# PREVENTION OF SQL INJECTION ATTACK IN WEB APPLICATION WITH HOST LANGUAGE

**Surabhi Agrawal[1], Upendra Singh[2]**

*[1, 2]Lecturer CSE&IT Department Government Engineering College Bharatpur (Raj.), India*

---

**Abstract:** In recent times, internet is widely used in every possible field. Applications used by internet and their databases contain data in secure form but still they can be attached ,SQLIA attacks which is one of the gruesome attack to steal the useful data information from database and malicious attackers will be able to get unauthorized access to useful data, by attacking the SQL queries ,websites , web applications . The malicious users aim to attack on the input validations by queries which is the most critical part of software security so to prevent input from gruesome attack is valuable and to prevent these types of attack and save our database we used PHP and Java as host language to prevent our query from these attacks.

**Keywords: - SQL injection, SQLIA, PHP, Java, Query**

## I. INTRODUCTION

As now days, the use of internet is spread on wide level in means of broadcasting information, and various online transactions, use of websites for any work , internet is used in every possible field. These applications used by internet and their databases also contain all the data in secure form but still they have sensitive data [1, 2]. The confidentiality of data in database is not proper and so many attacks can be performed on the database in which SQL Injection Attacks known as (SQLIA) is gruesome used attack and it targeted to access the data from database services. As website is used widely over the network, the malicious users attack the website in negative and harm the application. SQLIA consists of attack against web applications and has injection attacks in which attacker targets to modify the structure of SQL query [1, 3]. The changes imposed on SQL query which is placed by client but malicious user modify it by creating a well crafted data to username and password field of web browser without proper validation to get access to the SQL database and access the useful information by pretending the valid user [3].

SQL injection attacks consist of some methods of attacks to maliciously attack database. SQLIA can extract and read confidential or secure data from database or it can modify any data and can also execute some unreliable operations. The database result when expected from malicious database or queries gives incorrect or unsolicited result [1].

## II. TYPES OF SQL INJECTION ATTACKS

There are various methods of attacks present that are used by attacker to extract data and to make attack on queries these methods are classified as follows [1, 2, 3]:

1. **Tautology:** This method of SQL attack works on the injection of tokens to given query statement and it always considered is as true. This attack is mainly used by query with 'WHERE' clause. Query for login is [1, 2, 3]:

   "SELECT * FROM employee WHERE login ID = '423' and password = 'sss' OR '1' '=' 1'

   Here OR make this statement **as** tautology (1=1) and the query statement result is always true.

2. **Illegal/Logically Incorrect Queries**: When a query is given in extra and it is not needed, an error message is displayed from database including some debugging information. This message displayed sometimes useful for attacker to find vulnerabilities present in database [1, 2, 3].

   Example: SELECT *FROM users WHERE login='derived' AND password=password (select host from host)

   Here the data entered gives irrelevant result and which causes error by which attacker also comes to know about the backend applications which are used.

3. **Union Query:** In this type of attack the SQL gives permission to two queries to join and return as a one result. In union query attacks, an attacker exploits a given variable from query or it can add some queries by itself to infuse extra information and find out the data from database [1, 2, 3].

   **Example:**

   **Original query:** select acc_no from user where u_id='45'

   **Injected Query:** select acc_no from user where u_id='45' UNION select card_no from card_details where u_id='45'.

4. **Piggy – Backed queries:** There are many attacks which used in SQLIA in these on is piggy backed in which attacker tries to add malicious query by keeping original query as it is here query is modified by adding some new distinct data in query to extract some more information from database. Here, attackers used one delimiter to add in original

query such as ";", which used to append extra query [1, 2, 3].

**Example:**

SELECT name FROM users WHERE u_id='35' AND password="pass'; droptable users_.

5. **Stored Procedures**: In this type of attack, the attacker focuses on procedures that are stored in database system which is either user defined or default procedures. Stored procedure is just a code which contains some vulnerability such as buffer overflow [1, 2, 3].

6. **Inference**: In this form of attack, the attacker changes the behavior of database and this done by two techniques: blind injection, timing attacks. In BLIND injection type sometimes the data [1, 2, 3]

7. **Blind Injection**: The normal SQL injection attack is the procedure in which hacker know about the error message that is returned by database and rely on it. But in Blind injection attack the hacker need not to see the error message or rely on it so It is called blind injection attack. It described by two forms [1, 2, 3]:

**Content based blind injection attack**: in this the comparison of database queries is based on values 'TRUE' or 'FALSE'.

**Example**: SQL statement is

SELECT name, price FROM store_table WHERE id= '46'

Attacker manipulate it to

SELECT name, price FROM store_table WHERE id= '46' and 1=3

Here this will cause the value returned to be 'FALSE' and no items displayed on screen.

But when the query is

SELECT name, price FROM store_table WHERE id= '46' and 1=1

Here this will cause the value returned to be 'TRUE' and details of id=34 are shown. And this validates the value of page.

