International Journal of Advance Research in Science and Engineering
Vol. No.5, Issue No. 01, January 2016
www.ijarse.com
IJARSE
ISSN 2319 - 8354

SECURING WEB APPLICATION BY SQL INJECTION DETECTION TOOLS: A SURVEY

Gorakhnath[1], Mrs. Meenu[2]

[1,2]Department of Computer Science and Engineering,

Madan Mohan Malaviya University of Technology, Gorakhpur, Uttar Pradesh, (India)

ABSTRACT

SQL injection is an attack by which attacker targets the data in database by the help of structured query language code. It can get access to the database due to poor input validation in code. SQL injection allows an attacker to get access directly to web application and destroy the functionality. It is done by manipulating the SQL statement query. Attacker take benefits of web application security fault and pass malicious SQL statement code in this may attacker get full access to the database. Researchers have proposed different tool to detect and isolate this problem. In this paper we present SQL injection attacks and current tools that prevent this attack.

Keywords: SQL Injection Attack, Prevention , Detection, Tools, Authentication.

I. INTRODUCTION

Internet is basic necessities in our modern era it perform its basic roll in every field such as learning business and in social network life also. Today's people use internet as hacking system and disturbs the entire internet networking system that make very noise to whole internet world. That's why we adopt the network security system. The main focus of network security is to protect networks from any attack basically from hackers that exploit our networks. In each organization the network security is commonly controlled by network administration that applies different security policies. Its implementation is necessary to protect the network and its resource from unauthorized user so in this regards three main goals of security is integrity, availability and confidentiality.[3]

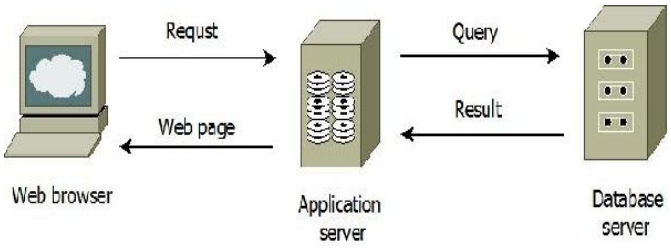

Fig-1: Architecture of web application

The most ordinary way to collect over the internet is World Wide Web (WWW). In internet system the hyper text transfer protocol (HTTP) is used to transmit data in web service its main application to communicate with each other for sharing business document and data. Each of web document and web pages can be accessed by using web browser. Web pages contain videos, text, sound, images and multimedia application in which they are attach to each other by the hyperlinks. URL is one of the important components of hyperlinks. URL is unique on

every website that makes a process of searching easier. Today web applications use a multi-tier design; there are three types of tiers presentation, processing and a data tier. HTTP web interface are in presentation tier. Application tier implement the functionality of software and data tier keep data structure and answer to the request by the application tier. [11] SQL injection is type of attack by which intruder or attacker adds SQL code to the make some changes in the database.

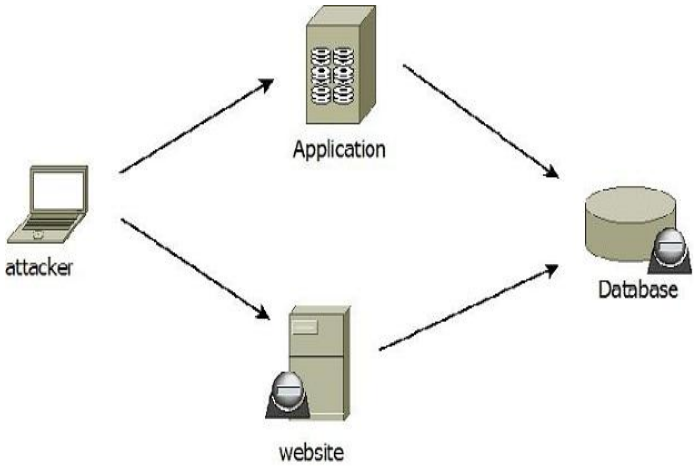

Fig-2: SQL Injection Representation

II. TYPE OF SQL INJECTION

The information from a database can be read by using three main forms of SQL injection.

1. Redirection and reshaping a query
2. Error message based
3. Blind injection

2.1 Redirection and reshaping a query

The user inputs can become an injected statement into the pre-written SQL statement. [12]The intruder can enter data into database repository, and then Webpages are changed to send user or client to another webpage. By redirecting and reshaping the query it can be done by the help of tautology, end of line comment, union query, piggy backed query and system stored procedure.

2.2 Error message based

An attacker used this technique to gain access the information gathering stage of this attack, which is considered primary method to collect information.[9,10] An attacker may collect knowledge that adds the attack by injecting logically false request like finding out injectable parameters, and table's names. The difficulty level in this attack is that application server returned default error page. The error messages generated are revealed as injectable parameter to intruder when attackers performing this attack then the tries to inject statement code that causes syntax, logical error and type conversion into database. Syntax errors can be used to identify injectable parameter. For e.g.

