ISSN:0975-9646

An Approach to Remove Security Vulnerability Affected By SQL Code Injection Attack

Shanu Verma[#], Poonam[*]

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

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

***Abstract*— In this era, we are totally dependent on web application like e-banking, e-shopping, online payments of bill etc. Sometime unauthorized users may access confidential data. As a consequence, the users could loss their confidential data or it may face complete destruction There are various type of attacks that can occur by the attacker these are Tautologies, Illegal/logical correct queries, union query, piggy based query, blind injection, timing attack. My method to attack on database is tautologies I implement a mechanism that detect & prevent the SQL injection by incorporating the technique of "CRYPTOGRAPHY HASHING FUNCTION USING MD5 to eliminate SQL Injection vulnerabilities up to some extent. The propose approach is a cryptographic for such attacks. This approach is based on a cryptographic hash-function, which computes the Hash value of user inputs, finds the database record based on the user inputs and compares the encrypted hash value of the input fields against the hash value of the login information stored in the database. The value of MD5 is considered to be a highly reliable fingerprint that can be used to verify the integrity of the file's contents. If as little as a single bit value in the file is modified, the MD5 value for the file will completely change. Forgery of a file generates the same result in MD5 as that for the original file is considered to be extremely difficult. The MD5 checksums for critical system, application, and data files provides a compact way to store information for use during periodic integrity checks of those files In this way, this proposed approach prevents the SQL injection attacks.**

***Keywords*— SQL,TYPE of Attacks,MD5,Cryptography hash function,PHP**

I. INTRODUCTION

E-commerce applications are applications that can be run over the Internet by using any web browser, any operating system or architecture. They have accessed everywhere due to the convenience, flexibility, availability, and interoperability that they provide. E-commerce applications are vulnerable to a variety of new security threats. SQL Injection attack is the most dangerous type of attack. SQL Injection Attacks (SQLIAs) are increasing continuously and create very serious security risks because they can give attackers various access to the database that occur on e-commerce applications.SQL Injection Attack (SQLIA) is a technique for attackers so that attackers attack directly into the database in an unofficial way and accomplish the maximum important information for remove and modifying information from any corporation.SQL injection is a code injection technique that make use of security vulnerability on the database layer of an application in which an attacker can attack on database server by running an unauthorized, unintended SQL query by TAUTOLOGIES ATTACK. SQL Injection attacks are dangerous attack this type of attack occur by the manipulation of the query to the database based on user input by using incorrectly filtered escape characters or bypass authentication method. "SQL injection is an attack in which code is inserted as a strings into database that are later passed to an instance of SQL Server for parsing and execution

II. TYPE OF ATTACKS

There are various types of attacks that can occur by the attacker these are

- Tautologies
- Illegal/logical incorrect queries
- Union query
- Piggy based query
- Blind injection
- Timing attack.

I have used tautologies to attack on database [1]. The Tautologies attack occur in conditional query statement .In SQL we can say that this type of attack occur in 'where' clause. The advantage of tautology attack is to bypass authentication control and return all information when the attack is successful. In which original query is not needed for an attack that' why in which attacker face a challenge to get the data from the database. In which an attacker add an intelligence to attack on data. The hacker has to create a similar website which he wants to attack. He tries to make a similar URL to the original website.
To create a similar website some steps are taken [2]

- Buying or otherwise getting authorized access to a similar piece of hardware and establishing the configuration page URLs
- Finding them out from an online manual or a Web forum
- Sniffing network traffic
- Educated guessing (perhaps remote.html)

In Illegal/logical incorrect queries, query is not needed an error message is returned from the database which contain some useful information for the attacker. In this type of attack attacker use illegal keyword such as convert and so on to the database and the result of the query is to create

syntax error, logical error and so on. With the help of this query attacker reach the original query. In Union query attacker attach injected query with the word UNION to the correct query [3]. The 'correct query' which is return from the illegal/logical correct queries and get data about other tables in a database.

In Piggy based queries attacker attach extra query or keyword such as ';' or ':' and so on to the original query to get the useful information from the database. In Blind injection attack in which attacker face a challenge to get the data from the database. In which error which is shown by this attack is hide by the developer from the database. In which attack would be very difficult but not impossible. An attacker attack data by asking a series of true/false question. In Blind Injection attack attacker success depend upon the application if the application is unsecured the attack would be successful. In Timing attack attacker collect information from the database by observing timing delay in the database response. This is implemented using 'if-then-clause' when you execute a long running query. The keyword used in timing attack is 'WAIT FOR' by the attacker to attack on the database.

