International Research Journal in Global Engineering and Sciences. (IRJGES)
ISSN : 2456-172X | Vol. 3, No. 1, March - May, 2018
Pages 30-36 | Cosmos Impact Factor (Germany): 5.195
Received: 06.03.2018 Published : 28.03.2018

Database Application Protection Using MD5 Algorithm

[1]P. Mullai, [2]Mahesh. [2]KP, S.Subasree
[1]Faculty, Department of Computer Science and Engineering
SRM Institute of Science and Technology, India
[2]Scholars, Department of Computer Science and Engineering
SRM Institute of Science and Technology, India
Corresponding Author: maheshkp3196@gmail.com

Abstract — In the present generation, people are more connected to the digital life. Thereby, the importance of securing the digital data has become more vital. The field of Security, the application of Internet, the Communication and Computer Network technologies have developed rapidly. Especially the emergence of Internet has increased the usage of Computer for different applications in government, business, health Care and other areas of society resulting in a greater impact on people's economy, work and life. There are different types of hackers like the script kiddie, white hat, black hat, gray hat, green hat, red hat and blue hat, where these hackers can hack all the information and make the security under serious threat. Likewise, the database hackers can hack the database information by various attacks like the code modification attacks, SQL injections and data centric attacks. Thereby, an efficient technique implementing the MD5 algorithm is used to provide more protection to detect the anomaly and to maintain the integrity of the data in the database applications.

Index Terms— Anomaly, Code modification attacks, Database, Data centric attacks, Security and SQL injections.

I. INTRODUCTION

The Structured Query Language (SQL) is a language that is used to query, operate, and administer the database systems such as Microsoft SQL Server, Oracle, and MySQL. The general use of SQL is consistent across all the database systems that support it. Web Intrusion refers to any attempt to threaten the confidentiality or availability of the data in the web application. SQL injection is a code injection technique used to attack the data-driven applications, in which the malicious SQL statements are inserted into an entry field for execution. It consists of injection of a SQL query via the input data from the client to the application. A successful SQL injection statement can read the sensitive data from the database, modify database data (such as Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS) and recover the content of a given file present on the DBMS file system.

SQL injection can be used to perform the following types of attacks:

- Authentication Bypass: The attacker can log on to an application, potentially with administrative privileges, without supplying a valid username and password.
- Information Disclosure: The attacker can obtain the sensitive information from the database.
- Compromised Availability of Data: This attack allows an attacker to delete information to cause harm, delete log or audit information in a database.

A SQL injection attack involves the alteration of SQL statements that are used in a web application through the use of attacker-supplied data. By injecting a SQL statement, the attacker can access the

International Research Journal in Global Engineering and Sciences. (IRJGES)
ISSN : 2456-172X | Vol. 3, No. 1, March - May, 2018
Pages 30-36 | Cosmos Impact Factor (Germany): 5.195
Received: 06.03.2018 Published : 28.03.2018

information stored in the web site's database. SQL injection can be used to attack any type of SQL database. SQL Injection attacks are common for two reasons:

- The occurrence of SQL injection vulnerabilities.
- Databases are attractive targets and they may contain critical application information.

II. PRELIMINARIES

A. System Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example, the design must not have any logic faults in the design is detected before coding commences, otherwise, the cost of fixing the faults will be considerably higher as reflected. Detection of design faults is achieved by means of inspection as well as walkthrough. Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

Unit Testing

Unit testing is used to verify the functional performance of every modular component of the software. It focuses on the smallest unit of the software design. The white-box testing techniques were heavily employed for unit testing.

Performance Testing

It determines the amount of execution time spent in various parts of the unit, program throughput, response time and device utilization by the program unit.

White box Testing

It is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been designed to perform test, can be conducted to check whether every function is fully operational and to check for errors in each function.

Black box Testing

In this testing, by knowing the internal operation of a product, test can be conducted to ensure that the internal operation performs according to the specifications and that all the internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

Security Testing

This testing attempts to verify the protection mechanisms built in a system, in fact, to protect it from an improper penetration. The system security must be tested for invulnerability from frontal attack and rear attack. During the security testing, the tester places the role of the individual who desires to penetrate system.

