IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 6, June 2016

SQL Injection Detection and Prevention using Pattern Matching Algorithm

Poornima Javali[1], Sushma V. Chougule[1]

Department of CSE, KLE Dr.MSS College of Engg & Tech, Belgaum[1]

Abstract: Web applications play important role in our daily life. It has made our life easy and simple, we use web applications in almost every fields like online banking, online shopping, to read news paper, in government sectors to pay tax these all applications maintain huge amount of user data and prone to attacks dangerous attacks like SQLI (Structural Query Language Injection) attacks. In this paper we used Pattern matching algorithm detect and prevent SQLI attacks on websites, hence provide security to website.

Keywords: Include (SQL Injection Attack); web applications; pattern matching; malicious SQL commands.

I. INTRODUCTION

Development of internet had huge impact on day to day life. Many organizations and companies provide their services through internet to their customers. Organizations like government, financial, online shopping portal, news sites, keep and maintain data driven web applications, where information regarding the company products, services, customers, partnerships, credit card details, etc, are stored.

Web – a complex system of interconnected elements decreases the cost of distributing information, products, and services. As the coin has two faces webs are also vulnerable to attacks. Numbers of attacks on these web applications are increasing every day leading to compromising of individual's confidential data. Thus providing security to data is essential. These data driven websites are vulnerable to SQLI (Structural Query Language Injection) attacks, the main reason is improper validation or filtration of user input. Top ten web security issues are of the type of SQL injection attacks. In SQLI types attacks users input is converted in to the SQL query, using this vulnerability advantage attacker modifies the query to gain unauthorized access to web applications and their underlying database by submitting malicious SQL commands.

SQLI is attack on database driven web applications or websites, which is done by inserting a malicious SQL statement to gain and modify the confidential data of database also called as code injection technique. Any procedures that construct SQL statement from user input are prone to SQL injection attacks. Also it can be stated as; SQLI is technique where attacker or intruder injects malicious input to gain unauthorized access to web applications.

In such web applications user inputs are taken as part of SQL query, thus attacker can take advantage of this vulnerability to gain unauthorized access to websites and its database by submitting SQL commands thus affecting data integrity.

II. RELATED WORK

[1] Haeng Kon Kim states that web application has become need of our life and availability of these web services incurs many problems about security and proposes a P_SQLIAD (Pattern based SQL Injection Attack Detection) to detect the SQL injection attacks using patterns. The P_SQLIAD has two modules PCM (Pattern Create Module) and ADM (Attack Detection Module). Here PCM creates SQL pattern and saves in database. ADM uses the pattern which is saved in database by PCM to detect SQL injection attacks. P_SQLIAD detects the SQL injection and cross site scripting (XSS) of web application attacks and the client side it shows an error message saying username and password are invalid.

[3] R. Joseph Manoj, Dr. A. Chandrashekhar proposed an approach to detect and prevent tautology type SQL Injection using XSchema validation with runtime monitoring. In this approach time taken to detect the attack was very less. But the limitation of this system is it can detect and prevent only tautology kind of attacks and not applicable to other type of SQLI attacks. In paper [4] explains that banking, shopping and other social networking applications deal with sensitive user information and it requires a high security to ensure confidentiality, integrity and availability.

This paper uses an encryption and tokenization techniques to detect and prevent SQL injection attacks on web applications. Here tokenizing process converts the input query in to tokens and stored in tables and also detects spaces, single quotes and double dashes etc. Here it compares dynamic table with stored table, if tables are same, query is safe - proceed to execution in database for data retrieval, if tables are not same then it considers it as an attack and query is rejected.

