Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol.6, No.2, 2015

SQL Injection Attacks - Its Prevention using Flag Sequencing Approach

Manveen Kaur
Department of Computer Science and Engineering, Amity University,Noida, Uttar Pradesh, India – 201301
B-68 Jalvayu Vihar , Gautam Budh Nagar, greater Noida, India
manveen77@gmail.com

Abstract
SQL injection attack is a code injection technique used to attack database through website. This happens when the user input is not correctly filtered for string literal escape characters which are present in the statement or when the user input is not strongly typed. In computer science, a type system is said to feature strong typing when it specifies one or more restrictions on how operations involving values of different data types can be intermixed.. [8]. SQL injection is one of the top ten web application attacks.In this paper a method is proposed in which two approaches, one static in which the database is created and another dynamic in which the query structure against the previously stored query structure is compared. If the two structures match then search is stopped and query is regarded as a valid query otherwise the query is an invalid query and is not allowed to access data from database. The Algorithm has been developed using Java.
Keywords: Malicious, Flag , Vulnerability**,** malicious, SQLIA's.

1 INTRODUCTION

SQL injection is very serious threat to web applications. In recent years, with the emergence of internet, databases have become even more important than ever before and are a critical part of network security. Database is the storage brain of a website. [1] A hacked database is the source for passwords and the other important information like credit card number, account number etc. Importance should be given for preventing database exploitation by SQL injection. [1] [6].Already some work has been done to prevent this attack. The methods used earlier are cumbersome, as they need to modify the source code which is an overhead, also it minimizes the runtime response time .In this approach there is no need to modify the source code and use of modern age processor architecture is done in a multithreaded way to minimize response time.

There are different types of web applications that are present at the server. To run the application, it must be properly configured first. The core part of any Web application is stored on the server site within the application server. The different browser supported languages like PHP, ASP, J2EE, Java/JSP, Perl, CGI contain the core in the form of software program.

As the J2EE web applications have good compatibility therefore they have been used in this work.

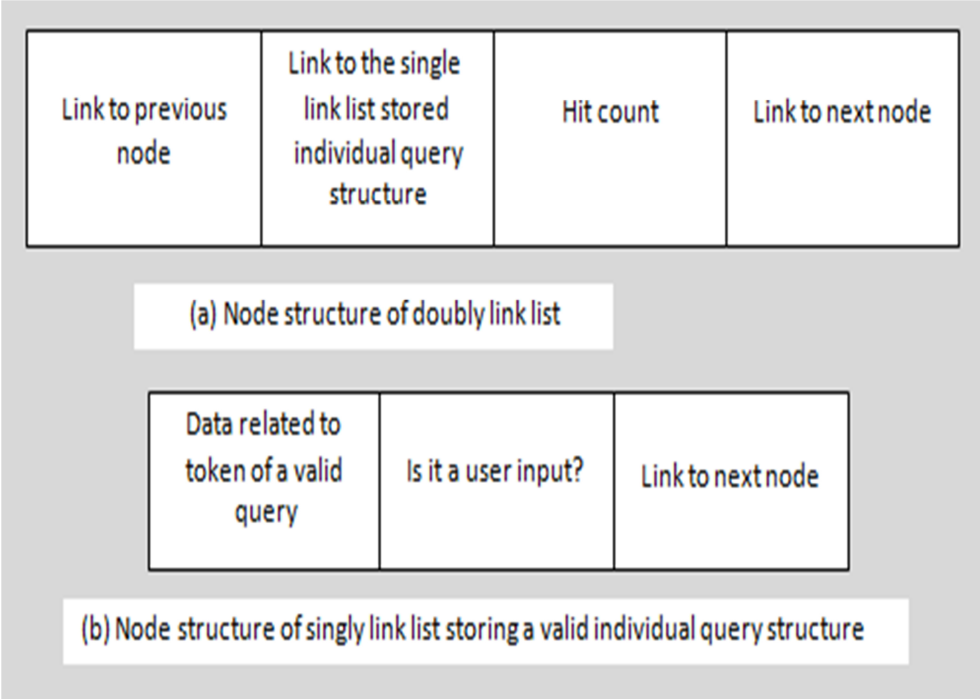

Fig. 1. Node structure

2 The Graphical representation of Linked lists

The figure below represents how Singly Linked Lists with same number of Flags are grouped together in a doubly Linked List. In order to keep track of all the doubly linked lists while searching, we store their reference in Array, that points to these Lists. This makes our Searching task simple. We have made the use of dynamic approach to searching, in order to make our search more efficient and time saving.

The doubly linked list that stores same number of Flags of all the singly linked lists is called a Group. Whenever an input query arrives, its structure is searched in the Group that corresponds to the same number of Flags. If the flags match, the query is regarded as a valid query otherwise it is invalid.

