Injection, Detection, Prevention of SQL Injection Attacks

Abhay K.Kolhe
Dept. Of Computer Engineering
MPSTME, NMIMS University
Mumbai, India
(Faculty)

Pratik Adhikari
Dept. Of Computer Engineering
MPSTME, NMIMS University
Mumbai, India
M.tech(student)

ABSTRACT

SQL injections have been always the top most priority for any website and web application. Every web application and website developed in php, asp.net, jsp which is connected to the database like MySQL, Microsoft SQL Server, and oracle are prone to SQL injection attacks. Most of the websites are created by using open source language such as php. The paper focuses the types of SQL injection attacks on the open source database in MySQL .The aim is to create a dummy web site where users can login and register. The attacker can login these dummy website using different types of SQL injection, make changes in the database, detect these types of attacks using IP tracking methods with their injection types and to prevent them.

General Terms

Web security, SQL injection, detection and prevention techniques.

Keywords

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

1. INTRODUCTION

Web security now a day is a major concern. Every web application, website is vulnerable to many attacks. The Majority of the web application and web site are made using PHP and MySQL as it is the open source for developing them. Open Web Application Security Project (OWSAP) has listed out the SQL injections as the top 10 vulnerability [1].The goal of the paper is to find out the vulnerabilities related to SQL injection, provide the detection mechanism and prevention techniques for these attacks. The aim is to create a dummy website to perform the various kinds SQL injection attacks to get authentication, perform the variety of operations related to insert, update, delete, and drop. It also detects the IP address of the attacker for the particular type of attacks and provides the prevention method for the same.

1.1 SQL injection attacks

The SQL injection is the code injection technique, in which malicious SQL statements are inserted into an entry field for execution and used to attack the database and performs different types of the database interaction, operations and functions without sanitizing the inputs in the entry field [2].

The SQL injection generally fools the database as a regular query by the user and gets access to the system easily [3].The paper is overall divided into three phase. Phases I is an injection phase where it will explain various types of SQL injection attacks, phase II deals with the detection of the phase I injection attacks with IP tracking mechanism and last phase III deals with the prevention mechanism.

2. PHASE I (Injection Phase)

This part gives the information about injection of the SQL queries for login, insert, update, drop and delete.

2.1 SQL Injection for Login

This part is divided into two sections first part is injection for the first row user and another is an injection for the particular user such as admin.

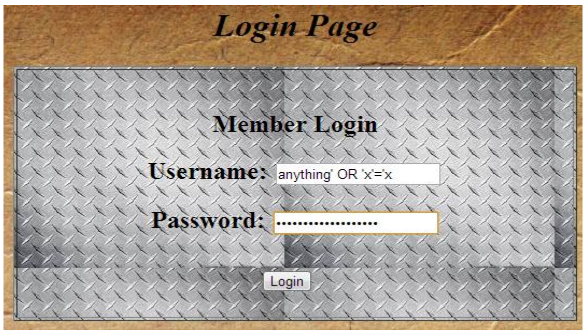
2.1.1.1 Injection for 1st row user

An Attacker is able to get login into the website successfully by giving username: anything' OR 'x'='x and password: anything' OR 'x'='x as the following shown in the **Fig 1**. This is the type of SQL injections is for incorrectly filtered escape character. Later in the prevention phase this paper will discuss how it can be prevented. The table name as the “user” is made where the values of all registered users are stored.

Injection query can be explained as:

"SELECT * FROM user WHERE name = '" + userName + "';"

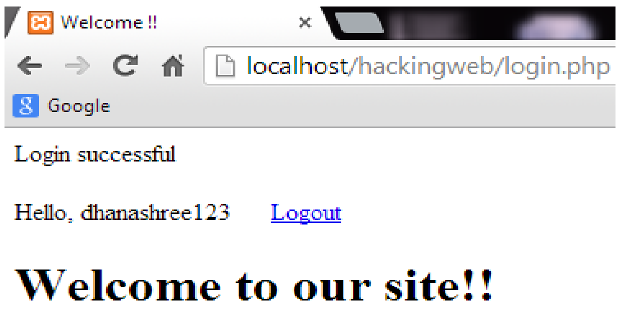
If the "userName" is replaced by SQL string, like anything' or 'x'='x during authentication, the database responds to the code in the same manner as the first code and displays the records. This is because the evaluation of 'x'='x' has been always true. So the attacker gets authenticated without any proper valid user name and password in the database. Here in this case single quote always allows the inside quote to get executed [3].



