Prevention of SQL Injection Attacks Using Hashing Algorithm

K RAMBABU[1], BATTULA YESU RAJU [2],

[1] **Assistant professor(HOD), MCA DEPT,** Dantuluri Narayana Raju College**, Bhimavaram, Andharapradesh**

[2]**PG Student of MCA, Dantuluri** Narayana Raju College**, Bhimavaram, Andharapradesh**

Abstract Web Applications form an integral part of our day-to-day life. The number of attacks on websites and the compromise of many individuals secures data are increasing at an alarming rate. With the advent of social networking and e-commerce, web security attacks such as phishing and spamming have become quite common. The consequences of these attacks are ruthless. Hence, providing an increased amount of security for the users and their data becomes essential. Most important vulnerability as described in top 10 web security issues by Open Web Application Security Project is SQL Injection Attack (SQLIA). This paper focuses on how the advantages of randomization can be employed to prevent SQL injection attacks in web-based applications. SQL injection can be used for unauthorized access to an database to penetrate the application illegally, modify the database or even remove it. For a hacker to modify a database, details such as field and table names are required So we try to propose a solution to the above problem by preventing it using an encryption algorithm based on randomization and other solution is using the Hirschberg algorithm, it is a divide and conquer approach to reduce the time and space complexity. It has better performance and provides increased security in comparison to the existing solutions. Also, the time to crack the database takes more time when techniques such as dictionary and brute force attack are deployed Our main aim is to provide increased security by developing a tool which prevents illegal access to the database.

Index Terms: - Web Application Security, SQL Injection Attack (SQLIA), Hirschberg algorithm

I Introduction

According to the report by the White Hat on web security vulnerabilities 2011, it shows that nearly 14-15% of web application attacks account for SQL Injection.With the increasing attacks on web applications, it is very important to have awareness about the existing attacks, because vulnerabilities such as phishing, social engineering attack, denial of service attacks have become very common. The most basic Social Engineering attacks are Phishing and Email spamming.

Another emerging phishing attack is Tab Nabbing, which can deceive even tech savvy online users. A survey on web security was conducted by us and a total of about100 students participated. It is really surprising to note that nearly 80 % were unable to identify phishing attack and around 70% could not identify Email spam. Hence there is an end that every one has basic awareness about web security since most the confidential transactions are carried out on the web.

In this paper, we take up SQL Injection, a critical web security vulnerability. SQLIA is a type of code-injection attack. It is caused mainly due to improper validation of user input. Solutions addressed to prevent SQL Injection Attacks include existing defensive coding practices along side encryption algorithms based on randomization. Defensive coding mechanisms are sometimes prone to errors, hence not complete in eradicating the effect of vulnerability. Defensive programming is sometimes very labor intensive, thus not very

effective in preventing SQLIA. SQL Injection Attack is application level security vulnerability.

The main intent to use SQL injection attack include illegal access to a database, extracting information from the database, modifying the existing database, escalation of privileges of the user or tool function an application. Ultimately SQLIA involves unauthorized access to a database exploiting the vulnerable parameters of a web application.

A novel idea to detect and prevent SQLIA, an application specific encryption algorithm based on randomization is proposed and its effectiveness is measured. There many methods to illegally access a database using SQLIA and most of the solutions proposed to detect and prevent it are able to solve only problems related to a subset of the attack methods.

.2 Literature survey

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy n company strength. Once these things are satisfied, then ext steps is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

SQL Injection is one of the main issues in database security. It affects the data base without the knowledge of the database administrator. It may delete the full database or records or tables without the knowledge of the respective user or administrator. It is a technique used to exploit the database system through vulnerable web applications. these attacks not only make the attacker to breach the security and steal the entire content of the database but also, To make arbitrary changes to both the database schema and the contents. SQL injection attack could not be realized about information compromisation until long after the attack has passed. In many scenarios, the victims are unaware that their confidential data has been stolen or compromised. With the help of a simple web browser, SQL Injection attacks can be performed by attackers.

In order to locate the hotspots where SQLIA vulnerability occurs, we first discuss the 3-tier logical view architecture of web applications.

A.3-tier Architecture of web application

User interface tier: This layer forms the front end of the web application. It interacts with the other layers based on the inputs provided by the user.

