International Journal of Innovative Research in Advanced Engineering (IJIRAE)
Volume 1 Issue 3 (May 2014) SPECIAL ISSUE

An Approach for Prevention of SQL Injection Attacks on Database: A Review

Prerna U. Randive
Computer Dept, Pune University
prernarandive311@gmail.com

Mahadev B. Khatke
IT Dept, Pune University
k.mahadev777@gmail.com

Malu B. Reddi
Computer Dept, Pune University
malu.reddi@rediffmail.com

Abstract— now days most of the work is done by web application and it consisting of databases that contain important information as well as sensitive information in the form of data. Data can be retrieve and manipulate from database that helps to organisms to extract information. All this operations is completed by the data management system. Nearly each web application is evolved in such a way that it required some data is completed via user input. So the attacker can take advantage of poor validation given to the input and administration. SQL injection attacks occurred when attacker succeeds in inserting the malicious code via query and that query is treated like input. Which can give attacker unauthorised access to use databases. Attacker gives input for execution of back-end database and SQL injection attacks happened.

Keywords— Database, malicious code, SQL injection, Unauthorised access, Attacker

I. INTRODUCTION

Web applications are very useful in making of online transactions, online banking, online shopping, reading news paper, email etc. Web application plays an important role in today's life. Security of web application is major challenge for web developers. Data or information is the important factor in organisation, business and industries in now a day environment. So security of information is needed to trace up to high level. Many applications totally based on databases, which gives services to the customers. By trusting this systems customers uses this web application to store the confidential data. Databases are core of many organisations for this reason attacker wish to target database because of this attacks are increasing day by day. One of the top most attacks is SQL injection attack which targets this poor web applications. It is just like hacking technique. This hacking technique gives unauthorised access to database by giving input which consists of malicious code included into the query. Further that query is treated like input. Attacker has different intensions for attacks. Firstly attacker wants to identify inject-able and weak parameter to attack. By the attacks he can modify data, change data and extract data. He can lick the confidential data from the storage of database. Attacker also wants to know about the database schema which consists of the number of rows and columns, name of table, columns data types, column name from this collection of information he can make use of all to inject information system. In some cases attackers can gain the control over the system and act as host by the unauthorised access by this authorisation it can modify the database. SQL injection attack is major issue and very serious so the anticipation of SQL injection attack is major challenge in day today life.

II. BACKGROUND

The top most issue occurred in database is the SQL injection attack. SQLIA is nothing but the class of code used as user input as query. This can crash system by corrupting the whole information present in the database. With the malicious intension attacker choose some web applications which have poor security. Attacker just not do this to gain unauthorised access over database, also to make changes in database, extract information, modification of database schema and information present in it.

There are some SQL injection attacks as follows,

A. Tautology Attacks

In the tautology attacks, these attacks are a simple attack that is why mostly used by attacker. Attacker make use of one or more conditional statements by injecting SQL tokens so that it always evaluated to true, it shows 1. In this type of attack, attacker used to bypass authentication pages and access to pages that consists data for execution purpose. In this type of injection, attacker exploits the fields, such field that can be inject-able that is content of query's where clause.
" SELECT name FROM bank WHERE userid='prerna' or 1=1 AND pswrd='12345' AND pin=' ' "
In this example code injected at the WHERE clause and retrieve results because the WHERE clause is always true. If attacker succeed in it then all or some records of database table will returned. This is well known attack for attacker and simplest attack among the all attacks.

B. Union Queries

By using tautology, attacker can succeed in bypassing authentication pages, but does not give flexibility to attacker to extract information. So this type of injection used for achieve this goal. This technique is used by attacker to trick the application into returning data from the database.

In this type of attack, attacker makes use of vulnerable parameter to make injected query and then join this injected query to the original query with the word UNION. Then it can retrieve data from database.

" SELECT name FROM bank WHERE userid=' ' UNION SELECT balance FROM employee WHERE empid='123' -- AND pswrd=' ' AND pin= "

In this query, the injected query will returned balance from employee table, whereas original query return the null set. Database takes the results of these two queries i.e injected query and the original query and then returns them to application. Final result will be the union of both results.

C. Piggybacked Queries

Piggybacked queries attack is one of the harmful attacks. In this attack, attacker does not modify the original query. Same like the UION query, attacker appends more additional injected queries to the original query. That is piggybacked on the original query; by this, database retrieves more number of SQL queries. The first query is original legitimate query which executes normally and the other are subsequent queries are injected queries. Attacker can inject virtually any type of SQL command to the database.

" SELECT name FROM bank WHERE userid='prerna' AND pswrd=' '; drop table Loans – AND pin='123' "

Database treats this query string as two queries separated by the delimiter (';') and it executes both query. The first query is original one and the second query is injected query. The injected query drops the loan table in database. So the information including in that table get deleted.

