ISSN:0975-9646

SQL Injection Attack on Data and Prevention through Hashing

Shanu Verma[#], Urvashi [*]

[#]*Computer Science, Lingaya's University*
Faridabad, Haryana, INDIA

[*]*Assistant Professor, School of CSE*
Faridabad, Haryana, INDIA

***Abstract*— SQL Injection Attack (SQLIA) is a method with help of attackers attack the data directly into the database in an unofficial way and accomplish the maximum important information for remove and modifying information from any corporation. In this paper, we examine the state of the separate types of attacks with explanation and illustration of how attacks of that type can be performed and describe their detection and prevention system .It is also describe the strength and weakness of SQL injection attack .It is well-known to all that SQL injection attacks can be easily prevented by applying more secure schemes in login phase and after login phase. Hence, we accomplish our proposed scheme through SQLENCP, the SQL injection prevention by cryptography using hashing technique, to handle the SQLIA and prevent them. Even though, the planned implemented system is unable to handle all the SQL injection attacks, but it can avoid tautology attacks, union based query attacks & illegal structured query attacks.**

***Keywords*— SQL injections, SQL injection attacks, SQL attacks, database attacks, Cryptography.**

I. INTRODUCTION

A SQL injection attack can be made of insertion or "injection" of a SQL query with the input data from the user to the application. A successful SQL injection can read sensitive data from the database, change database data and perform query such as Insert/Update/Delete and perform administration operations on the database such as shutdown the Database, recover the data present in a file on the Database and perform some commands on the operating system. A SQL injection attacks is one of the most common injection attack on data, in which SQL query are injected into data and get all the information regarding that data. SQL injection attacks permit attackers to take off identity and change it and create issues such as cancel the transaction and update the balance, allow the complete operation on data such as destroy the data or make it unavailable from the other people , and become administrators of the database server. SQL injection attack occurs when data enters into a program from unauthorized organization. The main reason of SQL injection attack are confidentiality, authentication, authorization and integrity. In confidentiality sql query contain sensitive data and authentication such as username and password. In sql injection attack has become a common issue. There are four main kind of SQL Injection attacks against Oracle databases.

1. SQL Manipulation
2. Code Injection
3. Function Call Injection
4. Buffer Overflows

The Sql Manipulation include change the sql statement with operation such as union, intersect and change the clause of where to return different results. The Code injection attack where an attacker insert new sql statement and execute those statement. Function call injection is the insertion of oracle database function into sql query and these function is used to make operating system calls and modify data in the database.The buffer overflows is a part of function call injection.

II. SQL INJECTION

SQL injection is a method in which various users can insert SQL query into an SQL statements.Inserted SQL query can change SQL statement and compromises the security of a web application[1]. SQL Injection attacks could happen when a fornt application use client-supplied data without proper confirmation and encoding as part of a query. SQL Injection permit an attacker to create, read, modify, change, or delete data stored in the back-end database. In its simplest form, SQL Injection permit an attackers to retrieve important information such as social security numbers, credit card number or other financial information

Key significance of SQL Injection

- SQL injection is a software problem that happen when data entered by users is sent to the SQL interpreter as a part of an SQL statement

Attackers offer especially important input data to the SQL interpreter and shark the interpreter to execute unintended query

III. SQL INJECTIONS METHODOLOGIES

An application of database is used in almost each and every field to capture their information and keeping their record also. There are several method with help of which we can prevent sql injection attack such as tautology, Illegal/logical incorrect query and union based query

Tautology -The aim of a tautology-based attack is based on to insert code in one or more conditional query so that they always evaluate to true. The significance of this attack depend on how the results of the query are used within the application. The most common applications are to bypass authentication pages and extract data. In this type of injection, an attacker exploits an inject-able field that issued in queries WHERE conditional.

