details are incorporated with SQLDOM generation. If database schema changes, then SQLDOM Dll also has to be compiled. This method gives the solution for static analysis alone. William G.J. Halfond and Alessandro Orso[7] have proposed the method, syntax aware evaluation process which will fragment the query and store the keyword and operators. It evaluates the SQL query by arrangement of the syntax sequence. This method is difficult to implement because there exists a need to collect the keywords, operation and sequence of operator which satisfy the SQL syntax. Frank S. Rietta[8] has determined the method for detect the SQL injection in application layer. This detection system breaks the data and watches the frequency of access, compare with historically observed normal data. Efficiency of method is not reliable when it implement in large web application. Gregory T. Buehrer [6] has applied tree concept for parsing the SQL query and compare the tree node of predefined parsed tree length and dynamic query string length which parse at runtime. Turnaround time of this method was increased. Debasish Das[11] has approached dynamic query matching based detection system. Legitimate queries are collect and stored in file in XML format. In runtime, SQL query is converting into XML and check of the predefined structure of legitimate query in master XML file. It doesn't work as dynamically in all web application. Inyong[12] proposed novel method by removing the input attributes of SQL query and measure the length of SQL query. But it only satisfied the string input parameters not fit for numerical input parameter. Y.kosuga[15] has analyzed the semantic and syntax of SQL queries for detecting SQL injection. To Create the repositories for collecting the normal queries and compared it with generated query during the attack. Sruthi Bandhakavi [9] has proposed the technique for detecting SQL injection is to dynamically mine the programmer-intended query structure on any input, and detect attacks by comparing it against the structure of the actual query issued.

3. Proposed Method

3.1 SQL Injection Attacks

Attacks can be made through malicious code at any part of the program. The logic may be changed, the content while as static or dynamic may be intruded or the database content can be modified. Even, if the attacker appends the characters or keywords through input parameters of the web page it will change the logic of the proposed query. One of the most important attack taken place in the database to change the logic is the SQL Injection attack. SQL injection attack or SQL insertion attack is one of the code injection technique which makes use of the vulnerabilities arises in the security occurring in the database. SQL injection is one of the types of attack which drip the information without extra usage of system resources. In the web environment the user request was fulfilled by the input query validation. Hence, the input validation must be checked for the proper response from the server. For example, user enters the value of username and value of password in Login page and submits into web server. In the database layer of the web application, SQL query is generated from value of the input parameters. The generated query passes into database and execute it, then the result send back to the web application. . Let us considered, the input values are "admin" and "admin#123", then the legitimate query dynamically get the structure like "select data from login where userid='admin' and password='admin#123' "and passed into database, retrieve the information and sent back to server page. The server page processes the information and render results in HTML page. Attacker adds the wild card characters or malicious code into legitimate query, as for example

"Select data from login where userid='' or 1 =1 – and password=''

In above query, syntactical structure remains correct whereas the logical remains mistaken. The attacker changes the meaning of query by injecting the malicious code and conciliation it.

3.2 Recent Attacks and Survey

- On March 27, 2011 mysql.com, the official homepage for MySQL, was Compromised by TinKode using SQL blind injection.
- In August, 2011, Hacker Steals User Records From Nokia Developer Site.
- In September, 2011, Turkish Hackers accessed NetNames DNS records and changed entries redirecting users.
- In October, 2011, Malaysian Hacker, Phiber Optik managed to extract data from www.canon.com.cn by SQL Injection.
- SQL injection has been responsible for 83-percent of successful hacking-related.

3.3 Proposed Method

SQL Injection is one of the major threats regarding the users. In this paper, we propose a method that detects the occurrence of the SQL vulnerability injection and replaces it with proper

alternative and satisfies the security of the web application. The algorithm works in the following procedure. First step is to replace the special character instead of using the input parameters as the SQL query. There exists two categories of special characters namely the string and the numerical (STR, NUM). These two constraints will replace the original string constraints in the place of input parameters as fed by the users. The proposed method can be experiment with web applications. There are two type of query execution method has involved, such as dynamic query analysis and static query analysis. Initially, replace the special character instead of input parameters of the SQL query. The Proposed work is used for the detection and prevention against following various type of attacks. Tautologies, Piggyback queries, Union queries, Untyped parameters, Stored Procedures, Inference Alternate encodings. We have proposed a technique for preventing the SQL injection attacks in web applications. Our technique satisfy both static and dynamic analysis. SQL injection detection based on replacing special string constraints instead of SQL query attribute values. Following symbols are used in this method

- PQ → Predefined Query
- GQ → Generated Query with user input
- STR → special string constraint of string input parameter
- NUM → special string constraint of numeric input parameter
- RPQ → Restructured Predefined Query
- RGQ → Restructured Generated query with user input

The user after entering the input value it is then replaced with the STR or NUM as per the input parameters. The distance of the input query was computed every time. The detection method explains with example in given below.