In this paper we provide a,

- Complete methodology and diagrams illustrating the SQLIA'S
- A technique for detection and prevention of SQLIA'S
- We make use of modern age processor architecture by using multithreading, inorder to reduce response time.

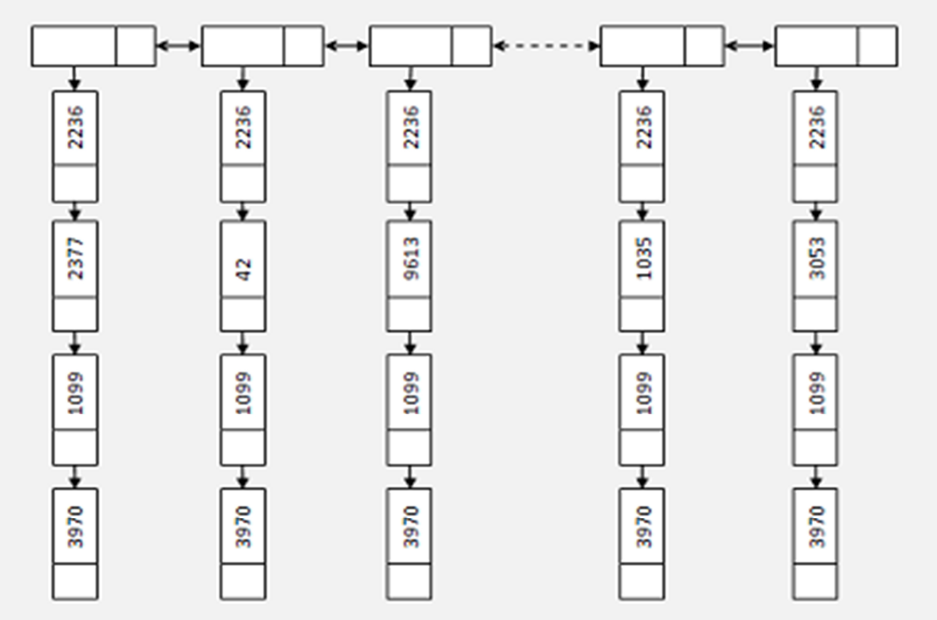

Fig. 2. Singly Linked Lists with same number of Flags

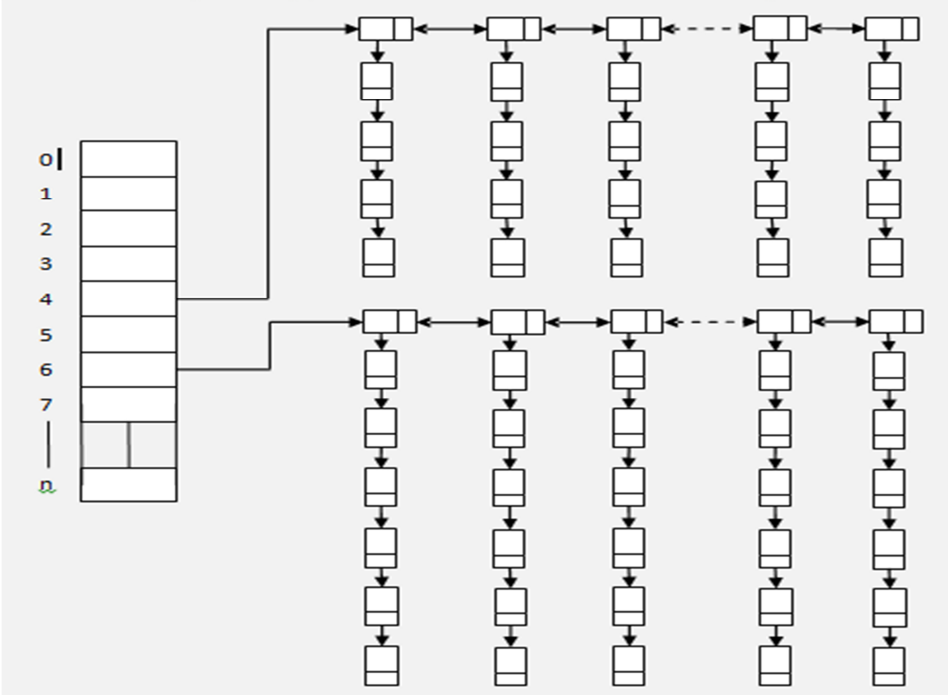

Fig. 3. Array pointing to Groups

3 CATEGORIES OF SQLIA

1. ***SQL manipulation*** - It means modifying SQL statements by the use of several operations like UNION. Another way is by changing Where clause to get different output.
2. ***Code Injection*** - IT is the technique of inserting new SQL statements into the vulnerable SQL statement Example Append EXECUTE command to the code.
3. ***Function call injection*** - Process of inserting numerous database function calls into SQL statement.
4. ***Buffer overflows*** - Caused by using function call injection. This attack is possible when Server is un – patched.

