

Detection and Prevention of SQL Injection

Shravan Singh, Ankit Kumar

Dept. of Computer Science & Engineering
Galgotias University, Greater Noida -201308
Shravans821@gmail.com, ankitkumar3110@gmail.com

Abstract – Penetration testing is widely accustomed to audit the protection of Web applications. This can be an awfully important phase for an online based company/organization since all of their data is stored in their databases and people databases are linked to the online server they own to host their website. More the new functionalities are being introduced day by day, the less secure the system is. Web application penetration testing is introduced for testing the net application for vulnerabilities and countering those using web application security mechanisms. As per current situation, the penetration testers use manual testing for examining the applying and maintaining reports, they face many problems finding vulnerabilities on web applications like they need to go looking on the web for vulnerabilities which may be present within the testing web app. An automated web application penetration testing can help them in some ways like from storing all of the vulnerabilities within the database and using those databases at the time of examination of the online application.

Keywords– SQL query, Web application, Mo- bile application, Desktop application, Dynamic method, SQL- injection free algorithm, Runtime environment, SQL-injection attack

I. INTRODUCTION

Web applications are widespread today, as they have become the requirement of way of life. There are thousands of security violations that occur daily. The web application is vulnerable because of loopholes, and these loopholes come at the time of designing an application or after we update something in our application.

The most common attack on the online is SQL (Structured Query Language) injection. The Classic SQL injections were easy to forestall and detect and lots of procedures, method- ologies were discussed to beat SQL injections. The assorted methodologies went to overcome the attack is writing secure code in keeping with an intensive investigation of Howard and his team that's associated with the defensive code writing with correct validation using coding and decoding techniques.

The concept of attacks by injection is to insert or inject a malicious code into a program to switch SQL query structure. Such an attack is done by adding malicious character strings within the data values within the style of arguments values within the URL. Injection attacks generally make the most of incorrect validation about input / output data. SQL injection attack or SQLIA (Structured search language Injection Attack) may be a style of code injection attacks consisting of malicious injection SQL commands through client computer file to application that's passed to the instance of the database for its execution and with the target of affecting the execution of predefined SQL commands.

The following script is executed on an online server. It's an easy example of authentication bypassing a username and a password. The instance database incorporates a table named users with the subsequent columns: username and password.

II. OVERVIEW OF SQL INJECTION

SQL injection attack (injection) is the commonest and easiest form of vulnerability technique adopted by the on- line attackers through data-driven web applications. By using simple SQL commands like Select, Where, Insert, Delete and Update, the malicious attackers efficiently re-structure the particular SQL code (statements) and execute vulnerable code into the net applications. Once nasty attacker attains their goal they will easily access sensitive information, modify secured data, execute the info, and even they will collapse the complete application. Since the privacy of the database administrator loses their role by unauthorized accesses of malicious.

SQL injection attacks are more lucrative for at- tackers as they mainly specialise in stolen bank accounts, mastercard numbers, etc. this kind of security issue on web applications is more susceptible, and might be handled by the authentication of users. Many sorts of SQL injection attacks exist. most typical takes the advantages of erroneously passed parameters, erroneous type handling, erroneous use of SQL statements, for e.g. (' OR) 1 = - '). Various sorts of SQL injection attacks are available like tautologies, illegal/logically incorrect queries, UNION

query, Piggy-backed queries, Stored Procedures, Blind SQL, Timing Attack, Alternate Encoding and etc. Defeating these forms of attacks isn't simple since the attacker actually changes the behaviour of predefined SQL queries.

III. TECHNIQUE TO DETECT AND STOP SQL INJECTION:

Many research authors explored a variety of methods to detect and forestall SQLIAs; the foremost chosen techniques are static analysis, dynamic analysis, combined static and dynamic analysis, web framework, defensive programming and machine learning techniques. The strategy of static analysis is extreme where it analyzes the code for vulnerability by without actually executing the code. Software metrics and reverse engineering are some sorts of static analysis.