PQ= " select * from user account where userid="'+usrid+"' and password="'+passwd+"' "

In PQ, there are two input parametes such as usrid and passwd. Here PQ is a legitimate query for accessing the useraccount table permitted users only. The proposed algorithm will replace the special string constraint instead of every input parameter. Before replacement entire SQL query rewrite into string format include input parameters. The following structure of SQL query string assigns to replace the special string constraint

PQ="Select * from user account where userid='+usrid+' and password='+apsswd+' " ;

After replacement process, the following restructured query is obtained

RPQ= Select * from user account where userid=STR and password=STR;

The algorithm then checks for the string and numeric constraint in the user given query and replaces it with the query provided. For the next process, we take generated SQL query with user input (GQ), replacing special string constraint which is covered with single quotes instead of input attributes values of the SQL query(GQ).

GQ= " select * from user account where userid='admin1' and password='tututu' "

Input attribute value of Userid is 'admin1' and password value is "tututu". The algorithm which replace the special string constraint instead of input attribute values. The following statements are obtained after replace.

GQ= " select * from user account where userid='admin1' and password='tututu' "

RGQ=" select * from user account where userid=STR and password=STR

Finally, we get two restructured query in given below

RPQ= Select * from user account where userid=STR and password=STR:

RGQ=" select * from user account where userid=STR and password=STR

Finally, we have to applied the levenstein method, to measure the distance between the two strings.

3.4 Levenshtein Method

A novel method for the SQL injection by replacing it with Query parameter values was implemented and the experimental results are analyzed and compared with the existing available techniques. The web application taken for the execution is the "Classifieds" application, that contains nearly 400 records as test samples. The outcome of out proposed method yields better results for the application.

The following table 1 shows the detection of various SQL injection attacks, and their detection methods with different parameters such as tautology, illegal queries, piggy backed queries, stored procedure etc., The Detection and prevention methods in the table are taken from various references. The proposed Method is capable of accepting the dynamic queries and efficiently replaces the detected Injected Queries with the proper attribute values. The table 2 shows the analysis of varied elements for each prevention method. From the table 2 it is clear that, the proposed method is automatically replaces the injected queries without the source code adjustment.

4. Results and Discussions

A novel method for the SQL injection by replacing it with Query parameter values was implemented and the experimental results are analyzed and compared with the existing available techniques. The web application taken for the execution is the "Classifieds" application, that contains nearly 400 records as test samples. The outcome of out proposed method yields better results for the application.

The following table 1 shows the detection of various SQL injection attacks, and their detection methods with different parameters such as tautology, illegal queries, piggy backed queries, stored procedure etc., The Detection and prevention methods in the table are taken from various references. The proposed Method is capable of accepting the dynamic queries and efficiently replaces the detected Injected Queries with the proper attribute values. The table 2 shows the analysis of varied elements for each prevention method. From the table 2 it is clear that, the proposed method is automatically replaces the injected queries without the source code adjustment.

DOI: 10.18535/ijecs/v4i8.29

Table 1 : Detection of SQL Queries against various attacks and their outcomes

Table 2 : Comparative Analysis for varied Detection and Prevention Methods

Detection Prevention Method	*Tautologies*	*Incorrect Queries*	*Union Queries*	*Piggy Backed Queries*	*Stored Procedures*	*Inference*	*Alternate Encoding*	*Static Queries*	*Dynamic Queries*
Amnesia	Possible	Possible	Possible	Possible	Not Possible	Possible	Possible	Not Possible	Not Possible
SQL Guard	Possible	Possible	Possible	Possible	Not Possible	Possible	Possible	Not Possible	Not Possible
SQLrand	Possible	Not Possible	Possible	Possible	Not Possible	Possible	Not Possible	Not Possible	Not Possible
SQLDom	Possible	Possible	Possible	Possible	Not Possible	Possible	Possible	Not Possible	Not Possible
WebSSARI	Possible	Possible	Possible	Possible	Possible	Possible	Possible	Possible	Not Possible
Existing Method	Possible	Possible	Possible	Possible	Possible	Possible	Possible	Not Possible	Partially Applicable
Proposed Method	Possible	Possible	Possible	Possible	Possible	Possible	Possible	Possible	Possible

Detection Method	*Source Code Adjustment*	*Attack Detection*	*Attack Prevention*	*Elements Added Additionally*
Amnesia	No Need	Automatic	Automatic	Not Available
SQL Guard	Desirable	Not Available	Automatic	Not Available
SQLrand	Desirable	Automatic	Automatic	Not Available
SQLDom	Desirable	Automatic	Automatic	Not Available
WebSSARI	No Need	Automatic	Automatic	Not Available
Existing Method	No Need	Automatic	Automatic	Not Available
Proposed Method	No Need	Automatic	Automatic	Not Available