A. IMPLEMENTATION OF TAUTOLOGIES ATTACK

Your User submits username and password for access the database as "Shanu" and "shanu@v," the application dynamically builds the query given below [1]

SELECT username, password FROM users WHERE username='Shanu' AND password='shanu@v'

Attacker enters 1=1' OR '1=1 as the username and passwords, the resulting query:

SELECT username, password FROM users WHERE username=1=1' OR '1=1 AND password=1=1' or '1=1

Another user submits username and password for access the database as "guest" and "secret" the application dynamically builds the query given below[4]:

Select member_id, member_level from members where member_login='guest' and member_password = 'secret'''

A malicious user enter input "' or 1=1- -"in the first field and leave the second input field as blank. The resultant query will be

Select member_id, member_level from members where member_login=''or 1=1- -'and member_password = ''''

retrieve all records from the database. A typical SQL statement for SQL injection attack will look the statement as follows:

SELECT * FROM Users WHERE User_id='abc' AND Password=.'tcy12'

This statement will retrieve the User_id and Password column from the user's table, returning all rows in the table where User_id is abc and password is tcy12. An important point to note here is that the string literals 'abc' and 'tcy12' are delimited with single quotes.

Now, presuming that the User_id and Password fields are being gathered from user supplied input at the time user logins through web page, an attacker might be able to 'inject' SQL query, by inputting values into web applications like this:

User_id='OR''='

Password='OR''='

The query string become like this:

Select * from users Where User_id='' ='OR''=' AND Password= '' ='OR''='

Now, when database attempts to run this query, it simply executes without giving any error. The attacker could log in as the first user in the user's table and can access the information in the database without having a valid login. By this way, attacker could gain access to unauthorized information

III. PREVENTING SQL INJECTION ATTACKS

In this world of Information technology, where E-commerce is most popular, the need for secure and safe data on Internet is must. Web applications, which are the foremost way of accessing data from web, are highly vulnerable to SQLIAs. Such applications and their underlying databases often contain confidential or even very sensitive information such as customer and financial records. With the increase in the availability and popularity of database driven web applications, there is a corresponding increase in number and sophistication of attacks that target them. Therefore it is very difficult to prevent these applications from attackers in order to save the critical information being hacked[5]. In this project propose a new approach to prevent SQL injection attacks and also to eliminate SQL Injection vulnerabilities up to some extent. I have used PHP to design the web page as the client side technology and the server side designed database on MYSQL. In this project I, implement a mechanism that detect & prevent the SQL injection by incorporating the technique of "CRYPTOGRAPHY HASHING FUNCTION USING MD5".

A. CRYPTOGRAPHY HASHING FUNCTION USING MD5

It is a hash function which is considered practically impossible to convert that is, to recreate the input data from its hash value alone. The input data is called the message, and the hash value is called the message digest or simply the digest.

The ideal cryptographic hash function has four main properties:

- It is easy to compute the hash value for any given message
- It is infeasible to generate a message that has a given hash
- It is infeasible to modify a message without changing the hash
- It is infeasible to find two different messages with the same hash

Cryptographic hash function has application used in Digital Signature, Fingerprinting to

Detect duplicate data, Password Verification. In 2013 Password Hashing Competition was announced to choose a new, standard algorithm for password hashing [6].

B. What is MD5 (Message Digest)?

The MD5 function is a cryptographic algorithm that takes an input of arbitrary length and produces a message digest that is 128 bits long. The digest is called the "hash" or "fingerprint" of the input. MD5 is widely used in security-related applications, and is also frequently used to check the integrity of files. The value of MD5 is considered to be a highly reliable fingerprint that can be used to verify the integrity of the file's contents. If as little as a single bit value in the file is modified, the MD5 value for the file will completely change. Forgery of a file generates the same result in MD5 as that for the original file is considered to be extremely difficult. The set of MD5 checksums for critical system, application, and data files provides a compact way to store information for use during periodic integrity checks of those files [7]. MD5 is used in many situations where a potentially long message needs to be processed and/or compared quickly. The most common application is the creation and verification of digital signatures. MD5 designed by well-known cryptographer Ronald Rivest in 1991. There are various series of MD such as MD2, MD4, MD but most popular is MD5 (Message Digest Algorithm. MD2 optimized for 8-bit machines, whereas MD4 and MD5 were aimed at 32-bit machines.MD5 has more secured than MD4.