III.TYPES OF SQL INJECTION ATTACKS

The There are different types of attacks, depending on the goal of attacker they are classified as follows:

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 6, June 2016

A. Tautologies**:**
This attack is used to bypass authentication procedure and used to inject code in conditional statements like WHERE condition. If the attack is successful, code will display all the records or at least one record is returned.
Ex. Tautology query to login is as follows:
SELECT * FORM user WHERE username = ‘ ‘ or 1=1 -- AND PASSWORD = ‘ ‘ ;

B. Illegal or logically incorrect queries:
In this type of attack attacker finds vulnerable parameters in the web application. Attacker makes use of error message sent by database server, which gives useful debugging information and give some information of database like table name filed names. By this information attacker can perform dangerous and more organized attacks.
Ex. Database will send error message for incorrect password input as follows:
SELECT * FORM employee WHERE Username = <uname> and password = <wrong password> or 1=1;

C. Union queries:
By using Union query technique attacker get information about other table form the database, by joining injected query to a safe query by word UNION.
Ex. Consider a query executed in the server side is of the following form
SELECT name, phone FROM Clients WHERE id=$id;
We will have following query after joining injected query using UNION as follows:
SELECT name, phone FROM Clients WHERE id =1 UNION ALL SELECT Credicard_num; FROM CreditCardTable

D. Piggy-Backed queries:
Here in this type of attack, additional queries are added in to the original query. Original query is not altered instead new query is inserted in the original query it is called as piggy-backed query. Attack intensions are extracting the data, modifying and executing remote commands.
The original query is executed first, additional query perform the attack on the database. Vulnerability lies in the configuration of database which allows executing multiple statements in single string. Database receives multiple SQL queries in a single query.

E. Alternate Encodings:
This attack is performed in conjunction with other attacks. Attack intension is to enable the techniques which allows attacker to avoid SQLI detection and prevention techniques.

Here query characters are replaced by some other characters like Unicode, ASCII, hexadecimals to hide all kinds of SQLI attacks. Databases parse comments out of an SQL statement prior to processing it, comments are often used in the middle of an attack to hide the attack’s pattern, thus avoiding detection by defensive coding practices and also many automated prevention techniques. / (Common defensive coding practices usually seen for characters such as single quotes and comment operators to bypass these defensive techniques they specially encoded strings such as hexadecimals, Unicode and ASCII characters hence these attacks are remain undetected.) Defence against alternate encoding is difficult to implement in real practice because developer has to consider all possible encodings

IV. PROPOSED WORK

In this proposed work we have implemented Aho-Corasick pattern matching algorithm, to detect and perevent SQL injections on Bank Application. Proposed architecture is shown in the figure 4.1.

It works as follows
A. Query is generated using username and password input provided by the client
B. Pattern matching algorith is applied to the generated query.
C. If pattern is not matched then allow the user to login else SQL Injection is found and alaram or notification is sent to the Administrator.

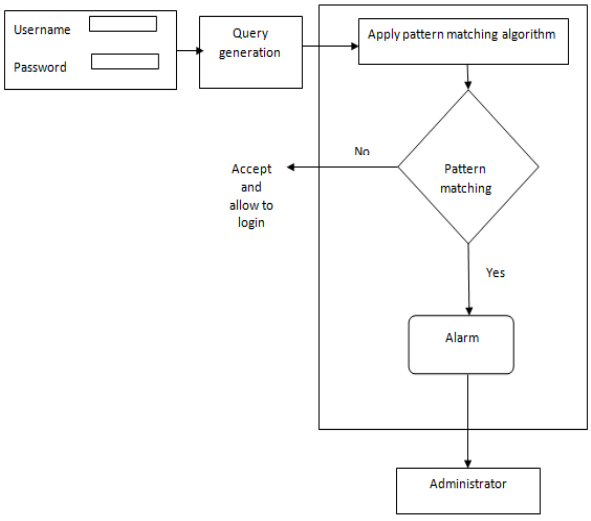

Figure 4.1Proposed Architecture

The Pattern matching algorithm of the proposed scheme is summarized as follows.