5. CONCLUSION

The proposed method can be experimented with classified web application. There are two types of query execution method namely dynamic query analysis and static query analysis. The proposed algorithms can also applicable for real web applications. Collection of compiled web applications are citied in online that are also used in many research works. The proposed algorithm used citied web application for implementation purpose. The proposed algorithm remains easy to implement and takes less turnaround time. The main advantage of the proposed algorithm is that it does not require

any proxy server for executing detection mechanism. SQL Injection detection was done for both the string and numerical constraints than the existing algorithms that works only for the strings. We have presented a technique for preventing the SQL injection attacks in web applications. It can prevent from more than 1500 attacks that were implemented on the considered application without indication of false positive results. It was quite efficient and reliable. In our future work, we will investigate the technique for tracing the attacks in data packets by scanning the port level. Designing a component without dependencies of web application environment that plug in database software to secure the application against SQL injection attack was also to be found.

References

[1] S. W. Boyd and A. D. Keromytis. SQLrand: Preventing SQLinjection attacks. In Proceedings of the 2nd Applied Cryptography and Network Security (ACNS) Conference, pages 292–302, June 2004.

[2] C. Anley Advanced SQL Injection In SQL Server Applications. Next Generation Security Software Ltd. White Paper, 2002.

[3] William G.J. Halfond and Alessandro Orso ,AMNESIA: Analysis and Monitoring for NEutralizing SQL-Injection Attacks, ACM,ASE'05, November 7–11, 2005, Long Beach, California, USA

[4] Y.-W. Huang, F. Yu, C. Hang, C.-H. Tsai, D. T. Lee, and S.-Y. Kuo. Securing Web Application Code by Static Analysis and Runtime Protection. In Proceedings of the 12th International World Wide Web Conference (WWW 2004), May 2004.

[5] R. McClure and I. Kr¨ger. SQL DOM: Compile Time Checking of Dynamic SQL Statements. in Proceedings of the 27th International Conference on Software Engineering (ICSE 2005), pages 88–96, 2005.

[6] Gregory T. Buehrer, Bruce W. Weide, and Paolo A. G. Sivilotti," Using parse Tree Validation to Prevent SQL InjectionAttacks" ACM September 2005

[7] William G.J. Halfond, Alessandro Orso, and Panagiotis Manolios,"Using Positive Tainting and Syntax-Aware Evaluation to Counter SQL Injection Attacks", ACM,SIGSOFT'06/FSE-14, November 5–11, 2006,

[8] Frank S. Rietta,"Application Layer Intrusion Detection for SQL Injection",ACM SE'06 March 10-12, Melbourne, Florida, USA,2006.

[9] Sruthi Bandhakavi Prithvi Bisht P. Madhusudan V. N. Venkatakrishnan,"CANDID: Preventing SQL Injection Attacks using Dynamic Candidate Evaluations" , ACM, November 2007.

[10] Romil rawat, Chandrapal singh dangi, Jagdish patil, "safe guards anomalies against SQL injection attacks", international journal of computer applications(0975-887),May 2011.

[11] Debasish Das ; Utpal Sharma ; D.K. Bhattacharyya,"An Approach to Detection of SQL Injection Attack Based on Dynamic Query Matching", International Journal of Computer Applications ,ISSN:09758887,2010

[12] Inyong,Lee Soonki,Jeong Sangsoo YeoJongsub,Moon**,"**A Novel Method for SQL Injection attack detection based on removing SQL query attribute values", Elsevier, Mathematical and computer modelling, Jan-2011,ISSN:0895-7177,

[13] Ke Wei, M. Muthuprasanna, Suraj Kothari, "Preventing SQL Injection Attacks in Stored Procedures",IEEE,Software Engineering Conference, 2006. Australian, ISSN: 1530-0803.

[14] Angelio ciampia,corrado Aaron visaggio, massimiliano di penta, "A heuristic-based approach for detecting SQL-injection vulnerabilities in Web applications",SESS10,ACM,2010.

[15] Y,kosuga,K.kernel,M,Hanaoka,M,Hishiyama,Y.Takaham a,Sania: "Syntactic and Semantic analysis for automated testing against SQL injection", proceedings of the computer security application conference,pp-107-117,2007.

[16] The Open Web Application Security Project, OWASP TOP 10 Project. http://www.owasp.org/.

[17] J. Park, B. Noh, SQL injection attack detection: profiling of web application parameter using the sequence pairwise alignment, in: Information Security Applications, in: LNCS, vol. 4298, 2007, pp. 74–82.