In this example SQL injection is based on 1=1 is always true in which an attacker

Give [‟ or 1=1 - -] for the user name input .It is usually used with double slash - - to origin the rest of a statement to be ignored and make sure that large amount of data to be extracted

The resulting query is:

SELECT status, user_name FROM table_user WHERE user_name = ’’ or 1=1 – AND pwd= ‘ ‘

Let’s take another example in which

User_id:104 or 1=1

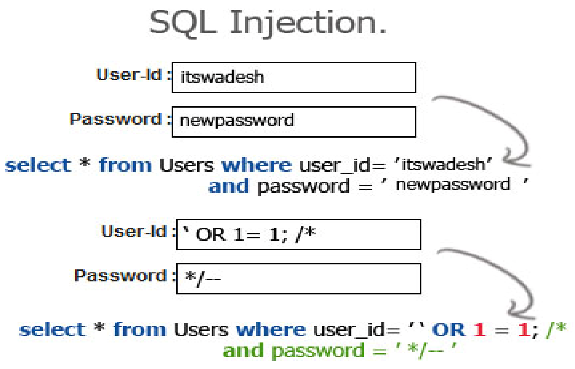
Server Result: SELECT * FROM Users WHERE User_Id = 104 or 1=1

The SQL above is valid. The above query will give all rows from the table Users, since WHERE 1=1 is always true.

The above SQL query is much the same as this:

SELECT User_Id, User_Name, Password FROM Users WHERE User_Id = 104 or 1=1

TABLE I
FONT SIZES FOR PAPERS



SQL Injection.

User-Id : itswadesh

Password : newpassword

select * from Users where user_id= 'itswadesh' and password = ' newpassword '

User-Id : ' OR 1= 1; /*

Password : */--

select * from Users where user_id= '' OR 1 = 1; /* and password = ' */-- '

Illegal/Logically Incorrect Query- *In this type of query an attacker enter incomplete statement such as query terminate without semicolon, causing syntax error in the statement and writing wrong username and password to retrieve information from regarding the table and column name2.The purpose of this query is to extract the data such as table name, column name from the database. If an incomplete statement is sent to a database, some web application servers returns the default error message and the attacker takes the advantage of this weakness3. -*

If the attacker insert the following words into input field pin:“convert(int,(select top 1 sys_name from sysobjects where user_type=’u’)) ”.

The resulting query is: SELECT user_accounts FROM users WHERE login=’’AND pass=’’ AND pin= convert (int,(select top 1 sys_name from sysobjects where user_type=’u’))

Union Query-This type of attack can be accomplished by injecting a union query into an adequate parameter which returns a dataset that is the union of the result of the original first query and the results of the injected statement3. The purpose of this query is to extract data and to bypass authentication. The rules for combining two or more statement using union query are as follows:

1. Column Name & order of columns of all queries should be same.
2. The data types of the columns on each table in the query should be same or compatible.
3. A query return data usually from the first query in the table

Example: The following query is executed from the server

SELECT user_name, phone FROM user_tables WHERE

userid=$id

In which an attacker inserting the following Id value into the credit card table

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

The resulting query is this:

SELECT user_name, phone FROM user_tables WHERE userid =1 UNION ALL SELECT creditCardNumber, 1 FROM Credit CardTable

This would link the resulting query with the original query with all the credit card user

IV. CONSEQUENCE OF SQL INJECTION

As we know the SQL injection is related with the database and in today’s scenario where the database is important assets in a company.Hence with these SQL injection an attacker can take complete control of the database and change the database according to their requirement

- Inject the statement to retrieve information from the company related to username and password.
- Manipulating data in the database
- Delete the database and its description

Insert a user name and password to retrieve credit card information from the system[4].

V. HOW WE DETECT SQL INJECTION ATTACK

The SQL injection attack has been detected through pattern matching techniques against signatures and keywords known to be malicious. Now a days this technique has been successful. The SQL injection attack has been prevented with Trojan Horse through pattern matching techniques against signatures and keywords to identify malicious requests.But these technique has been failed because attackers have know that Trojan Horse is easily recognized with application firewalls and they invent a new technique known as Trojan Zebra5.It is same as Trojan Horse but its colouring and patterns are different.But now a days these type of technique has been useless