Username: anything' OR 'x'='x
Password: ....

Fig.1 SQL injection 1st row user

It can be seen in the **Fig.2** that the attacker is able to login to the website without any username and password. In this case the row for the first user is always getting selected for the login on the website. This form of the SQL injection is easy to check on any website. Most of the coders and developers forget to filter the escape character and this attack is always a vulnerability in PHP as well as asp.net for both MySQL and MS SQL server 2005, 2008.Consider the situation where the confidential medical reports can be leaked to the attacker where the attacker is not even registered in the database.



Login successful

Hello, dhanashree123 Logout

Welcome to our site!!

Fig 2.Results for injection 1st row user

2.1.1.2 Injection for particular user such as admin

Sometimes administrative rights such as admin are given to the particular user to maintain the database and website. So the attackers try different methods of trial and error techniques to get login using administrative privileges. **Fig.3** shows SQL injection for a particular user**.**

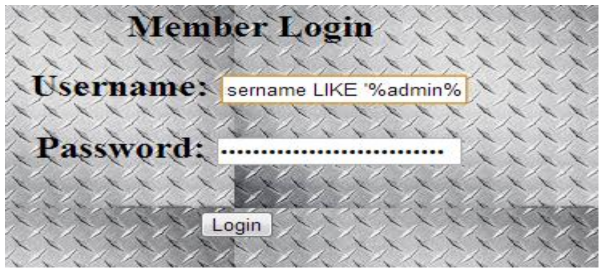
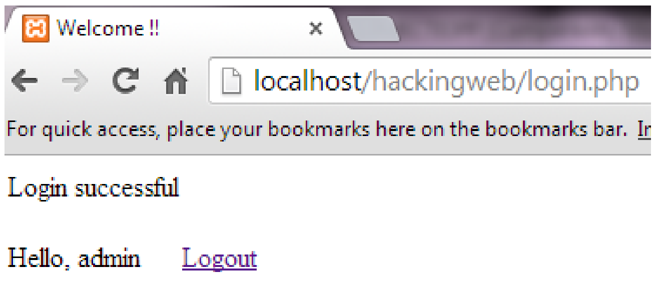


Fig.3 SQL injection for particular user


Welcome !!
localhost/hackingweb/login.php
For quick access, place your bookmarks here on the bookmarks bar.

Login successful

Hello, admin Logout

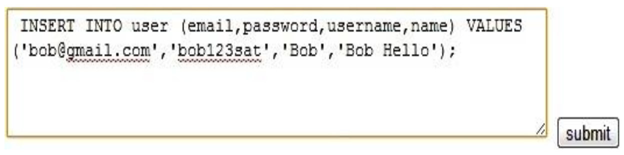
Fig.4 Results of injection

In this case, such as admin, the attacker uses the username as “admin” and can login as x' OR username LIKE '%admin%. The attacker supply both query to our login details such as username: x' OR username LIKE '%admin% and Password: x' OR username LIKE '%admin%.

When attackers enter a “LIKE” clause with the username then the database will return the matching criteria to the user immediately. SELECT username, password, email, full_name FROM user WHERE username=‘x‘OR full_name LIKE _%admin%*‘;* Here, the database will return information of any user where the name starts with “admin”. **Fig.3** shows the injection for the particular user such as “admin” and **Fig.4** shows the login results. “%” is the wild character used to select a particular user from the database.

2.1.3 Injection for insert operation.

Here the attacker is injecting the insert query to the database as username, password, email, and the name for the particular user. **Fig.5** shows the injection for insert query.

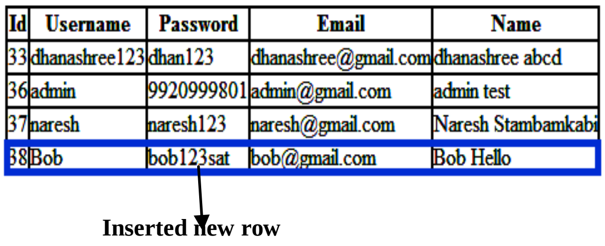


```
INSERT INTO user (email,password,username,name) VALUES
('bob@gmail.com','bob123sat','Bob','Bob Hello');
```

Fig.5 SQL Injection for Insert Query

Table 1 shows the insert operation results. It can be seen that email, password, username and name is added in the table name user.