SELECT student FROM university WHERE login='' AND pass ='' AND pin= convert (int, (select top 1 name from sysobjects where xtype='u'))

2.3. Blind injection

In case of incorrect queries programmers believes in hiding database information to protect the information from attacks.[8] That's why the hackers fail to collect any clues about that particular database the objective use to compromise security by checking its result with a lot of true/false database queries. So the hackers attempt illegal access in response of these queries. As timing attack is one of the most prominently used to identify the behavior of the application. The queries result forms in Boolean value and present output of hyper text markup language (HTML) page. It provided by blind SQL injection technique in database.

III. TYPE OF SQL INJECTION ATTACK

3.1 Tautologies

This type of attack injects SQL tokens to evaluate conditional query statement that is true.[5,11] In this type of attack bypass authentication control is used and access is given to data by exploiting vulnerable input field which use WHERE clause.

"SELECT * FROM university WHERE userid='115' and password='gorakh' OR '1'='1'''

As the tautology statement (1=1) has been added to query statement so it's always true.

3.2 Union Query

In this technique, intruder or attackers join a safe query that is injected by the help of UNION and then it can get data from the application.[7] Suppose for our examples that query executed from the server as following.

SELECT Name, Phone FROM Users WHERE Id= $id

By injecting the Id value as follows:

$id=1 UNION ALL SELECT credit Card Number, 1 FROM Credit sys Table

We have the following query:

SELECT Name, Phone FROM Users WHERE Id=1 UNION ALL SELECT creditcardNumber, 1 FROM Credit sys Table

This will join the result with original query for all the credit card users.

3.3 Stored Procedure

It is part of database programmer that set an extra abstraction layer on the database [12,13]. Stored procedure is a part of inject able as web application forms it is depend on specific procedure of the database attacker disrupt stored parameter procedure.

CREATE PROCEDURE DBO. Is Authenticated

@user Name varchar2, @pass varchar2, @pin int

AS EXEC ("SELECT accounts FROM users

WHERE login="' +@user Name+ "' and pass='"

+@password+

"' and pin=" +@pin);

Go

For authorized and unauthorized user the stored procedure returns true and false. As an SQLIA, intruder input "';

SHUTDOWN; - -" from username or password. Then the stored procedure generate the query as follows:

SELECT accounts FROM users WHERE login='doc' AND pass=' '; SHUTDOWN; - - AND pin=

Therefore this type of attack commonly behaves as a piggy - back attack. The query is executed by one as first and then second query which is illegitimate execution that cause the shutdown of database.

3.4 Piggy- Backed Queries

In piggy backed queries attack, attacker exploit database by query delimiter code. Such as ";", to add some extra query to the original query. Databases receive with successful attack and execute multiple numbers of queries. [1,2]The first query is correct query where as following all may be incorrect query. By this attacker can insert a query into database. In the following example, attacker inject " 0 ; drop table user" into the pin input field instead of logical value, an the application would produce the query:

SELECT info FROM users WHERE login='doe' AND pin=0; drop table users.

In this ";" all the queries are accepted and executed. After that second query is incorrect that's why it is drop from the database table. Some databases do not need special character separation in multiple numbers of queries so detection is not an impressive solution for an attacker.

IV. TECHNIQUES FOR DETECTION AND PREVENTION OF SQL INJECTION ATTACK

To prevent the SQL injection attack, prevention is concerns with concerns with correctness of input value that is given by user or client at coding level. By the help of these code technique client want to enter with correct data, incorrect data is based which is harmful in the database.

4.1 AMNESIA Approach: SQL injection attacks are detected by AMNESIA technique these help in both static approach and runtime monitoring. It detects the query which is executed into database by the help of model based approach. [5,] This approach gives two ways static part and dynamic part static part builds legal queries. By the help of pregame analysis and dynamic part generates queries dynamically against static build queries using runtime monitoring system. The technique has four steps for preventing injection identify the hotspot, build SQL query models, Instrument application.

4.2 CANDID(Dynamic Candidate Evaluations) Approach- design a program transformation which modifies web application that is written in java it compare against the structure of query issue and detects attacks of intended query structure input CANDID's simple, natural approach and very powerful for detection of SQL injection attacks[5,6].

4.3 VIPER for Detecting SQL injection Attacks- this technique used heuristic based approach for SQL injection attacks detection. It used performs penetration testing for web application.[4,11] This technique used for analyzing of web application for input supplied and determining hyperlinks structure and give error message for user input, if any SQL injection attacks occur.

4.4 Stored procedure approach- this technique is suggested by Suraj Kothari, Ke Wei, and M, Muthuprasanna. Subroutines are used in stored procedures which help web application to interact with data base server and performing computation on the database.[11] Identification of SQL injection attacks through combination of static analysis and dynamic analysis. Command verification are used in static analysis which used by runtime validation and subroutines parser using SQLCHAKER for input validation.

4.5 WAVES approach- this technique developed by hung and colleagues which used a black-box testing technique for testing web application for SQL injection modification [9]. This tool identify all points of web application that may be used for inject SQLIAs.These points and monitors of the web application, how response to the attacks through machine learning utilization.

