International Journal of Engineering & Technology, 7 (2.31) (2018) 16-18

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Vulnerability detection and prevention of SQL injection

B.J. Santhosh Kumar[1*], P.P. Anaswara[2]

[1]Department of Computer Science, Amrita Vishwa Vidyapeetham, Amrita School of Arts and Sciences, Mysore, Karnataka.
[2]Department of Computer Science, Amrita Vishwa Vidyapeetham, Amrita School of Arts and Sciences, Mysore, Karnataka.
**Corresponding author E-mail:santhoshbj50@gmail.com*

Abstract

SQL injection attack is the most serious security vulnerabilities on databases are connected with web or within an intranet, most of these vulnerabilities are affected by lack of input validation and SQL parameters are use. The attackers are trying to steal the data which was hidden and by attacking the database using the attacking technique that is called SQL injection attacks. The SQL injection attack detection and prevention technologies are experimented in this paper. There are different defence methods are used to prevent such as, parameterized statement, stored procedures and white list input validation. The comparative results of these methods are highlighted in the table with SQL injection query, prepared statement insertion and selection queries, stored procedures and modify queries. The comparison of these methods used for detection and prevention vulnerability in web server.

Keywords: *SQL injection attack, SQL queries.*

1. Introduction

Softwares are spreading everywhere throughout the world and having the difficulties as security issues. Web applications are seeing commonplace among the people now a days, a portion of the web applications are net keeping money, web mail, online sales, online deals retails, interpersonal organizations and websites are the natural one. Web vulnerabilities have made substantial scale development in web applications where the web engineers comes up short to write programming code. It is important to play out a legitimate sentence structure approval and to take after the security lead to secure for anticipation and amid the programming stage.

"Numerous business and open source instruments exist in advertise with particular highlights despite the fact that analysts have dissected and demonstrated not even a solitary recognition scanner gives best outcome to every one of the classifications of powerlessness. It is profoundly challengeable undertaking for security-situated designers to assemble dependable apparatuses that give simpler way to deal with handle the security issues. Weakness identification scanners are exceedingly exceptional, utilized frequently among substantial associations as they not distinguish potential vulnerability[1]".

SQL infusion assault is a code infusion assault and a simplest procedure, by utilizing SQL summons, for example, Select, Where, Insert, Delete and Update, the aggressors outline the SQL proclamations and executes defenseless code into the web applications. There are enormous measure of security issues on the web application, that can be taken care of by validation of clients and there are numerous types of SQL infusion assaults exist.

2. SQL injection methods to prevent SQLIAs

To prevent the databases from the intruders using the SQL queries are injecting and preventing the security issues. To avoid SQL injection flaws is simple and easier. There are three methods using to prevent such as

- Method 1: Use of Prepared Statements (with Parameterized Queries)
- Method 2: Use of Stored Procedures
- Method 3: White List Input Validation

Method 1: Use of Prepared Statements (with Parameterized Queries)

Database programmer and database end users (naïve user) used to write database different queries to get result for performing task. Both make use of simple and dynamic queries to perform tasks. Prepared statements and parameterized queries insist the developers to define SQL code and pass as a parameter and query it. Framed statement ensure that an attacker is not able to change the intend of a query. For example, attacker want to enter the user_id of name or '1'='1 the query is vulnerable and will look for user_name which matched the string.

Method 2: Stored Procedures

Stored procedures are also similar methods of SQL injection making use of parameterized queries. Developer has to build SQL statements with parameters for performing SQL injection. Stored procedure is defined and stored in the database further call from the application. Both the techniques are efficient in preventing SQL injection.

Method 3: White List Input Validation

Input validation using white list using SQL queries, return to a name of tables or columns. Input validation is the appropriate design for names of tables or columns and those values received from the code not from the user input. If user inputs are used to make a different for table and column names then input values should be mapped to expected tables or column names.