Business logictier: The user request and its processing are done here. It involves the server side programming logic. Forms the intermediate layer between the user interface tier and the data base tier.

Pros

The complex application rules are easy to apply in the application server.

- Business logic is off-loaded from the database server and client ,this enhances performance.

- If there are changes to the specific business's logic, these changes are automatically imposed by the server. The changes only require new application server software to be installed.

- Application server logic is portable to other database server platforms by virtue of the application software.

Cons

- It uses more complex structures
- It is more difficult to set up and maintains

3 Implementation Study

Using tautology: The example quoted in the section II.B to explain SQL injection describes the attack using tautology.

This section briefly describes about the types of SQL I attacks [10] for which we propose solutions in the next section.

Using illegal/incorrect queries: By providing in correct inputs, the data base might return some important information regarding the table and fields used in the database. Using successive requests like these, the security can be compromised. In the following example instead of a Valid username (xyz), an incorrect input (xyz') is provided and an error message is returned giving clues about the data base making it vulnerable to intrusion.

Error: SELECT username, password from student WHERE username =xyz'Here field names such as username and password from table student are exposed.

1) Using Piggy-Backed Queries:Previously mentioned attack type strytogain

unauthorized access to the application or fetch details about the database without any additional queries added to the input query .If additional queries are piggy- backed in the input area using special characters such as " ","; –" or" ; " [5] hackers can modify or delete the database. An example to this is:

SELECT*from User WHERE id=123;droptable User;

Here";"acts as a de limiterand additional an drop statement may be executed and deletes the table.

2) Blind injection: In case of incorrect queries, programmers try to hide database information a same a sure to protect the application from attacks using illegal/incorrect

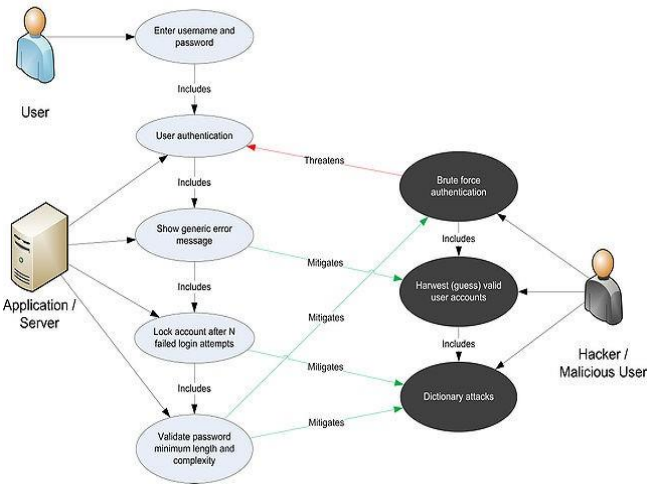
3) queries. Now, the hacker does not get any clues about the database. The option hence used to compromise security is, by querying the database with a lot of true/false queries and checking its result. Based on the response of the application to the queries, the hackers attempt illegal access. This type of attack is most prominently used. It includes timing attacks [10] as one of the technique to identify the behavior of the application.

3.1proposed methodology

A. Client Side Variation

Using client side script validation such as Java Script, a lot of SQL injection attacks can be prevented in Web application. Though this approach does not solve all the attack types, it is necessary to provide the basic security to prevent illegal attacks. The sequence of steps that increase the level of security in case of vulnerabilities is depicted in the activity diagram[Fig3].

The advantage of client side validation is that it reduces CPU cycles since it avoids a number of round trips to the server. Some of the steps involved in client side validation include limiting the input size, restricting the use of special characters etc. But limiting the size of the input and restricting the use of special characters cannot be imposed on users in all applications. Also the protection provided by client side scripts can be easily by passed. The use of this approach can solve attack susingt autology or in correct queries. It cannot solve the threat posed by blind injection techniques. Hence we prefer server side validation techniques.



.

