Prevention of SQL Injection by Self Generating Triplet Patterns

Abhay K.Kolhe
Dept. Of Computer Engineering
MPSTME, SVKM'S NMIMS Mumbai, India
M.tech.Faculty

Pratik Adhikari
Dept. Of Computer Engineering
MPSTME, SVKM'S NMIMS
Mumbai, India
M.tech.student

ABSTRACT

The paper is focused with the new prevention method for SQL injection as it is always the top threat to any web site or web application. The paper focus generation of the pattern from the training query to prevent the SQL injection for the new query by forming triplet pattern. The new concept of matching pattern ratio has been introduced in the paper. In the last section advantages and the disadvantages of the method is discussed.

General Terms

Web security, SQL injection, prevention techniques, triplet pattern for SQL injection queries

Keywords

SQL injection, SQL injection vulnerability, web security, prevention of SQL injection

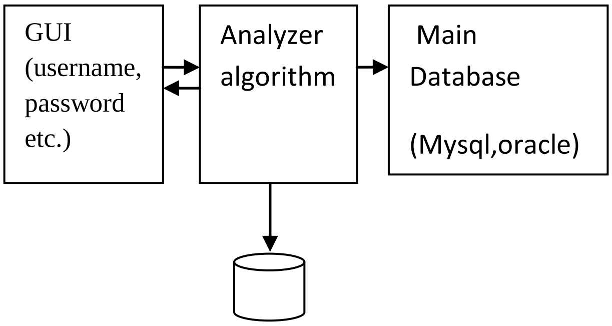
1. INTRODUCTION

The prevention of the SQL injection is always a matter of concern to any web applications and website. There are many methods that have been discussed to prevent the SQL injection like mySQL_real_escape_string(), mysqli , stored procedure , parameterized queries but newer attacks always defeat these prevention mechanism[1]-[5]. The aim of the paper is to prevent SQL injection through the training queries that generate the pattern which can be useful for the unknown query up to some extent although this method have the limitations that will be discussed at the later section of the paper.

2. ARCHITECTURE PROPOSED

The proposed GUI is the client side. The web server and the database are at the local machine only.

So detailed architecture is avoided in the Fig.1



Database storing patterns from training query

Fig.1 Proposed Architecture

2.1 GUI coding

GUI coding can be done by using html to have the fields for the username and password that can be given to the user. Attacker is able to inject SQL injections from these fields to get login access.

PHP/ASP any language can be used for server side coding.

2.2 Analyzer Algorithm

The analyzer algorithm consists of the training query that will analyze and generate the pattern, which can be useful of identifying the malicious query given by the attacker. The analyzer algorithm must run before the query is sent to the database.

2.3 Database

The database can be made from MySQL, Oracle MS SQL 2008 etc.

The functions of the database is same as tradition, it inserts the data and fetches the data to the user/attacker.

2.4 Database storing patterns from training query

In this database, older patterns from the training query are stored so the analyzer is able to check from its older history whether if the new query is malicious or not. Fig. 1 should be checking the patterns first before it sending the query to the main database.

3. WORKING

The working section is mainly divided in two parts, training query pattern generation without token and training query pattern generation with token [6].

3.1 Pattern generation without any tokens (Training Query)

The query is given by the user or the attacker and stored in the form of array to generate the patterns for the same query. This phase is called training phase of the method as huge number of patterns can be generated from the different queries and stored in the database.

3.2 Algorithm

1) Input the malicious query for the training, remove extra spaces if it in the query.

2) Calculate the string length using predefined string length function.

3) Break the strings into single character in the array of length of the string.

4) Form the series of triplets by joining the 3 array index.

5) Store these triplets in the database as patterns that can be used for the new query.

6) Input the new query, check whether the triplet pattern is present in the new query or not.

7) If it is present discard the query with appropriate error message else proceed to the main database.

3.3 Steps

1) **Training query**: 'or '1'='1 [7].

2) String length: 9

3) Breaking into array of single character shown in **fig. 2.**

'	o	r	'	1	'	=	'	1
0	1	2	3	4	5	6	7	8

Fig.2 Breaking strings into character for training query

4) Forming Series of triplet patterns for the training query is shown in **Table 1.**

Table 1. Triplets pattern for training query

id	Generate id	Values
0	0,1,2	'or
1	1,2,3	or'
2	2,3,4	r'1
3	3,4,5	'1'
4	4,5,6	1'=
5	5,6,7	'='
6	7,8,9	='1

5) Store the triplets into the database for triplets shown in **fig. 3.**It can be stored in the form of tables so it is easy to match.

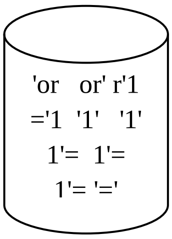


Fig.3 Database for triplets for training query

6) Input the new attacking query: ') or '1'='1-- ' [8].

Check for the pattern from the triplets using search function for the checking triplets.

7) If at least one match is found in new attacking query, it should be discarded

Else the query is given to the main database or other scanning or analyzer algorithms can be applied.

For explanation the new attacking query is broken into triplets to show the match results

3.4 Pattern generation and matching without tokens (attacking query)

Attacking query is the new query given by the attacker to exploit the database or to get login.

Attacking Query: ') or '1'='1-- '

Fig .4 shows the breaking of attacking query into character.