Pattern Matching Algorithm
Algorithm: Pattern Matching Algorithm
Input: user generated query
Output: Pattern matched or no

Step 1: User generated SQL query is tokenized and sent to the Pattern Matching algorithm.
Step 2: Pattern matching algorithm is given in the Figure 4.2
Step 3: User query is compared with stored pattern and Anomaly score is evaluated.
Step 4: If Anomaly score is greater than threshold value, pattern is matched and SQL Injection attack is detected. Notification is sent to the Administrator regarding attack on website.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 6, June 2016

Step 5: If Anomaly score is less than threshold value then query is accepted. User is allowed to access the website.

Static pattern matching Algorithm

1. Procedure SPMA(Query, SPL[])
   INPUT: Query <- User Generated Query
   SPL <- Static Pattern list with m Anomaly Pattern
2. For j=1 to m do
3. If(AC(Query, String.Length(Query), SPL[j][0]==Ø) Then
4. $Anomaly_{score} = \frac{Matching_{value}(Query, SPL[j]) \quad X\ 100}{String.Length(SPL[j])}$
5. If ($Anomaly_{score}$ > =$Threshold_{value}$)
6. Then
7. Return Alarm -> Administrator
8. Else
9. Return Query -> Accepted
10. End If
11. Else
12. Return Query -> Rejected
13. End For
    End Procedure

Figure 4.2 Pattern matching Algorithm

V. RESULTS

Website for Bank application is implemented using Java, Apache Tomcat and MySQL database to store customer information. Model proposed in [3] it detects and prevents only Tautology type of attack and [4] detects four different types of attacks; compared to these two our proposed model detects these following five types of attacks as shown in Table I.

Table I. Types of SQLI attacks prevented

SR NO.	SQL Injection Types	Proposed model Outcome
1.	Tautologies	Prevented
2.	Illegal or logically incorrect queries	Prevented
3.	Union queries	Prevented
4.	Piggy-Backed queries	Prevented
5.	Alternate Encodings	Prevented

VI. CONCLUSION

In this paper we have proposed Pattern matching technique to detect and prevent SQL Injection attacks on the websites. It successfully detects five types' attacks on websites and provides security to the websites. It is a useful application for data driven web applications, hence give security to websites. In future work it can be extended to cover all types' threats on websites.

REFERENCES

[1] Haeng Kon Kim "Framework for SQL Retrieval on Web Application Security" IMEQCS 2010, March 17.

[2] Ms. Zeinab Raveshi, Mrs. Sonali R. Idate "Efficient Method to Secure Web applications and Databases against SQL Injection Attacks" IJARCSSE Volume 3, Issue 5,May 2013

[3] R. Joseph Manoj, Dr.A.Chandrasekhar, M.D.Anto Praveena "An Approach to detect and Prevent Tautology Type SQL Injection in Web Service Based on XSchema validation" IJECS Volume 3 Issue 1, Jan 2014

[4] S.Anjugam, A.Murugan "Efficient Method for Preventing SQL Injection Attacks on Web Applications Using Encryption and Tokenization"

[5] Nida Khan, Abdul M.Siddiqu "Proposed Technique on 3-Tier Architecture for Developing SQL-Injection Attacks Proof Website "

[6] Zhendong Su, Gary Wassermann "The Essence of Command Injection Attacks in Web Applications"

[7] Kharche ,Jagdish patil, Kanchan Gohad, Bharti Ambetkar "PREVENTING SQL INJECTION ATTACK USING PATTERN MATCHING ALGORITHYM "

BIOGRAPHIES

Poornima Javali received B.E. degree in Computer Science Engineering form Visvesvaraya Technological University of Belgaum in 2013. Currently pursuing her M.Tech degree in Visvesvaraya Technological University of Belgaum. Her research interests include Web services security.

Mrs Sushma V Chaugule is working as Assistant professor in the Department of Computer Science and Engineering in KLE Dr M S Sheshgiri College of Engineering and Technology, Belgaum, Karnataka, India. She has obtained her BE in CSE in the year 2005 and M.Tech Digital Communication and Networking in the year 2007. Her areas of Research Interest are Image processing and Pattern Recognition, Document Image Analysis and Medical Image Processing. She has number of publications in peer reviewed International conferences.

DOI 10.17148/IJARCCE.2016.5630