Model checking, data flow analysis, abstract interpretation and use of assertions in ASCII text files are the several techniques of static code analysis. The strategy of dynamic analysis is performed automatically by the analysis of vulnerabilities during the execution of web applications which avoids thousands of tests by doing several times manually. Example: CANDID tool. Both the techniques have merits and demerits and so variations are identified from the efficacy. However the research study analyzed various existing works and it's been proved that dynamic analysis (penetration testing) tool is effective to check the net applications [1, 4, 8]. Penetration testing tools are easy to use and assure to supply security information systems to their users by fixing the safety weaknesses before they get exposed. The key advantages of penetration (dynamic) testing are: (a) Not necessary to vary the event lifecycle (b) Avoids static analysis challenges (c) No need for the ASCII text file, (d) Deployment-security.

The method of combined static and dynamic analysis can compensate the constraints of every method, which is taken into account as highly proficient against SQLIAs but it's very complicated. one of the simplest examples for such a way is the AMNESIA tool. It uses static analysis to investigate the web-application code and automatically build a model of the legitimate queries that the applying can generate. At run-time, the technique monitors all dynamically-generated queries and checks them for compliance with the statically-generated model. When the technique detects a question that violates the model, it classifies the query as an attack, prevents it from accessing the database, and logs the attack information. the net framework method could be a filtering method of user input parameters. This method is proven to be ineffective while it's unable to filter some special characters. The machine-learning method is the most typically used method whereas the strategy ends up in high false positives and low detection rate.

IV. SQL-INJECTION FREE SECURE ALGORITHM

The newly proposed algorithm relies on dynamic technique which violates SQL injection attacks. This algorithm concentrated to develop IF (Injection Free) attacks; whereas a special sort of test suite is developed to detect SQL injection attacks. Systematic flowchart of this proposed approach is given below-

The below shown Algorithm performs its function by assigning the shape Status (FS) as attack and free with collection of fields obtained from the gathering of forms. the shape (fm') is obtained from the set of forms (Fm) whereas values of each field are obtained from the shape (Fm'). Three well-defined functions are generated inside the tactic called CheckVulnerability (f') to test for any special characters, keywords and Boolean characters.

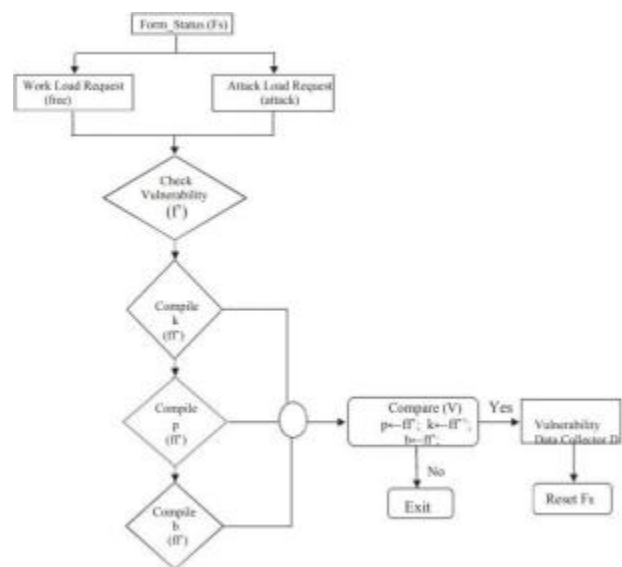


Fig. 1. Fig1: Systematic flowchart of the model shows SQLi Free Secure

V. ALGORITHM FOR SQLI FREE SECURE

Algorithm SQLIAD (Form F)

Input: Fm denotes the gathering of forms with collection of Fields.

Input: Enumerate Form_Status FS = Input: Default Value to FS as free Output: FS for every fm' summation of Fm do for every f' summation of fm' do f' κ fields contains the values if f' isn't empty string the FS κ output from the tactic CheckVulnerability (f') if FS as attack D κ Add the sector f' within the collection Reset the Http requests to issue warning; return FS;

Generic detection method 1: Generated to test for Special characters, keywords and Boolean keywords

Table 1 The comparison results are shown in.