D. Logically Incorrect Queries

This attack allows attacker to gather the important information about the database type and structure of the back-end databases of a web application. This attack is useful to make other attack by the gathering information; it is one of the steps for the other attacks. In this attack, attacker tries to inject the statements that help to cause syntax or logical error into the database. When the query gets rejected that time error message is generated and returned from database with including important information. This error message helps to attacker to find out the inject-able parameter in application and gain information about the schema of back-end database.

"SELECT name FROM bank WHERE userid='prerna' AND pass='123' AND pin= convert (int,(select top 1 name from sysobjects where xtype='u')) "

In this query attacker attempt to extract table (xtype='u') is made by injected SQL query. Then this query uses for conversion of table name string value in the integer value. This is an illegal type conversion so the database shows and returned an error message. From this query attacker may succeed to know the name of table which is used for storage of information.

E. Inference

This attack is used to find out information about database schema. From this attack, query gets created that cause application of database to behave differently on the basis of results of query. There are two attacks techniques based on inference attack i.e blind injection and timing attacks.

In blind injection attacks, sometimes web application developer hides the error message details which can help to attacker to find out information of database. Attacker intentionally put injected string in query to generate error but instead of error message attacker face to generic page which is provided by developer instead of the error message details. So this type of injection attack is difficult in such case but can be inject successfully. Attacker asks series of questions simultaneously for getting the indirect response from database. It has results in the form of Boolean result.

In timing attack, attacker get information by the time delays in database responses. From this attacker gathered information. This attack uses the if-else statement to create the injected queries and uses the WAITFOR keyword with one of the branch that will cause delay in database response by specified time.

F. Alternate Encoding

Many types of attacks use the special characters as input to the web application. Hence there are some prevention techniques which block such inputs. Alternate encoding allows attacker to modify their injected strings in such a way that they can avoid typical signature based and filter based checks. So attacker makes use of encoding such as ASCII, Hexadecimal and Unicode can be used in conjunction with other techniques to allow an attack to escape from detection approach that simply scan the query for certain known "bad characters".

III. RELATED WORK

There are some techniques used to identify and prevent from input manipulations vulnerabilities. From these techniques web application program can improve them to reduce input vulnerability. Some techniques are described below.

A. SAFELI

A Static Analysis Framework for Detecting SQL Injection Vulnerabilities [12] construct and outline the design of SAFELI, intended to identify the SQL injection attacks at compile time, not at run time. This static analysis framework has two advantages i.e white-box static analysis and hybrid-constraint solver.

In the white-box static analysis SAFELI analyse byte code and generate user inputs as hard-evidence of vulnerability. In the hybrid-constraint solver involve Boolean, integer and string operations.

B. Pattern Matching Algorithm

Dr.M.Amutha Prabakar, M.KarthiKeyan and Prof.K. Marimuthu [6] proposes pattern matching algorithm. This algorithm is used for the identification and prevention of SQL injection attack using Aho-Corasic pattern matching algorithm. Evaluation of this algorithm is done by well known attack pattern. This pattern matching step takes O(n) time. This proposed scheme consists of two modules i.e static phase and dynamic phase. In static phase user generated query are checked by using static pattern matching algorithm. In dynamic phase, if any form of new anomaly is detected then alarm will indicate and new anomaly pattern will be generated.

C. SQLRand Scheme

SQLrand: Preventing SQL Injection Attacks [8] proposed by StephenW. Boyd and Angelos D. Keromytis. SQLRand is used for give practical protection mechanism against the SQL injection attack. By this it achieve portability and security gain. It uses randomized SQL query language to special CGI application from this it is possible to find out the injected code. Care must be taken by the CGI implementer itself to avoid exposing randomized query.

D. Combinatorial Approach

Preventing SQL Injection Attack Using Combinatorial Approach [9] is proposed by Dimple D. Raikar, Sharada Kulkarni and Padma Dandannavar. Combinatorial approach is used to prevent web application from the SQL injection attack. The proposed system implemented for the java-based web application. WASP is written in java used to prevent SQL injection. WASP is get modified for the combinatorial approach, it is enhancement tool. This approach consisting of the detection of vulnerable spot from application, trusted data source and allow data entering from trusted sources. It has practical advantage like it uses syntax-aware evaluation which shows security problem. And has the practical advantage over different techniques such as whose application required customized and complex runtime environments.

E. Negative Tainting Approach

A.S. Gadgikar have suggested negative tainting approach[1] is used to preventing SQL attacks without modification of existing code and reduce time and space complexity. Negative tainting approach has advantage like it does not require any costly hardware and can work with any type of database. This approach consist of identifying the vulnerable spot from web application, by using negative tainting algorithm detect the SQL injection attacks, then lastly inserting the newly identified SQL injection attacks to improve accuracy of the system.