Table 1.Insert operation

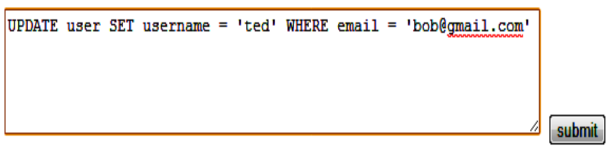


Id	Username	Password	Email	Name
33	dhanashree123	dhan123	dhanashree@gmail.com	dhanashree abcd
36	admin	9920999801	admin@gmail.com	admin test
37	naresh	naresh123	naresh@gmail.com	Naresh Stambamkabi
38	Bob	bob123sat	bob@gmail.com	Bob Hello

Inserted new row

2.1.4 Injection for update

The attacker is injecting the update query to the database and updating the username for the particular email address. **Fig.6** shows the injection for the update query.

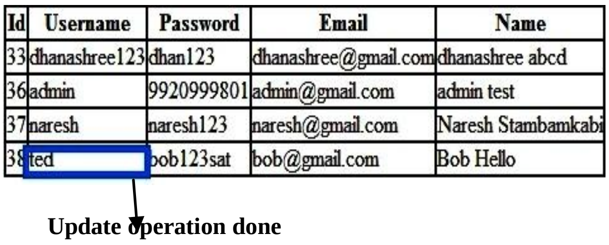


```
UPDATE user SET username = 'ted' WHERE email = 'bob@gmail.com'
```

Fig.6 SQL Injection for Update Query

Table 2 shows the update results, it can be seen clearly in the result that the username for the id 38 have been updated from “bob” to “ted”

Table 2.Update operation

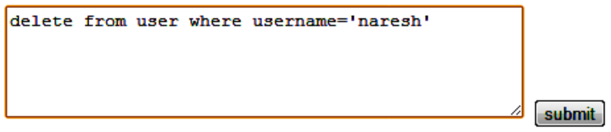


Id	Username	Password	Email	Name
33	dhanashree123	dhan123	dhanashree@gmail.com	dhanashree abcd
36	admin	9920999801	admin@gmail.com	admin test
37	naresh	naresh123	naresh@gmail.com	Naresh Stambamkabi
38	ted	bob123sat	bob@gmail.com	Bob Hello

Update operation done

2.1.5 Injection for delete

The attacker is injecting the delete query to the database and he is deleting the particular row in the database based on the username. **Fig. 7** shows injection for the delete query.



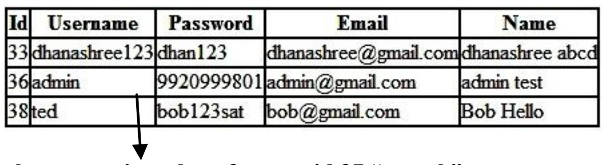
```
delete from user where username='naresh'
```

submit

Fig.7 SQL Injection for Delete Query

Table 3 shows the delete results .It can be seen clearly that the username "naresh" with id "37" has been deleted successfully.

Table 3. Delete operation



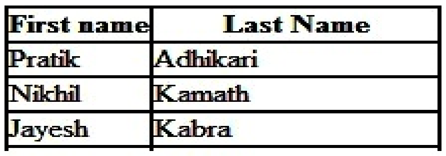
Id	Username	Password	Email	Name
33	dhanashree123	dhan123	dhanashree@gmail.com	dhanashree abcd
36	admin	9920999801	admin@gmail.com	admin test
38	ted	bob123sat	bob@gmail.com	Bob Hello

Delete operations done for user id 37 "naresh"

2.1.6 Injection for Drop

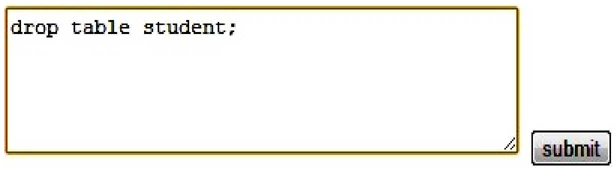
To inject the drop query extra table named "student" was made for the first name and last name so the main table "user" doesn't get altered. **Table 4** shows the table name "student".

Table 4.Student Table



First name	Last Name
Pratik	Adhikari
Nikhil	Kamath
Jayesh	Kabra

The attacker is injecting the drop query and is dropping the table name "student". **Fig.8** shows the injection for the drop query.