C. MD5 Algorithm

The algorithm consists of four distinct rounds, which has a slightly different design from that of MD4. MD5 algorithm can be used as a digit signature mechanism

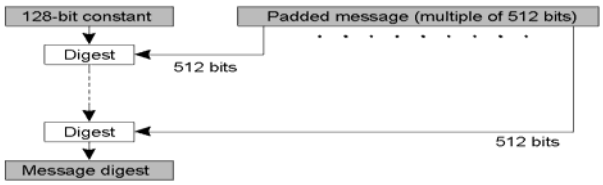

Figure MD5 Algorithm Structure [8]

Implementation Steps of MD5 Algorithm

- Step1 Append padding bits
- Step2. Append length
- Step3. Initialize MD buffer
- Step4. Process message in 16-word blocks

MD5 is being used heavily from large corporations, such as IBM, Cisco Systems, to individual programmers. MD5 is considered one of the most efficient algorithms currently available.

D *Query used for Prevention:*

```
If (isset ($_POST) && !empty ($_POST))
 mysql_real_escape_string (stripslashes ($_POST ['username']))
 mysql_real_escape_string (stripslashes ($_POST [‘email']))
 $match = "select id from $table where username = '". $username."' and password = '". $password."';";
   $qry = mysql_query ($match);
   mysql_fetch_array ($qry);
```

I have used MD5 cryptographic hash function for creating a signature that is used to authenticate user based on their user name and password. Hashing is a mechanism for figuring out if two things are similar and is a one-way process. You consider an object such as file, string of text, ISO and convert it to a fixed length string. We enter the username name and password in the login page after that username and password are matched with stored username and password in the database. This technique is vulnerable by SQLIAs, see “Cryptographic countermeasure for SQLIAs” is based on a cryptographic Hash-Function which computes the Hash value of user inputs, finds the database record based on the user inputs and compares the hash value of the input fields against the hash value of the username and password found in the database. Hash value never be same of the original and the modified or changed object [9]. That is why, SQLIAs cannot break the hash value. In programming, one of the most common methods for hashing is password. In this scenario it is done for two reasons:

1. *You not store your passwords in Plaintext in a database*
2. *You never should care what the user's password is*

Since #2 means you never need to know what the hash stands for, Hashing a light-weight alternative to encryption while still is providing security.

***E** Create hash password using MD5*

```
$password = md5($_POST['password']);
$username = $_POST['username'];
$result = mysql_query("SELECT username, password FROM user_accounts WHERE password = '$password' AND username = '$username'");
if( mysql_num_rows($result) == 1 ) { echo 'Found a proper account!'; } else { echo 'Invalid username and Password'; }
```

The two most common ways to hash a file was to use either of the built-in hashing functions for MD5 or SHA1.But like MD5, SHA1 is not considered a good cryptography hash function. that’s why I, use MD5.MD5 works on every platform [10].

```
$sha1 = sha1 ('This is a string');
$md5 = md5 ('This is a string');
```

IV. CONCLUSIONS

The version of this template is V2. Most of the formatting instructions in this document have been compiled by Causal Productions from the IEEE LaTeX style files. Causal Productions offers both A4 templates and US Letter templates for LaTeX and Microsoft Word SQLIAs have

evolved over years. Information Security Researchers invented newer and newer techniques to eliminate the number of problem and sophistications of attacks have increased rapidly. The mode of attack and its various methodologies define the work of providing security to web based applications. It would be quite inappropriate to tell exactly the future work in this area because it can only be evolved according to the sophistication of a new attacks found by the security persons. Web applications may be designed in such a way that any attempt to attack via SQL is monitored and it is checked before trying to generate a signature based on the malicious input. This will save time and optimize the solution. By conducting a comprehensive survey on existing techniques, I have realized that many SQL injection countermeasures have their limitations. Understanding and identifying the working mechanisms, as well as advantages and disadvantages of current techniques will benefit future work in this area.

REFERENCES

[1] S. Labs. SQL Injection. White paper, SPI Dynamics, Inc., 2002. http://www.spidy namics.com/assets/documents/WhitepaperSQLInjection.pdf
[2] M.TimJones, Firmware Architect, Independent author www.ibm.com/developer works/security/library/se-sql-injection-attacks/index.html?ca=drs
[3] C. Anley. Advanced SQL Injection In SQL Server Applications.White paper, Next Generation Security Software Ltd., 2002
[4] http://hackmageddon.com/tag/sql-injection/
[5] http://blog.insecure.in/?tag=sql-injection
[6] "Password Hashing Competition". Retrieved March 3, 2013
[7] Arnoud Engelfriet iusmentis technology/hashfunctions/md5/
[8] CS265 Spring 2003 Jerry Li Computer Science Department San Jose State University
[9] https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat _Sheet
[10] http://php.net/manual/en/function.md5.php
[11] Web Application security Consortium the web hacking incident database