Traditional SQL Injection Attack	Evasion Technique	Hidden SQL Injection Attack
…71985’ OR ‘1’ = ‘1’	White Space Manipulation	…71985’OR’1’=’1’
‘&id=111 UNION /**/ SELECT *…’	‘C’ Syntax Comment	&id=111/*This is my comment…*/UN/*Can You*/IO/*Find It*/N/**/S/**/E/* */LE/*Another comment to*/CT/*Find. Can you dig*//*it*/*
1 UNION SELECT ALL FROM WHERE	Encoding: HEX	&#x31;&#x20;&#x55;&#x4E;&#x49;&#x4F;&#x4E;&#x20;&#x53;&#x45;&#x4C;&#x45;&#x43;&#x54;&#x20;&#x41;&#x4C;&#x4C;&#x20;&#x46;&#x52;&#x4F;&#x4D;&#x20;&#x57;&#x48;&#x45;&#x52;&#x45;
	Encoding: BASE 64	MSBVTklPTiBTRUxFQ1QgQUxMIEZST00gV0hFUkU=
	Encoding: DECIMAL	&#49&#32&#85&#78&#73&#79&#78&#32&#83&#69&#76&#69&#67&#84&#32&#65&#76&#76&#32&#70&#82&#79&#77&#32&#87&#72&#69&#82&#69
…71985’ OR ‘1’ = ‘1’	Variations on a Theme	…71985’ OR ‘city’ = ‘Seattle’

Today we are using four categories of evasion technique

VI. WHITE SPACE MANIPULATION

In White space manipulation SQL injection detection engines are proficient of detecting attacks that vary the number and encoding of white spaces around the malicious SQL code. But they fail to detect pattern of text that include one or more spaces they also fail to detect the same pattern of text when no spaces are involve.

Comment Exploitation

Attacker in past days used double hyphen comment syntax for example – to hide their intention but today many engines detect this comment that’s why attacker change their tactics. Now a days attacker use “C” style comments in place of spaces that separate commands and they are easily detected through matches of signature and break up keywords.

Encoding Techniques

It is the easiest method of detecting defection through signatures or pattern matching engines. The effect of encoding is same as cryptography changes the text it is meant to hide from unauthorized users

The most popular encodings used to avoid detection are:

- URL Encoding
- Unicode/UTF-8
- Hex Encoding
- Char() function

But these technique fail because they need to support multiple language and character sets

Variations on a Theme

There are multiple variations of avoidance attacks that are defined in the SQL99 standard.

Concatenation-It detects SQL engine attack by breaking up identifiable keywords to build a single string from multiple pieces. It uses the (+) sign or pipe(||) character to indicate concatenation at the SQL level[6].

For Example

EXEC(‘DES’+’CRIBE US’+’ER’)

EXEC(‘DES’||’CRIBE US’||’ER’)

Conversion- It helps the attacker to avoid detection by introducing valid SQL functions that change

The signature of the statement.

For example: OR username = char(39) /* 39 is equivalent to the SQL wildcard character, % */

VII. PROPOSED TECHNIQUE

A new technique has been proposed in this paper for preventing Database against SQL injection attack. In this approach for storing user account table in final hash value one extra column is required. This value is stored in user account table together with user name and password at the time of new user registration as shown in table

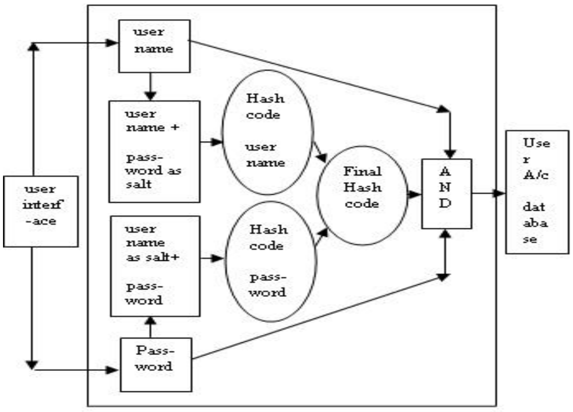
TABLE1

User account table

User Name	Password	Final Hash Code

The final hash code value is calculated at the time of login using stored procedure at run time and user is authenticated by identifying exact matching of username, password and final hash code. According to architecture we will proceed the calculation of final hash code in next section