3.1 Sql Injection attacks examples

Suppose a user wants to access the email-id in the form "email me my password" form .[4] He writes the query

```
SELECT data FROM table WHERE Emailinput =
'$email-input';
```

The "$email-input" contains what the user types in the email address form field.[4] [2]

Suppose that the attacker knows that the database is vulnerable to attacks, he can type any malicious code on the form field to get more information.

```
Y'; UPDATE table SET email ='hacker @
Ymail.com' where email = 'Smith@Ymail.com';
```

Here the extra quote which is followed by semicolon that allows the close the statement & run another statement of his own. [2] [4]

When the malicious code is executed by the application,

```
SELECT data FROM table WHERE Emailinput = 'Y';
UPDATE table SET email = 'hacker @ Ymail.com'
WHERE email = 'Smith @ ymail.com';
```

This code resets the email- id of "Smith @Ymail.com" to "hacker @ Ymail .com".

Hacker now has a mail – id & password to the application which is under someone else account.

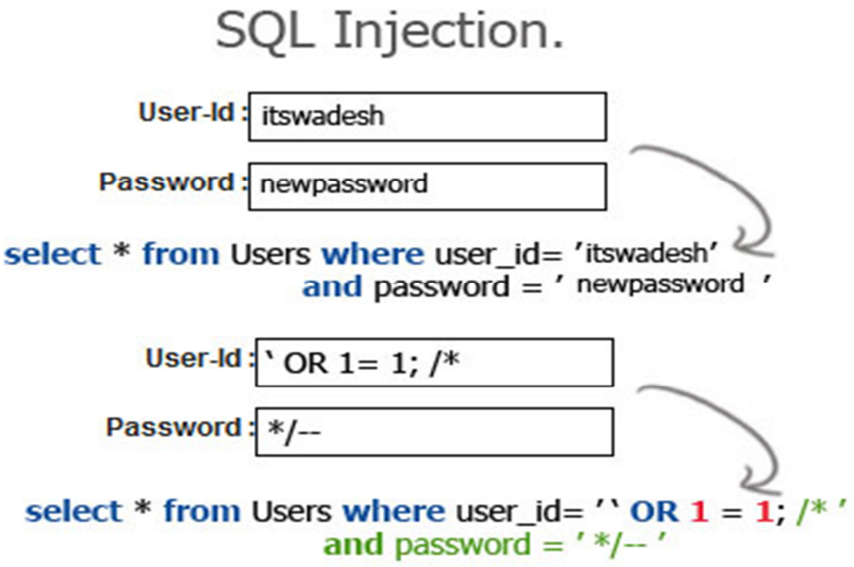

Fig. 4. An SQL Injection attack

4 How the web sites can be attacked

1. *Tautology*

Tautology means a formula that is true in every case. Here the code is injected in such a way that it always evaluates to true. If a condition is transformed to a tautology, it can return all the rows of the database. The main aim of the attacker is to make all the conditional statements evaluate to true by injecting code in one or more statements.

```
"SELECT * FROM student
WHERE name = ' " + request.getParameter("name")  +
" 'AND password =' " + request.getParameter("pass") +
" ' ' ";
```

```
SELECT * FROM Student

WHERE name = 'Amit'
AND Password = ' 'OR 'a' LIKE '%a%'
```

2. *End of Line comment*

```
SELECT * FROM Student
WHERE Stud_name  = 'admin'--'AND password = ''
```

3. *Logically Incorrect query*

Sometimes it is the intentional approach of the attacker to write incorrect query. The application server of the backend database returns error messages that contain enough information for the attacker to make good attempt.

4. *Union Query*

An attacker sometimes causes information to be derived from a desired table that was not actually intended by the developer. In order to do this, he first exploits the vulnerable parameter so that the dataset which is returned by the query can be changed. This is done by making use of UNION SELECT. [4] E.g.

```
SELECT * FROM Student
WHERE login" =
"UNION SELECT password FROM Stud_info
WHERE stud_name = 'Amit' – 'AND pass ="
```

The database will return column "Password" for stud_name "Amit".

5. *Piggy backed Query*

In this approach additional queries are added to the original query. There is no need to change the original query, instead more are added which "piggy- back" the original query [4]
E.g. If the user inputs "; DROP table student—

```
SELECT  Rollno  FROM Student
WHERE login = 'Amit' AND pass = ";
DROP table Student --'
```

Database will recognize (";") & and execute the second injected query. As a result table is dropped.