F. AMNESIA

In [3], MeiJunjin suggested that AMNESIA is the detection tool. This modifies SQL query model generated by AMNESIA to trace the user input that reaches a SQL query and to create injected attack input for the test cases. And test oracle is used to test the execution passed or failed. JCrasher generates the test cases input for string type values and SQLInjectionGen for detection of hotspots. Advantage is that automatic test case generation for the detection of SQL injection vulnerabilities also helps to programmer to find the vulnerable location. Disadvantage for this approach is that it includes number of steps by different tool.

G. Positive Tainting and Syntax-Aware Evaluation

The use of positive tainting and syntax-aware evaluation has been proposed by William G.J. Halfond, Alessandro Orso and Panagiotis Manolios [2]. They proposed highly automated approach against SQL injection that has both conceptual and practical advantages over most existing approaches. In the conceptual advantage the approach is based on positive tainting and concept of syntax-aware evaluation. In the practical advantages their technique is at same time precise and efficient. And impose minimal deployment requirement. They used WASP tool to implement their technique that used to stop all attacks without generating false positive. These prevention techniques identify the trusted data source and allowing only data from such trusted source to become SQL keyword or the operator in the query string.

IV. CONCLUSIONS

From the above description it concludes that database security is major issue in day today life. Database is primary step towards the many applications. Today most of the applications are purely based on the database. So database is backbone for most of the applications and security of database is major issue. SQL injection attack is harmful attack for the database. Database is favorite target for attacker because database is containing sensitive and important information. In this paper I have presented review on different types of SQL injection attacks and also the different detection and prevention techniques for the web application database against the SQL injection attacks.

ACKNOWLEDGMENT

The presented paper would not have been possible without college AISSMCOE, Pune. I got Support from friends and family. I am thankful to the teachers who suggesting me, which help me in improving my work, from this I learnt many new things.

REFERENCES

[1] A.S. Gadgikar, "Preventing SQL Injection Attacks Using Negative Tainting Approach," 978-1-4799-1597-2/13/$31.00 ©2013 IEEE
[2] William G.J. Halfond, Alessandro Orso and Panagiotis Manolios," WASP: Protecting Web Applications Using Positive Tainting and Syntax-Aware Evaluation," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 34, NO. 1, JANUARY/FEBRUARY 2008
[3] MeiJunjin," An approach for SQL injection vulnerability detection," Huangshi Institute of Technololgy, 2009 Sixth International Conference on Information Technology
[4] Nikita Patel, Prof. Shivshakti Shrivastava and Prof Hitesh Gupta, "An Approach of Preventing Code Injection Attack in Web Environment," IJARCCE, *Vol. 1, Issue 5, July 2012*
[5] Pushkar Y.Jane and M.S.Chaudhari, " SQLIA: Detection And Prevention Techniques: A Survey," *(IOSR-JCE) ISSN: 2278-0661, ISBN: 2278-8727, PP: 56-60*
[6] Dr.M.Amutha Prabakar, M.KarthiKeyan and Prof.K. Marimuthu "AN EFFICIENT TECHNIQUE FOR PREVENTING SQL INJECTION ATTACK USING PATTERN MATCHING ALGORITHM," 978-1-4673-5036-5/13/$31.00 © 2013 IEEE, (ICECCN 2013)
[7] Shelly Rohilla and Pradeep Kumar Mittal, " Database Security by Preventing SQL Injection Attacks in Stored Procedures", *IJARCSSE* , Volume 3, Issue 11, November 2013
[8] StephenW. Boyd and Angelos D. Keromytis, "SQLrand: Preventing SQL Injection Attacks", Department of Computer Science University
[9] Dimple D. Raikar, Sharada Kulkarni and Padma Dandannavar, "Preventing SQL Injection Attacks Using Combinatorial Approach", *(IJARCET)Volume 1, Issue 8, October 2012*
[10] Mr. Saurabh Kulkarni1 and Dr. Siddhaling Urolagin, " Review of Attacks on Databases and Database Security Techniques," ISSN 2250-2459, Volume 2, Issue 11, November 2012
[11] Asha. N ,M. Varun Kumar and Vaidhyanathan.G , "Preventing SQL Injection Attacks," IJCA(0975 – 8887) Volume 52– No.13, August 2012
[12] Xiang Fu, Xin Lu, Boris Peltsverger, Shijun Chen, Kai Qian, Lixin Tao, "A Static Analysis Framework For Detecting SQL Injection Vulnerabilities", (COMPSAC 2007)0-7695-2870-8/07 2007 IEEE Transaction of computer