VIII. ARCHITECTURE



A Proposed technique of architecture is shown in above figure

IX. WORKING METHODOLOGY

The working of proposed technology could be separated into two parts-

Registration of new user- A new user whenever wants to register he/she will fill the login form with a ubiquity name and password at user application. According to the proposed architecture in middle tier this ubiquity name and password is processed

Some steps are given below

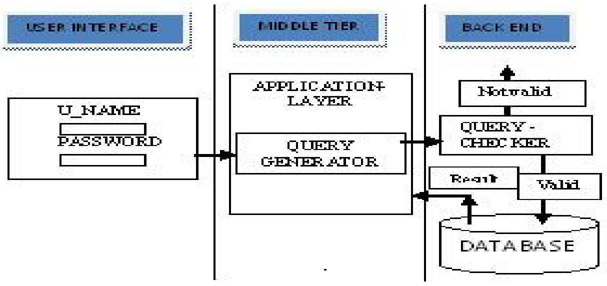
1. Hash code of login name is find using password as salt.
2. Hashcode of password is find using login name as salt.
3. Final hashcode is find by concatenating output of step1 and step 2
4. Login name, Password and Finalhashcode is store (output of step3) to user account table.

Login and verification- A new user whenever wants to register he/she will fill the login form

Some steps are given below

1. Enter a ubiquity name and password at user application.
2. Entered user name is compared with the name stored in user account table.
3. To find Final hash code at run time username matches properly
4. Final Hashcode and Password is verified with stored values
5. Authenticated user is valid to retrieve information from the database otherwise error message is displayed

Working can be shown in below fig



X. EVALUATION OF TECHNIQUE

Evaluation of technique has been performed on table with various number of user records and then computed the response time of the system with embedded technique and without embedded technique. To evaluation of our proposed technique we assume a dummy table in data with different number of values such as 10, 20,30,40,50 and the result display that our proposed method put unimportant expenses into the server in terms of time required in milliseconds. Processing expenses for various numbers of users is describe into the table given below:-

TABLE2

Performance analysis of proposed technique

Total Records	With embedded proposed technique	Without embedded proposed technique
10	11.3	10.3
20	11.8	10.8
30	12.5	11.5
40	12.8	11.8
50	13.2	12.1

XI. CONCLUSIONS AND FUTURE WORK

It is clear from above category that SQL injection attacks is one of the biggest classes of security problems. In this technique developers require to manually or automated specify the interface to an application when applied to modern complicated web application. The main important SQL injection related issues have been reviewed in this paper. We planned a new technique which is based on hash function which is easy and extremely safe from attackers. This paper shows an Authentication method for preventing SQL injection attack and describe its limitation and its application also and the future estimated work describe the efficiency of the system

REFERENCES

[1] Microsoft. "SQL Injection". Retrieved 2013-08-04. "SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives. Even parameterized data can be manipulated by a skilled and determined attacker

[2] SQL Injection Attacks: Detection in a Web Application Environment

[3] Abeer Alhuzali,Hamid Tora,Que Nguyen Analysis of SQL Injection Methods and Its Prevention

[4] Secerno.com," SQL Injection Attack: A Security Threat",

[5] http://www.secerno.com/?pg=SQL-Injection#2

[6] Lori Mac Vittie Technical Marketing Manager F5 Networks, Inc. White Paper

[7] [http://www.owasp.org/index.php/Top_10_2010-A1-Injection, retrieve

[8] on 13/01/2010

[9] Debasish Das,Utpal Sharma & D.K. Bhattacharyya "An Approach to Detect and Prevent SQL Injection Attack Based on Dynamic Query Matching"

[10] K. Amirtahmasebi, S. R. Jalalinia, S. Khadem, "A survey of SQLinjection defense mechanisms," Proc. Of ICITST 2009, vol., no., pp.1-8, 9-12 Nov. 2009