6. *System stored procedures*

System stored procedures are as vulnerable as other queries. In order to attack the database, attackers first determine the database and then make queries against the stored procedures. E.g.


```
CREATE  PROCEDURE   EMPLOYEE_SEARCH
 @EmpName  Varchar(200)  =  NULL AS
  DECLARE @Sql  nVarchar(2000)

   SELECT @Sql = 'SELECT
   EmpId, EmpName, Category, Price' + 'FROM Employee

  WHERE'   IF @Empname  IS NOT NULL
SELECT @Sql + 'EmpName  LIKE" ' + @EmpName  + " "EXEC(@sql)
```

Now if the input is 1' or'1' = '1'; exec mastr.dbo.xp_cmdshell 'dir'--, then the query executed at the server is [4]

```
SELECT   EmpId,  EmpName  FROM Employee
  WHERE EmpName LIKE '1' or '1' = '1';
EXEC master.dbo. xp_cmdshell 'dir'--'
```

It returns all the rows from table and executes command DIR.

7. Inference

In this technique when the injection attack becomes successful, no error message is returned by the database server. So therefore attacker has to insert commands in the website and observe the behaviour of that website. When the response of the website changes, he can deduce different values from the database.

8. Blind injection

In this type of attack, true/False questions are asked and information is derived from the behaviour of the page. If the result of injection is true, the site functions normally otherwise if it is false the page behaves differently.

9. Timings attacks

In this technique, the attacker uses If-then statement to derive information. The attacker makes note of the timing delays in the response of a database. E.g. WAITFOR keyword causes the database to delay its response by a specific time.

4.1 Some ways to avoid these attacks

1. Minimum use of dynamic SQL queries should be made if there is some alternate way.
2. The Stored procedure must be executed using a safe interface such as callable statements in JDBC or command object in ADO
3. All the input from the user must be validated thoroughly.
4. In order to run the database, use of low privilege account must be made.
5. Proper roles & privileges must be given to the stored procedure that is used in the application.
6. Use of parameterized stored procedures with embedded parameters must be made.

5 Algorithm

1. Input a query that probably contains the SQLIA code.
2. Convert the query into Flags & separate the Flags.
3. Convert the Flags into corresponding integer values.
4. Search the query structure that matches with the input query structure by using appropriate Searching technique.
5. If a match is found, search is successful and the query structure is a valid one
6. Else
7. The search is not successful and the query is regarded as invalid.
8. EXIT

6 Conclusion

Sql injection is a common technique that the hackers use to attack databases to extract their confidential information. These attacks have thus made it essential to develop methods to disable such attacks so that no invalid query can exploit the database. In this paper a method has been developed to differentiate between the valid queries and invalid queries. In the dynamic approach developed a query is first examined to see if it is

valid , if it is so by comparing its structure only then the query is allowed to execute. This is an efficient method as it minimizes the searching time and response time because searching is performed in a multithreaded way as well as it makes use of modern age processor. There are also some complexities involved such as FLAG separation, FLAG to integer conversion and the searching process in link list. Complexity of FLAGS to integer conversion is 0(n) where n is the total number of literals in all FLAGS of query.

References

[1]. http://en.wikipedia.org/wiki/SQL-injection
[2]. http://www.unixwiz.net/techtips/sql-injection.html
[3]. http://cwe.mitre.org/documents/vuln-trends.html
[4]. http://ferruh.mavituna.com/sql-injection-cheatsheet
[5]. Preventing sql injection attacks Using AMNESIA William G.J Halfond and Alessandro Orso Georgia Institute of Technology
http://www.cc.gatech.edu/orso/papers/halfond.orso.ICSEDEM006.Presentation.pdf
[6]. Silberchatz, Korth, Sudarshan Database system concepts, 4th edition.
[7]. The Secret of JAVA thread pools. Developer.amd.com/documentation/articles.
[8]http://en.wikipedia.org/wiki/strongly-typed-programming-language.
[9]http://msdn.microsoft.com/en-us/library/ms161953(v=sql.105).aspx

The IISTE is a pioneer in the Open-Access hosting service and academic event management. The aim of the firm is Accelerating Global Knowledge Sharing.

More information about the firm can be found on the homepage:
http://www.iiste.org

CALL FOR JOURNAL PAPERS

There are more than 30 peer-reviewed academic journals hosted under the hosting platform.

Prospective authors of journals can find the submission instruction on the following page: http://www.iiste.org/journals/ All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Paper version of the journals is also available upon request of readers and authors.

MORE RESOURCES

Book publication information: http://www.iiste.org/book/

Academic conference: http://www.iiste.org/conference/upcoming-conferences-call-for-paper/

IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digtial Library , NewJour, Google Scholar