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

3. Related work

"Kanchana Natarajan et al [1] proposed an SQL-injection free secure algorithm to detect and prevent SQL injection attacks. Implementation done through java and algorithm describes the method that how they follow the procedures for preventing SQL-injection attacks. Showing the comparison of similar types of attacks with the features and the result is the evaluation proves the algorithm works efficiently to detect the SQLIAs. Voitovych O.P et al[2] proposed a known approach to protect Web applications against SQL injection attacks in the article. To improve the Web software security it is developed defence mechanism that protects Web resources from SQL injection performing. To implement this software it is used PHP, JavaScript and formal language theory known as regular expression. As a result it is received a software tool which allows to protect Web software from SQL injection vulnerability. Developed software tool allows user to protect his own Web application from an attack with using SQL. Jose Fonseca et al[3] proposed a method and prototype tool evaluate web application security mechanisms. The method is injecting the realistic vulnerabilities in a web application and attacking automatically can be used to support the rating of existing security mechanisms. In this paper explaining the implementation of the Vulnerability and Attack Injector Tool(VAIT) that is used for the authentication of the entire process. Used the tool to run a set of experiments that demonstrate the feasibility and the effectiveness of the proposed methods. The experiments include the evaluation of coverage and false positives of an intrusion detection system for SQL injection attacks. The result shows that the injection of vulnerabilities and attacks is indeed an effective way to evaluate security mechanisms to point out not only the weakness but also their improvement. Joel Brynielsson et al [4] introduced one such weakness found within version 2.2 of the popular Apache HTTP Server software. The weakness concerns how the server handles the persistent connection feature in HTTP 1.1. An attack simulator exploiting this weakness has been developed and shown to be effective. The attack was then studied with spectral analysis for the purpose of examining how well the attack could be detected. The results show that disproportionate amounts of energy in the lower frequencies can be detected when the attack is present. However, by randomizing the attack pattern, an attacker can efficiently reduce this disproportion to a degree where it might be impossible to correctly identify an attack in a real world scenario. Li Qian, Zhenyuan Zhu et al [5] introduced typical SQL injection attack and prevention technologies. The detecting methods not only validate user input, but also use type-safe SQL parameters. SQL injection defence model is established according to the detection processes, which is effective against SQL injection vulnerabilities. Mohammad Qbea'h et al [6] presented a formal approach to detect and prevent common types of SQLIA considering multi-languages. Formalize tautology and alternative encoding attacks using regular expressions and finite automata and provide regular expressions and code in ASP.net which can be used by developers to detect and prevent attacks on websites that use Microsoft SQL server 2014(MS-SQL). Validate the work manually and by using tools and results show that model can detect and prevent SQL injection attacks including languages other than the English language. Zoran Djuric [7] focused to develop a reliable black-box vulnerability scanner for detecting SQLI vulnerability SQLIVDT (SQL Injection Vulnerability Detection Tool). The black-box approach is based on simulation of SQLI attacks against web applications. Thus, the scope of analysis is limited to HTTP responses and HTML pages received from the application server. In order to achieve efficient SQLI vulnerability detection, an efficient algorithm for HTML page similarity detection is used. The proposed tool showed promising results as compared to six well-known web application scanners. Zainab S. Alwan et al [8]presented classical and modern types of SQLIA and display different existing technique and tools which are used to detect or prevent these attacks that is surveyed the most popular existing attack issues, which is SQLIA. Also we have presented a survey report on classical and modern types of SQLIA, their working methods, and detection and prevention techniques against classical and modern types of that attack. For evaluation, we compare the detection and prevention techniques in terms of their ability to detect the attack, or prevent the attack or partially stop the attack. Regarding the results, the efficiency of some techniques should be improved to overcome the SQLIA.B. Deva Priyaa et al [9] proposed a hybrid framework using the EDADT (Efficient Data Adaptive Decision Tree)algorithm which is the semi–supervised algorithm and SVM classification algorithm. It uses the internal query tree from the database log for good performance of framework. To get internal query tree, the query tree is converted to n–dimensional feature vector by using multidimensional sequence. The semantic features are used as the component of feature vector. And also the syntactic and semantic feature is used to generate multi – dimensional sequences. Then the extracted feature is converted into numeric value, if the feature contains any string value. Experimental results show that the proposed approach is more accurate in detecting the attacks than existing approaches. Geogiana Buja et al [10] proposed a detection model for detecting and recognizing the web vulnerability which is; SQL Injection based on the defined and identified criteria. In addition, the proposed detection model will be able to generate a report regarding the vulnerability level of the web application. As the consequence, the proposed detection model should be able to decrease the possibility of the SQL Injection attack that can be launch onto the web application. B.R.Pushpa [11] proposed the basics of network security and types of symmetric algorithms and new approach for symmetric encryption algorithm for encryption and decryption of data and the merits of the new proposed algorithm. S. Joseph et al [12] presented a study of the popular SQL Injection Attack (SQLIA) techniques and the effectiveness of conventional fixes in reducing them. For addressing the SQLIA's in depth, a thorough background study was done and the mitigation techniques were evaluated using both automated testing. The results indicate the importance of incorporating these mitigation techniques in the code apart from going for complex fixes that require both effort and time".