**Time based blind injection attack:** In this type of attack the attacker performs a time intensive operations in which mostly used time based command is "SLEEP"

SELECT name, price FROM store_table WHERE id= '46' and if(1=1, sleep(10) , false)

This query displayed the result which is delay by 10 seconds.

## III. PREVENTION OF SQL INJECTION ATTACKS USING HOST LANGUAGE PHP/JAVA

The SQLIA prevention methods consist of many methods from which data is to make secure by providing algorithms in SQL query. In all these methods one method is by using host language as PHP and Java which is used as a prevention method by inserting some statements to query so it is not accessed by any attacker.

### 1. PHP AS THE HOST LANGUAGE

PHP is a dynamic language which is used in every field now days. And it can also used to prevent SQL attacks [4].

Example: If we enter the query to extract a password for particular id or number by PHP

```
$_name =$_get ['username'];
$_query = "SELECT password FROM table WHERE name = '$_name'";
```

Here the attacker add new information to extract the user's information from database in which it adds some extra data as

```
$_name = "data' OR 1=1--'"
$_query = " SELECT password FROM table WHERE name = '$_name'";
```

Then the result obtained contains

```
$_query = " SELECT password FROM table WHERE name = 'data' OR 1=1--";
```

HERE this is valid query but it displays the whole passwords of query so it is injected by hacker.

To resolve this type of sql injection attacks PHP gives some

### A. prepared statement :

```
$name = $_get['username']
If ($ stmt = $mysqli -> prepare("SELECT password FROM table WHERE name = '$_name'")){
$stmt->mysqli_bind_param("s",$name);
$stmt->execute();
$stmt->bind_param($pass);
$stmt->fetch();
Printf( "password for user %s is %s \n", $name ,$pass);
$stmt->close();  }
```

### B. Escaping string:

```
<?php
$username = mysqli_real_escape_string($conn,$_post["username"]);
```

```
$username = mysqli_real_escape_string($conn,$_post["password"]);
mysqli_close($conn);
?>
```

### C. Using trim() tags:

```
<?php
$username = strip_tags(trim($_post["usernsame"]));
$username strip_tags(trim($_post["password"]));
mysqli_close($conn);
?>
```

## 2. JAVA AS THE HOST LANGUAGE

Using java as host language the sql injection attacks can be secured by using prepared statements or escape operation or other methods by which data can be secured.

**Example**: Suppose a system wants to succeed or extract some information from database if query returns any result then authentication succeeds otherwise it fails:

SELECT *FROM emp_table WHERE username= '<username>' AND password='<password>'

And in between an attacker inserts some operation to extract more information from database i.e.

SELECT *FROM emp_table WHERE username= '<valid>' OR '1'='1 AND password='<PASSWORD>'

Here '1'='1 always returns value true which gives result of all rows displays on screen .

To remove this in Java we can use prepared statements

### A. Using prepared statement

If we use

```
preparedStatement= "SELECT * FROM emp_table WHERE username = ?";

preparedStatement.setString(1, valid);
```

After this we will be safe and hacker not able to extract our information from backend.
The prepared statement works on the principle that by using this we can force the user input to be handled as content of a parameter not as the sql command part. But there is a limitation in using prepared statement it works when we do not build our sql command by concatenate two strings here our query is vulnerable to sql attacks [5].

### B. Using callableStatement

By using callable statement we can also prevent our query from attackers to get information from our database illegally. It is used as

```
public interface callableStatement
extends PreparedStatement
```

The API of java provides SQL Escape command which allows all the stored procedures to be called in a standard way for all RDBMS. And this syntax is used as syntax of one parameter [6].

## IV. CONCLUSION

SQLIA is an attacking technique here malicious users input well crafted data to username and password field without proper validation. In this paper the various methods for SQLIA prevention are discussed by using PHP and Java as host language. But with these methods also there is no complete security provided for database.

## V. REFERENCES

[1]. IndraniBalasundaram, Dr. E. Ramaraj "An Approach to Detect and Prevent SQL Injection Attacks in Database Using Web Service" International Journal of Computer Science and Network Security, VOL.11 No.1, January 2011, p197-205.

[2]. Rashmi Yeole, Shubhangi Ninawe, Payal Dhore, Prof. P. U. Tembhare " A Study on Detection and Prevention of SQL Injection Attack" International Research Journal of Engineering and Technology Volume: 04 Issue: 3 | Mar -2017 p 435-438.

[3]. Udit Agarwal, Monika Saxena, Kuldeep Singh Rana "A Survey of SQL Injection Attacks" International Journal of Advanced Research in Computer Science and Software Engineering Volume 5, Issue 3, March 2015 p – 286-289.

[4]. PHP Manual, The PHP Group, [Available online] at http://php.net/manual/en/security.database.sql-injection.php [Accessed] on 15 November 2017.

[5]. Java Documentation, Oracle Inc., [Available online] at https://docs.oracle.com/javase/tutorial/jdbc/basics/prepared.html [Accessed] on 15 November 2017.

[6]. Java Documentation, Oracle Inc., [Available online] at https://docs.oracle.com/javase/7/docs/api/java/sql/CallableStatement.html [Accessed] on 15 November 2017.