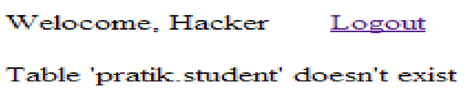


```
drop table student;
```

submit

Fig.8 SQL Injection for Drop Query

Fig.9 shows the results after dropping the table.



Welocome, Hacker Logout

Table 'pratik.student' doesn't exist

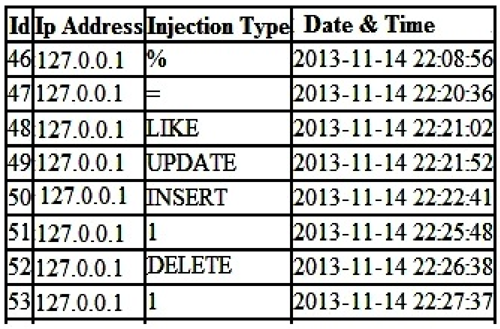
Fig.9 SQL Injection for Drop Query

The message is made using PHP and HTML coding after dropping the table this message appears as results, same table can be checked in the GUI for the database.

3. DETECTION BASED ON THE IP ADDRESS

For the detection part, the array has been made where the types of the operations that can be performed are listed out and the injection query is match out to an array. It can track down the list of operations that can be performed. The IP address and particular date and time are recorded here in the table. **Table 5** shows the Detection Based on the IP address with date and time recorded [4].

Table 5. Detection Based on the IP address.



Id	Ip Address	Injection Type	Date & Time
46	127.0.0.1	%	2013-11-14 22:08:56
47	127.0.0.1	=	2013-11-14 22:20:36
48	127.0.0.1	LIKE	2013-11-14 22:21:02
49	127.0.0.1	UPDATE	2013-11-14 22:21:52
50	127.0.0.1	INSERT	2013-11-14 22:22:41
51	127.0.0.1	1	2013-11-14 22:25:48
52	127.0.0.1	DELETE	2013-11-14 22:26:38
53	127.0.0.1	1	2013-11-14 22:27:37

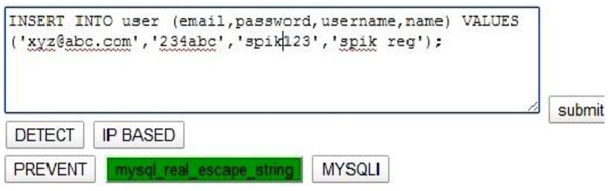
4. PREVENTION TECHNIQUE FOR SQL INJECTION

This section deals the prevention mechanism for SQL injection using **Escape routine** and **MySQLi.**

4.1 mySQL_real_escape_string()

Escapes special characters in the unescaped_string, taking into account the current character set of the connection so that it is safe to place it in a mySQL_query()[5].

mySQL_real_escape_string() calls MySQL's library function mySQL_real_escape_string, which adds backslashes to the following characters: **\x*00***, ***n***, ***r***, \, ', " and ***x1a***. The query is filtered out before it is passed to the database. So attacker is unable to inject the query after using this mechanism [5]. It can be seen in **Fig.10** an on-click event was made,on clicking on the on-click event "mySQL_real_escape_string" it changes its color to green and gets activated.

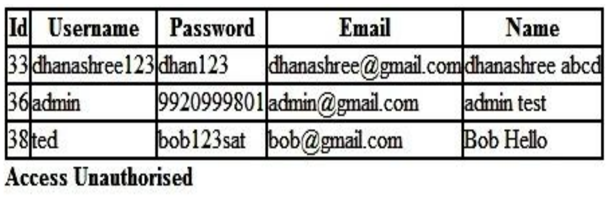


```
INSERT INTO user (email,password,username,name) VALUES
('xyz@abc.com','234abc','spik123','spik reg');
```

Fig.10 mySQL_real_escape_string()

It can be seen in the **Table 6** that it is not possible to a insert new query after applying **mySQL_real_escape_string** so it get prevented.

Table 6.Using mySQL_real_escape_string()



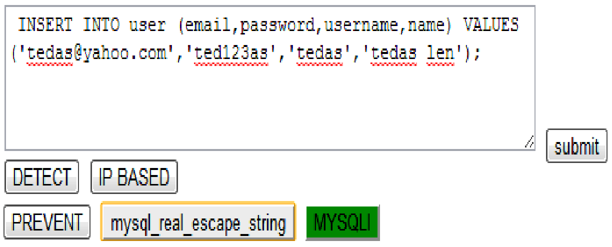
Id	Username	Password	Email	Name
33	dhanashree123	dhan123	dhanashree@gmail.com	dhanashree abcd
36	admin	9920999801	admin@gmail.com	admin test
38	ted	bob123sat	bob@gmail.com	Bob Hello

Access Unauthorised

4.2 MySQLi

MySQLi optionally allows having multiple statements in one statement string [6].