Comparison of detection and prevention methods with various features and attack types

Detection/Prevention Methods	Static Analyzer	Dynamic Analyzer	Detailed Output Info	Illegal Queries	Piggy-backed Queries	Stored procedures	Union Queries	Real Time
PSR - Algorithm [1]	✓	x	x	x	x	x	x	x
Novel Method [5]	✓	✓	x	x	x	x	x	✓
Safe Query Objects [5]	✓	✓	x	✓	✓	x	✓	x
Technology-checker [5]	✓	x	x	x	x	x	x	x
Web App Hardening [5]	x	x	x	✓	✓	x	x	x
Proposed SQL-IF	x	✓	✓	✓	x	x	✓	✓

VII. CONCLUSION AND FUTURE WORK

The proposed algorithm is considerable in inspection of its simple detection operation against SQL injection attacks. Testing of web applications, mobile application and desktop application for SQL injection attack may be a significant step for ensuring its performance and quality. The proposed algorithm performs much faster and endowed with a proficient solution to resolve against SQL injection attacks. The paper-work has been analyzed with various detection methods and therefore the proposed method cannot only be implemented on web applications and can also be used on any applications which interact towards databases. The future research is going to be considered to construct SQL parsers. Generation of parsers to detect critical vulnerabilities is another one-plex approach. Also dynamic checking compilers are often designed to harden the online applications in three-tier internet services for shielding from SQL Injection attacks (SQLIAs). Both the approaches were quite feasible to attain effectiveness and efficiency.

REFERENCES

- [1]. Shaukat Ali, Azhar Rauf, Huma Javed, SQLIPA: An Authentication Mechanism against SQL Injection, European Journal of Scientific Research, 2009, Vol.38, pages: 604-611
- [2]. MeiJunjin, An approach for SQL injection vulnerability detection, IEEE Sixth International Conference on Information Technology: New Generations, pages: 1411-1414, 2009
- [3]. J. Park, B. Noh, SQL injection attack detection: profiling of web application parameter using the sequence pair-wise alignment, Journal of Information Security Applications, LNCS, 2007, vol. 4298, pages: 74-82.
- [4]. W.G.J. Halfond, A. Orso, P. Manolios, WASP: protecting web applications using positive tainting and syntax-aware evaluation, IEEE Transactions on Software Engineering, 2008, vol. 34 (1), pages: 65-81.
- [5]. T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer, "Social phishing," Communications of the ACM, vol. 50, no. 10, pp. 94-100, 2007.
- [6]. Y. Kosuga, K. Kernel, M. Hanaoka, M. Hishiyama, and
- [7]. Y. Takahama, "Sania: syntactic and semantic analysis for au- tomated testing against SQL injection," in Proc. the Computer Security Applications Conference , 2007, pp. 107-117.
- [8]. Inyong Lee, Soonki Jeong, Sangsoo Yeo, Jongsub Moon, A novel method for SQL injection attack detection based on removing SQL query attribute values, Journal of Mathematical and Computer Modeling, Elsevier Ltd, 2011, pages: 1-
- [9]. Wichman, "Mass sql injection for malware distribution," SANS Institute, Tech Rep , 2010
- [10]. Joa~o Antunes, Nuno Neves, Miguel Correia, Paulo Verissimo, and Rui Neves, Vulnerability Discovery with At- tack Injection, IEEE Transactions on Software Engineering, 2010, Vol. 36, pages: 357-370.
- [11]. Splaine, S. (2002). Testing Web Security: Assessing the Security of Web Sites and Applications. John Wiley & Sons.
- [12]. Stampar, "Data retrieval over dns in sql injection attacks," arXiv preprint arXiv:1303.3047, 2013
- [13]. Dimitris Mitropoulos, Diomidis Spinellis, SDriver: Location-specific signatures prevent SQL injection attacks, Journal of Computers & Security, Elsevier Ltd, 2009, pages: 121-129.
- [14]. W3af.org, "w3af - Open Source Web Application Security Scanner", 2015. [Online]. Available: <http://w3af.org/>. [Accessed: 29- Dec- 2015].
- [15]. Stephen Thomas, Laurie Williams, Tao Xie, On auto- mated prepared statement generation to remove SQL injection vulnerabilities, Journal of Information and Software Technol- ogy, Elsevier Ltd, 2009, pages: 589-598.
- [16]. Lijiu Zhang, Qing Gu, Shushen Peng, Xiang Chen, Haigang Zhao, Daoxu Chen, "D-WAV: A Web Application Vulnerabilities Detection Tool Using Characteristics of Web Forms", IEEE Fifth International Conference on Software Engineering Advances, pages: 501-507, 2010.