4. Experimental result

Fig. 1: Result using SQL injection

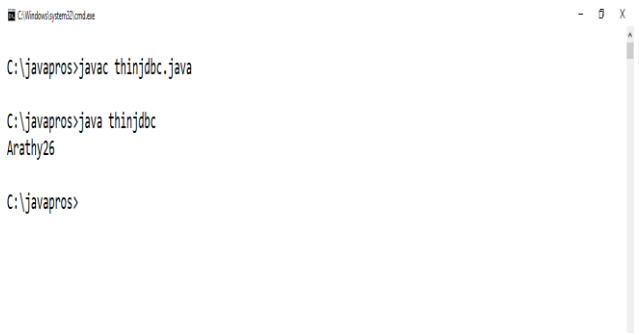

Fig. 2: Result without SQL injection

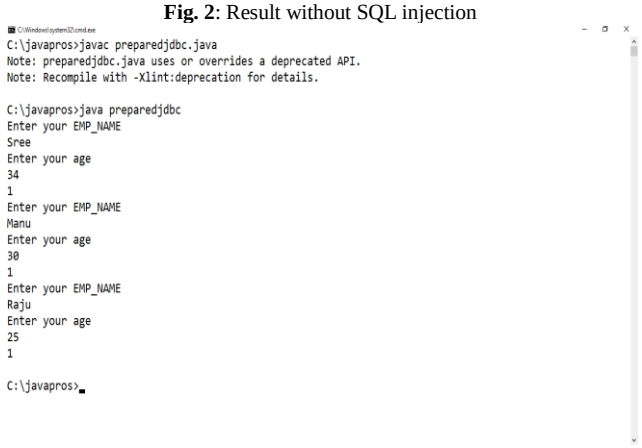

Fig. 3: Prepared statements for insertion

In Fig 3: the database query with prepared statement is not exposed to the user and he has to pass only the values for insert, delete or update the data. In prepared statement the values are assigned to int parameters which is represented as(?). The values are mapped to the int parameter so that the intruder cannot see the data type of the values or problems.

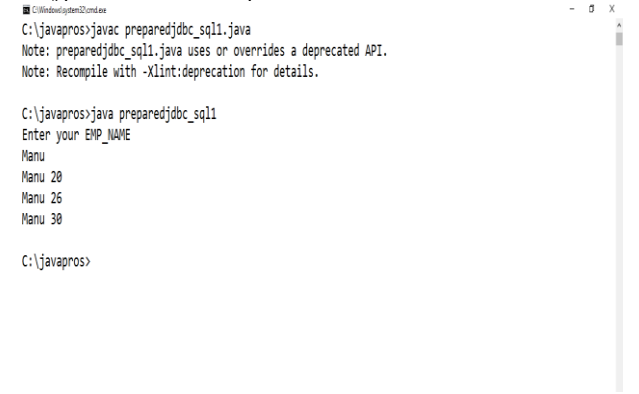

Fig. 4: Prepared statements for selection

In Fig 4: the callable statements are return and stored by the database administrator and this procedure are not exposed to the user. We can write jdbc callable statements to call the stored procedure to perform insert, delete or update queries. This completely prevents the database access to intruder.