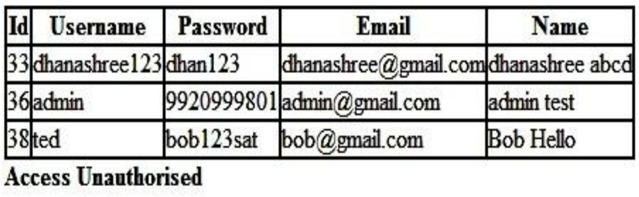
Multiple statements or multi queries must be executed with mySQLi_multi_query(). The individual statements of the statement string are separated by semicolon. Then all the result sets returned by the executed statements must be fetched. The MySQL server allows having statements that do return result sets and statements that do not return result sets in one multiple statements [6]. An extra API call is used for multiple statements to reduce the likeliness of SQL injection attacks [6].It can be seen in **Fig.11**



```
INSERT INTO user (email,password,username,name) VALUES
('tedas@yahoo.com','ted123as','tedas','tedas len');
```

Fig.11 MYSQLI

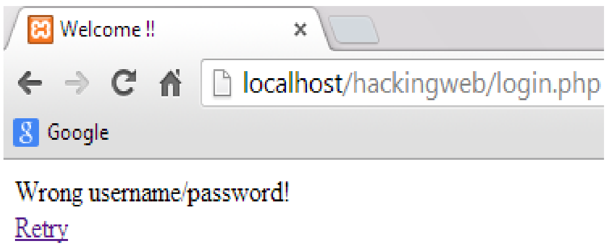
Table 7.Using MYSQLI



Id	Username	Password	Email	Name
33	dhanashree123	dhan123	dhanashree@gmail.com	dhanashree abcd
36	admin	9920999801	admin@gmail.com	admin test
38	ted	bob123sat	bob@gmail.com	Bob Hello

Access Unauthorised

It can be seen in the **Fig.12** by applying the preventing techniques, login error is displayed. So injecting both the queries for login is prevented and mySQL_error() is used, so the MySQL database on backend no longer issues warnings, which can give the valuable information to the attacker[7].



Wrong username/password!
Retry

Fig.12 Login Page Error

5. CONCLUSION AND FURTHER WORK

SQL injection has been always a topmost threat to any website or web application. Any web application or website that is developed PHP ,asp, jsp with database Mysql, MS SQL server, Oracle is prone to SQL injection[8][9]. This paper provides the different types of SQL injection attacks (php and database Mysql) for e.g login without valid credentials and performing various operations to the database like insert, updates, delete, and drop. These attacks have been detected using IP tracking mechanism. **MSQLi** and **mySQL_real_escape_string()** technique have been used for prevention. Thus injection, detection and prevention of SQL injection attacks in this paper have been done.

6. REFRENCES

[1] https://www.owasp.org/index.php/Top_10_2013-Top_10 25th Nov 2013.

[2] http://technet.microsoft.com/enus/library/ms161953(v=SQL.105).aspx 25th Nov 2013.

[3] Ramakanth Dorai,Vinod Kannan, "SQL Injection- Database Attack Revolution and Prevention", Journal of International Commercial Law and TechnologyVol6, Issue 4 (2011).

[4] Perumalsamy Ramasamy, Dr.Sunitha Abburu,"SQL INJECTION ATTACK DETECTION AND PREVENTION" International Journal of Engineering Science and Technology (IJEST) ISSN: 0975-5462 Vol. 4 No.04 April 2012

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

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

[7] http://us1.php.net/mySQL_error 26 Nov 2013

[8] Sid Ansari,Edward R. Sykes ,"SQL Injection in Oracle: An exploration of vulnerabilities", International Journal on Computer Science and Engineering (IJCSE) ISSN :0975-3397 Vol. 4 No. 04 April 2012 522.

[9] Zeinab Raveshi, Sonali R. Idate," Investigation and Analysis of SQL Injection Attacks on Web Applications: Survey", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-2, Issue-3 February 2013.