Fig 1: - proposed model

4 Results and Evolution Metrics

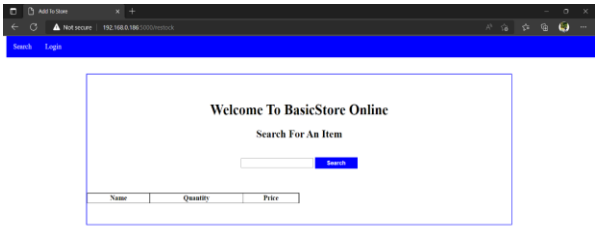


Fig 2:.home page

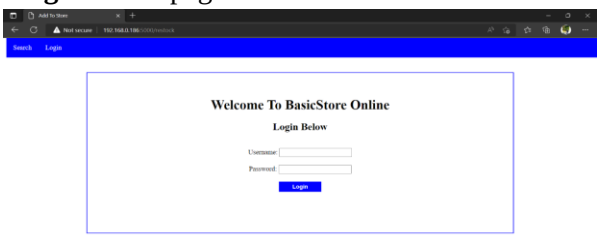


Fig 3:Login page

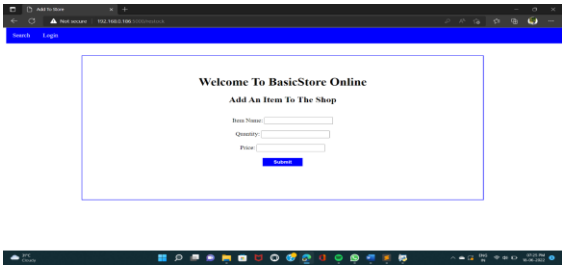


Fig 4:- adding item to the database

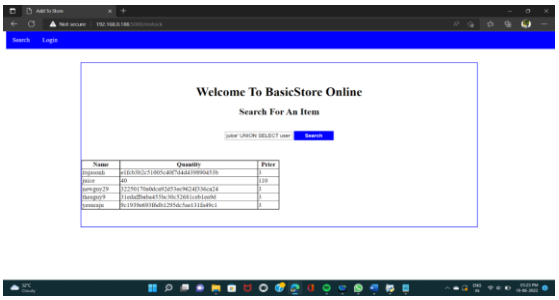


Fig 5 :- If the hacker uses the above injection commend in the search bar. The commend used shows the employee's user id and passwords. But not show the password because they encrypted the passwords using the hashing algorithm. So, attackers are understand the passwords.

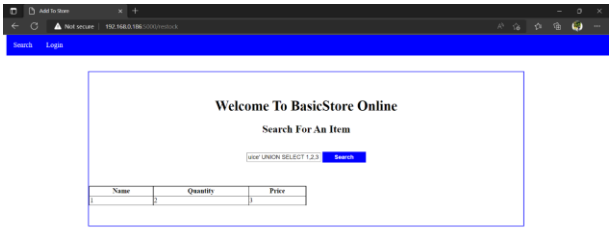


Fig 6:-user inputs query **juice' UNION SELECT 1,2,3 from sqlite_master WHERE type="table"; -- to hack the information**

5 Conclusion

SQL injection is one most important web security threat that needs attention so as to improve security for the users and their data. This paper deals with an application specific randomized encryption algorithm to detect and prevent it. Further its effectiveness was compared with other existing techniques and its performance was quantified. Hence we took up this web security vulnerability and analyzed its attack types. Security threat posed SQLIA is really high and it is very necessary to protect user's data in a web application, since it is very confidential and sensitive.

6 References

1.Ashishkamra,ElisaBertino,GuyLebanon," Mechanismsfordatabaseintrusiondetectionan dresponse",Datasecurity&privacy,Pages31-36,ACM,2008.

2.Xiang Fu, Xin Lu, Boris Peltsverger, Shijun Chen, "A Static Analysis Framework For Detecting SQL Injection Vulnerabilities", IEEE Transaction of computer software and application conference,2007.

3.C.Gould, Z.Su, P.Devanbu,Static Checking of Dynamically Generated Queries in Database Applications ,ICSE2004

4.G.Wassermann,Z.Su,An AnalysisFramework for Security inWebApplications,pp.70-78,SAVCBS2004

5.G.T.Buehrer,B.W.Weide,P.A.G.Sivilotti,Usi ngParseTree ValidationtoPreventSQL InjectionAttacks,SEM2005

6.W.G.Halfond,A.Orso,CombiningStaticAna lysis&RuntimeMonitoringtoCounterSQL-InjectionAttacks,WODA2005

7.W.G.Halfond,A.Orso,AMNESIA:Analysis andMonitoringforNEutralizingSQLInjection Attacks,ASE2005