Table 1: Comparison of detection and prevention methods

	Normal	Intruders
SQL injection query	Can access rows in table	Can access rows and all tables
Prepared Statement	Can access one row	Can access one row at a time
Stored procedures	Can insert, delete, update values in the table	Database admin can revolve execution
White list input validation	A special query has to be passed	A special query has to be passed

5. Conclusion

In this paper used three methods for the vulnerability detection and prevention of SQL injection. The three prevention methods using are prepared statement, stored procedures and white list input validation. The simple queries are used in the methods to prevent the SQL injection attacks. The comparisons of detection and prevention of methods is shown as Table 1.

Acknowledgment

First of all we would like to express our sincere thanks to Mata Amritanandamayi Devi (Amma) for her inspiration and guidance both in unseen and unconcealed ways. Wholeheartedly, we would like to express our sincere thanks to Br.Sunil Dharmapal, Director and Br. Venugopal, Correspondent, Mysuru campus for providing studying environment with excellent infrastructure and continuous encouragement for carrying out my dissertation work at Amrita Vishwa Vidyapeetham University Mysuru campus. We would like to express our sincere thanks to beloved principal Prof. Vidya Pai C for giving us moral support and continuous encouragement.

References

[1] Natarajan K & Subramani S, "Generation of SQL-injection free secure algorithm to detect and prevent SQL-injection attacks", *Procedia Technology 4 Elsevier Ltd*, (2012), pp.790–796

[2] Voitovych OP, Yuvkovetskyi OS & Kupershtein LM, "SQL Injection prevention system", *International Conference Radio Electronics & Info Communications (UkrMiCo)*, (2016), pp.1-4.

[3] Fonseca J, Vieira M & Madeira H, "Evaluation of web security mechanisms using vulnerability & attack injection", *IEEE Transactions on Dependable and Secure Computing*, Vol.11, No.5,(2014), pp.440-453.

[4] Brynielsson J & Sharma R, "Detectability of low-rate HTTP server DoS attacks using spectral analysis", *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, (2015), pp.954-961.

[5] Qian L, Zhu Z, Hu J & Liu S, "Research of SQL injection attack and prevention technology", *International Conference on Estimation, Detection and Information Fusion (ICEDIF)*, (2015), pp.303-306.

[6] Qbea'h M, Alshraideh M & Sabri KE, "Detecting and preventing SQL injection attacks: a formal approach", *Cybersecurity and Cyberforensics Conference (CCC)*, (2016), pp.123-129.

[7] Djuric Z, "A black-box testing tool for detecting SQL injection vulnerabilities", *Second International Conference on Informatics and Applications (ICIA)*, (2013), pp.216-221.

[8] Alwan ZS & Younis MF, "Detection and Prevention of SQL Injection Attack: A Survey", *International Journal of Computer Science and Mobile Computing*, Vol.6, No.8, (2017), pp.5–17.

[9] Priyaa BD & Devi MI, "Hybrid SQL injection detection system", *3rd International Conference on Advanced Computing and Communication Systems (ICACCS)*, (2016), pp.1-5.

[10] Buja G, Jalil KBA, Ali FBHM & Rahman TFA, "Detection model for SQL injection attack: An approach for preventing a web application from the SQL injection attack", *IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE)*, (2014), pp.60-64.

[11] Pushpa BR, "Enhancing Data Security by Adapting Network Security and Cryptographic Paradigms", *International Journal of Computer Science and Information Technologies*, Vol.5, (2014), pp.1319–1321.

[12] Joseph S & Jevitha KP, "Evaluating the Effectiveness of Conventional Fixes for SQL Injection Vulnerability", *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics: ICACNI*, (2016).