International Research Journal in Global Engineering and Sciences. (IRJGES)
ISSN : 2456-172X | Vol. 3, No. 1, March - May, 2018
Pages 30-36 | Cosmos Impact Factor (Germany): 5.195
Received: 06.03.2018 Published : 28.03.2018

B. SQL Injection Attacks (SQLIA)

The SQL injection attacks are classified into different types. The tautology attacks are the easiest to detect, because they introduce tautologies in the WHERE clause that can be easily detected by a simple SQL parser. Some attacks consist of sending illegal queries, because by analyzing the resulting error messages, it is possible to infer meaningful information about the schema and the type of database. Such attacks can be identified by analyzing the error logs. The remaining attacks use special encoding or uneasy parameters to alter the executed query.

III. SYSTEM ARCHITECTURE

The system architecture consists of several modules, supporting the Database application protection, that we describe in what follows.

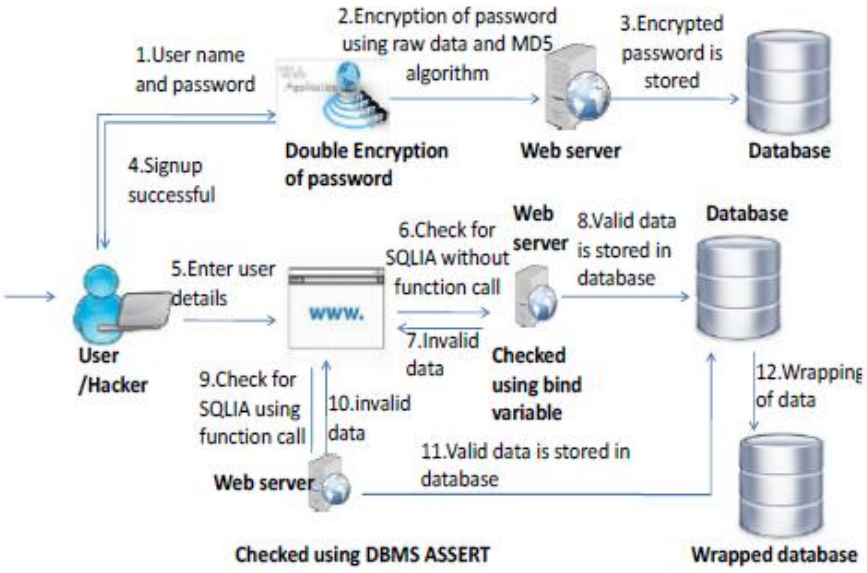


Fig. 1. System Architecture

A. Encryption

Here, the input password is converted into hexadecimal digits, using the MD5 algorithm. The converted information is stored onto the database in encrypted form.

Pseudocode- MD5 Algorithm:
INPUT: Encrypted password in raw data form.
OUTPUT: Double encrypted password
BEGIN STEP 1: Append Padding Bits. The original message is "padded" (extended) so that its length (in bits) is congruent to 448, modulo 512.
STEP 1.1: The original message is always padded with one bit "1" first.
STEP 1.2: Then zero or more bits "0" are padded to bring the length of the message up to 64 bits fewer than a multiple of 512.

International Research Journal in Global Engineering and Sciences. (IRJGES)
ISSN : 2456-172X | Vol. 3, No. 1, March - May, 2018
Pages 30-36 | Cosmos Impact Factor (Germany): 5.195
Received: 06.03.2018 Published : 28.03.2018