'	)	o	r	'	1	'	=	'	1	-	-	'	
0	1	2	3	4	5	6	7	8	9	10	11	12	

Fig.4 Breaking strings into character for attacking query

Table 2. Show forming of triplet pattern for the attacking query and matching with training triplet query [9].

Table 2. Triplets pattern for attacking query

Id	Generate id	Values	Match /Not Match
0	0,1,2	')o	Not match
1	1,2,3	)or	Not match
2	2,3,4	or'	Match with id 1 table 1
3	3,4,5	r '1	Match with id 2 table 1
4	4,5,6	'1'	Match with id 3 table 1
5	5,6,7	1' =	Match with id 4 table 1
6	6,7,8	'='	Match with id 5 table 1
7	7,8,9	='1	Match with id 6 table 1
8	8,9,10	' 1 -	Not match
9	10,11,12	--'	Not match

Matching pattern ratio: Number of triplets match
(MPR) Length of array attacking query
For this case: 6/13 = 0.461

In this method in any case if the matching pattern ratio is greater than zero, the query will be rejected.

3.5 Pattern generation with tokens (Training Query)

In these section triplets patterns are generated with the fixed tokens given by the developer.

Training Query: x' OR username LIKE '%admin%

Fixed Tokens: or, username, admin, ', %

Fig.5 shows the breaking of strings into character for Training query.

x	'	OR	username	LIKE	'	%	Admin	%
0	1	2	3	4	5	6	7	8

Fig.5 Breaking strings into character for Training query

The array length is calculated and the triplets for each fixed token are generated.

Triplet pattern for the training query for fixed token shown in the **Table 3**.

Table 3. Triplets pattern for training query with using tokens

id	Generate id	Values id
1	0,1,2	x' OR
2	1,2,3	'OR username
3	2,3,4	OR username like

4	3,4,5	Username like '
5	4,5,6	Like ' %
6	5,6,7	' % admin
7	6,7,8	% admin%

3.6 Pattern generation and matching with tokens (Attacking query)

Anything other than fixed token is considered as token for e.g

Attacking Query: asdfjv' OR username LIKE '%admin%.

Fixed Tokens: or, username, admin,' %,asdfjv

Due to unknown string "asdfjv" it is added in the fixed token shown in the **Fig 6.**

asdfjv	'	OR	username	like	%	admin	%
0	1	2	3	4	5	6	7

Fig.6 Breaking strings into character for attacking query

In this section triplets for the attacking query is generated that is matched with the training query, shown in the **table 4.**

Table 4. Triplets pattern for attacking query using tokens

id	Generate id	Values id	Match /Not Match
1	0,1,2	asdfjv' OR	Not match
2	1,2,3	' OR username	Match with id 2 table 3
3	2,3,4	OR username like	Match with id 3 table 3
4	3,4,5	Username like %	Not match
5	4,5,6	Like % admin	Not match
6	5,6,7	% admin %	Match with id 7 table 3

Matching pattern ratio **(MPR)**: $\frac{\text{Number of triplets match}}{\text{Length of array attacking query}}$

For this case: 3/8 = 0.375

In this method in any case if the matching pattern ratio is greater than zero, the query must be rejected.

4. ADVANTAGES OF THE METHOD

In this method huge number of the triplets pattern can be formed which can be used for blocking many new SQL injections based on the previous triplets patterns used. If in any case the attacking query bypass the analyzer algorithm, the triplets for the attacking query can be formed that can be used for blocking the attacking query and similar query related to that.

5. LIMITATIONS OF THE METHOD

Limitations for this method are that it is difficult to decide about the new token in the pattern generation with token method.

Also creation of the extra database for the triplet pattern can be limitation for additional usage of resources such as database.

The spaces in the query must be taken into care, it is also another problem if developer wishes to recreate query from the triplet pattern using machine.

6. CONCLUSION

The paper clearly shows that the SQL injection can be prevented by forming the triplet patterns from the training SQL injection query which will acts as a base for the attacking SQL injection query, through this method system can be made more robust than traditional approaches that have been used to counter SQL injection attacks.

7. REFERENCES

[1] Nontarak, S. Leelawat T, "Securely Web-Based Application for Construction Material Testing" International Journal of Computer Applications (0975 – 8887) Volume 42– No.11, March 2012 .

[2] http://php.net/manual/en/function.mySQL-real-escape-string.php 26 Nov 2013

[3] http://www.php.net/manual/en/mySQLi.quickstart.multiple-statement.php 26 Nov 2013

[4] Preshika Tiwari, Ashish Kumar Srivastava, " A Survey on Authentication Mechanism against SQL Injection in XML" International Journal of Computer Applications (0975 – 8887) Volume 78 – No.7, September 2013.

[5] Abhay K.Kolhe, Pratik Adhikari "Injection, Detection, Prevention of SQL Injection Attacks" International Journal of Computer Applications (0975 – 8887) Volume 87 – No.7, February 2014.

[6] Gaurav Shrivastava, Kshitij Pathak, "SQL Injection Attacks: Technique and Prevention Mechanism" International Journal of Computer Applications (0975 – 8887) Volume 69– No.7, May 2013.

[7] https://www.owasp.org/index.php/Testing_for_SQL_Injection_(OWASP-DV-005) 16th Feb 2014

[8] http://www.sqlinjectionwiki.com/Categories.aspx?catId=1 16th Feb 2014

[9] http://www.bbc.co.uk/bitesize/higher/biology/cell_biology/rna/revision/2/