V. RELATED WORK

The protection of web application from SQL injection attack can be done by two ways. Firstly, there is much need of mechanism to detect and isolate the SQL injection attack by identifying the attack in the database.[3] Secondly, SQL injection knowledge is must for securing the web application. So for many platform or frameworks have been used. Any suggested to identify the SQLIV in web application. Here some prominent solution are working methods are described.

5.1 William G.J.Halfond - work shows the combination of static analysis and runtime monitoring. In the static part program analysis technique is used to build a model of the legitimate queries that is generated by the application. In the dynamic part programmers monitors the generated queries at runtime and check them for statically generated model. Queries that reject the model represent SQLIA and prevented from executing from database repository.

5.2 Safeli – proposes a framework of static analysis in order to detect the SQL injection attacks. Its aim to identifying the SQL injection during the runtime Two advantage of static analysis tool firstly, it does a white-box static analysis and secondly, it uses hybrid constraints. In the white box analysis the proposed approach considers the byte-code and deals with strings. And for hybrid static analysis the method implement an efficient way to deal with Boolean integer and string variables.

5.3 Thomas- has proposed that automated prepared generation algorithm remove SQL injection threats or vulnerabilities. He parsed for open source projects namely, (i) Net-trust, (ii) I Trust, (iii) Web Goat, (iv) Roller, by these 94% of SQLIV statement code was replaced.

5.4 Rues approach- automatic test case generation proposed to detect the SQL injection vulnerabilities. The idea behind these frameworks is based on specific model that automatically deals with SQL queries. These approaches identify the relationship between queries and sub queries. These methodologies show that 85% and 70% reduction of threat by experimenting few examples.

5.5 Ali et al.'s scheme- he proposed for hash value approach to improve the mechanism of user authentication. In this they use user value and password in SQLIPA (structure query language injection protector for authentication).this help in order to test the framework. The user name and password hash value are calculated and created at runtime for the first time when the user account is created.

5.6 Buehrer approach- He adopted the parse tree framework. He compared the parse tree of particular statement to its original statement at runtime. This method was tested by using SQL guard on student web application. It is efficient best two drawback are there first is additional overhead computation and testing of input.

VI. CONCLUSION

In this paper we presented various type of SQLIA. Here we have investigated SQL detection and prevention tool. It is known that SQL injection attack is one of the largest classes of security problems. By detecting the problem in SQLIA, the current tools were compound that is based on modifying source code, additional infrastructure and automation of detection or prevention. In the future work we will propose a framework for messing effectiveness, efficiency and performance of tool in order to prove the strength and weakness of SQL query. This will help in evaluating the different web application that is available in public domain. We are also exploring the possibility of implementing the middleware of database engine, to avoid the explicit behavior of source code.

REFERENCES

[1] G. Buja and T. F. Abdul, "Detection Model for SQL Injection Attack : An Approach for Preventing a Web Application from the SQL Injection Attack," pp. 60–64, 2014.

[2] Y. Fouad and K. Elshazly, "Detection and Prevention of SQL Injection Attacks on Web Applications," vol. 13, no. 8, pp. 1–7, 2013.

[3] P. Y. Jane and M. S. Chaudhari, "SQLIA : Detection And Prevention Techniques : A Survey," pp. 56–60.

[4] D. Kar and S. Panigrahi, "Prevention of SQL Injection attack using query transformation and hashing," Adv. Comput. Conf. (IACC), 2013 IEEE 3rd Int., pp. 1317–1323, 2013.

[5] P. Kumar and R. K. Pateriya, "A Survey on SQL Injection Attacks , Detection and Prevention Techniques," no. July, 2012.

[6] O. Lounis, S. Eddine, B. Guermeche, L. Saoudi, and S. E. Benaicha, "A new algorithm for detecting SQL injection attack in Web application," pp. 589–594, 2014.

[7] A. Pramod, A. Ghosh, A. Mohan, M. Shrivastava, and R. Shettar, "SQLI Detection System for a safer Web Application," pp. 237–240, 2015.

[8] C. Sharma, "Analysis and Classification of SQL Injection Vulnerabilities and Attacks on Web Applications," 2014.

[9] A. K. Singh and S. Roy, "A Network Based Vulnerability Scanner for Detecting SQLI Attacks in Web Applications," 2012.

[10] T. Surya Fajri, E. D. Meutia, and E. Mustafa, "Analysis of SQL injection attack in web service (a case

study of website in Aceh province)," Instrumentation, Commun. Inf. Technol. Biomed. Eng. (ICICI-BME), 2013 3rd Int. Conf., pp. 431–435, 2013.

[11] A. Tajpour, S. Ibrahim, and M. Sharifi, "Web Application Security by SQL Injection DetectionTools," vol. 9, no. 2, pp. 332–339, 2012.

[12] K. Wei and M. Muthuprasanna, "Preventing SQL injection attacks in stored procedures," Aust. Softw. Eng. Conf., p. 8 pp.–198, 2006.

[13] K. Wei, M. Muthuprasanna, and S. Kothari, "Preventing SQL Injection Attacks in Stored Procedures," 2006.