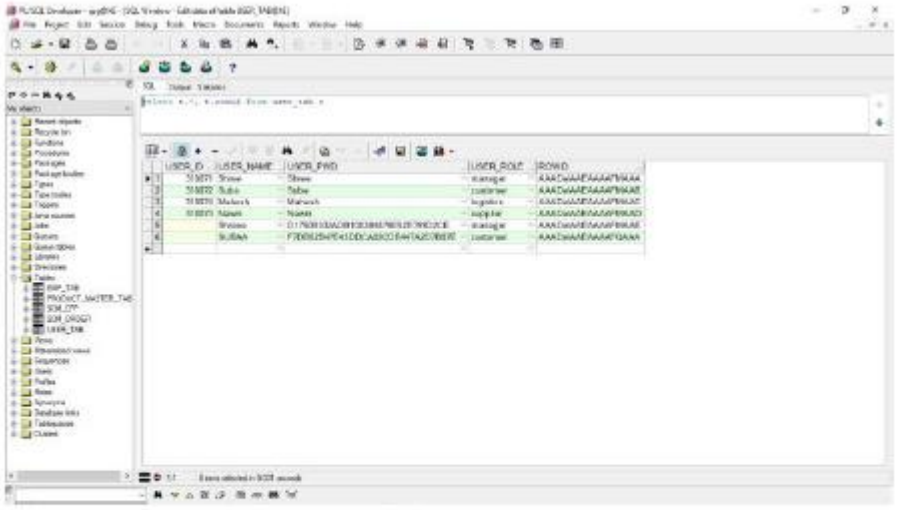
STEP 2: Append Length. 64 bits are appended to the end of the padded message to indicate the length of the original message in bytes.
STEP 2.1: The length of the original message in bytes is converted to its binary format of 64 bits. If overflow happens, only the low-order 64 bits are used.
STEP 2.2: Break the 64-bit length into 2 words (32 bits each).
STEP 2.3: The low-order word is appended first and followed by the high-order word.
STEP 3: Initialize MD Buffer. MD5 algorithm requires a 128-bit buffer with a specific initial value.
STEP 3.1: The buffer is divided into 4 words (32 bits each), named as A, B, C, and D.

Word A is initialized to: 0x67452301.
Word B is initialized to: 0xEFCDAB89.
Word C is initialized to: 0x98BADCFE.
Word D is initialized to: 0x10325476.

STEP 4: Processing Message in 512-bit Blocks. This is the main step of MD 5 algorithm, which loops through the padded and appended message in blocks of 512 bits each. For each input block, 4 rounds of logical operations (Using and, or and not) are performed with 16 operations in each round. For example, F(X,Y,Z)= (X and Y) or (not(X) and Y)
STEP 5: Output. The contents in buffer words A, B, C, D are returned of sequence with low-order byte first. END

MD5 Conversion:

```
declare
pass varchar2(32):='ad';
pass2 varchar2(32);
pass3 varchar2(32);
begin
dbms_output.put_line('My password :'||pass);
pass2:=pass;
pass2:= utl_raw.cast_to_raw(pass2);
dbms_output.put_line('My first conversion :'||pass2);
pass3:=dbms_obfuscation_toolkit.MD5(input=>pass2); dbms_output.put_line('My second
conversion :'||pass3);     end;
```



B. Identification and Prevention of SQL Injection attack using Bind Variable

International Research Journal in Global Engineering and Sciences. (IRJGES)
ISSN : 2456-172X | Vol. 3, No. 1, March - May, 2018
Pages 30-36 | Cosmos Impact Factor (Germany): 5.195
Received: 06.03.2018 Published : 28.03.2018

The main idea is identification and prevention of the SQL injection attack. The user details which are supposed to be the input for this module are checked for Bypass Login Attack (SQL Injection Attack) using the bind variable. If it is not a SQL injection code, the process proceeds. If it is a SQL injection code, the attacks are prevented.

Steps- Bind Variable Technique: STEP 1: Each time an SQL statement is sent to a database, an exact text match is performed to see if the statement is already present in the shared pool. STEP 2: If no matching statement is found, the database performs a hard parse of that statement. STEP 3: If a matching statement is found, then the database initiates a soft parse.

C. Identification and prevention SQL injection attack using DBMS assert.

Here, the attackers try to insert a number of additional illegal SQL statements into the normal structure of the dynamic SQL statement, that results in the execution of the normal SQL statements that the client sends along with these additional SQL statements which attackers construct. These additional SQL statements are often key operations to the database such as deleting the data table, modifying table fields, adding data and deleting data. In this type of SQLIA attackers inject a statement of the form "UNION < injected query >". By using function call statements, the attackers can retrieve the table name and information from the specified table. To overcome this, the DBMS_ASSERT package is used. Using which, we avoid the Function Call Injection problems. The product details are checked for the function call injections and SQLIA. They are validated using the DBMS assert package. If there are no intrusions, the details are stored in the database.

If there is any intrusion, then raises an error on the webpage. It validates and ignores the fake actions that the attackers try to execute.
Pseudocode- DBMS assert
INPUT: Product details.
OUTPUT: Validated product details.
BEGIN STEP 1: The queries are passed through the DBMS ASSERT package.
STEP 2: Each query is processed word by word and the keywords are checked for intrusions.
STEP 3: An error is reported to the web application if there is an intrusion.

International Research Journal in Global Engineering and Sciences. (IRJGES)
ISSN : 2456-172X | Vol. 3, No. 1, March - May, 2018
Pages 30-36 | Cosmos Impact Factor (Germany): 5.195
Received: 06.03.2018 Published : 28.03.2018

STEP 4: If there is no intrusion then the process is continued. END

D. Wrapping Database using Wrapping function

The database schema information and the source codes are to be secured and hidden from the hackers. This is done using the wrapping function. It hides all the information from hackers by using the wrapping function in the same package. The wrapped information is then processed to be stored into another database (proxy database).

IV. CONCLUSION

In this paper, we analyze the various attacks on the database and find the preventive measures. The attacks are identified and validated using various techniques and algorithms. Hackers cannot bypass login authentication, since we provide double encryption. Hacking the source code from the server is prevented using the wrapping function. Thus, the system will prevent the web intrusions and query attacks in the web page. This is implemented in Oracle 10g. It is more flexible for further development. This can be further enhanced by reducing the disk space required for storing the data. The information should be wrapped in a more secure manner so that no one can access it. The level of security for database application can be enhanced using different algorithms and techniques in the future.

APPENDIX

SQL- Structured Query Language
DBMS- Database Management System
SQLIA- SQL Injection Attack

International Research Journal in Global Engineering and Sciences. (IRJGES)
ISSN : 2456-172X | Vol. 3, No. 1, March - May, 2018
Pages 30-36 | Cosmos Impact Factor (Germany): 5.195
Received: 06.03.2018 Published : 28.03.2018

REFERENCES

[1] P. Anbalagan and M. Vouk, "Towards a Unifying Approach in Understanding Security Problems," Proc. Int'l Symp. Software Reliability Eng., pp. 136-145, 2009
[2] R. Chillarege, I.S. Bhandari, J.K. Chaar, M.J. Halliday, D. Moebus, B. Ray, and M. Wong, "Orthogonal Defect Classification—A Concept for In-Process Measurement," IEEE Trans. Software Eng., vol. 18, no. 11, pp. 943-956, Nov. 1992.
[3] A. Avizienis, J.C. Laprie, B. Randell, and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," IEEE Trans. Dependable and Secure Computing, vol. 1, no. 1, pp. 11- 33, Jan.-Mar. 2004.
[4] Acunetix Ltd., "Is Your Website Hackable? Do a Web Security Audit with Acunetix Web Vulnerability Scanner," http://www.acunetix.com/security-audit/index/, May 2013.
[5] G. _ Alvarez and S. Petrovic, "A New Taxonomy of Web Attacks Suitable for Efficient Encoding," Computers and Security, vol. 22, no. 5, pp. 435-449, July 2003.
[6] P. Anbalagan and M. Vouk, "Towards a Unifying Approach in Understanding Security Problems," Proc. Int'l Symp. Software Reliability Eng., pp. 136-145, 2009.
[7] J. Dura˜es and H. Madeira, "Emulation of Software Faults: A Field Data Study and a Practical Approach," Trans. Software Eng., vol. 32, pp. 849-867, 2006.
[8] A. Adelsbach, D. Alessandri, C. Cachin, S. Creese, Y. Deswarte, K. Kursawe, J.C. Laprie, D. Powell, B. Randell, J.Riordan, P. Ryan,W. Simmonds, R. Stroud, P. Verissimo, M. Waidner, and A.Wespi, "Conceptual Model and Architecture of MAFTIA," Project IST-1999-11583
[9] JM. Cukier, R. Berthier, S. Panjwani, and S. Tan, "A Statistical Analysis of Attack Data to Separate Attacks," Proc. Int'l Conf. Dependable Systems and Networks, pp. 383- 392, 2006.
[10] J. Christmansson and R. Chillarege, "Generation of an Error Set That Emulates Software Faults," Proc. IEEE Fault Tolerant Computing Symp., pp. 304-313, 1996.