# Detection and Prevention of Web Application from SQL Injection: A Study, Survey and Analysis

**Tapan kumar***
School of Computer Science
Lingaya's University ,
Haryana, India
tapanjha91@hotmail.com

**Sital Methwani****
School of Computer Science
Lingaya's University
Haryana, India
sitalmethwani@gmail.com

**Daman Deep Singh*****
School of Computer Science
Lingaya's University
Haryana, India
damandeepsingh77@gmail.com

**ABSTRACT:** *The goal of this paper is to build an automated fix generation method to prevent SQL injection vulnerability from plain text SQL statements. In an automated method approach, a server will gather information about previously known vulnerabilities, specifically SQL statements, generate a patch, and apply patch. The process can be completed by someone with no security expertise and secure legacy code, which will allow developers to fix the SQL injection vulnerability.*

*Database services having interactive web applications targeted by an SQL Injection. Data as a input is given by the user and that input is used as to form SQL statement at runtime in these applications. An attacker might input a malicious or harmful query segment when user inputs any SQL statement during SQL Injection attacks, which could result in a different database request. Sensitive/Confidential information can be add or modify by an attacker by SQL Injection attacks. SQL Injection vulnerability can also be used by an attacker as a rudimentary IP scanner. There are several paper published in literature having discussed that how to prevent SQL Injection attacks in the database at runtime, by examining dynamic SQL query semantics. However, for secure stored procedures in the database layer / application layer a very less attention is given, which can also be suffer from SQL Injection attacks.*

**KEYWORDS: SQL QUERY, IP SCANNER, DATABASE, SQL INJECTION, ATTACKS.**

## 1. INTRODUCTION

Structured Query Language injection is one of the most challenging fact to impact on the business because it can explore all of the sensitive information which is stored in our database, including most highly important information such as credit card details, usernames, passwords, names, address, phone, email id etc.[1]

Structured Query Language injection is the liability that when attacker gets the ability with sql queries which is passes to a back-end database. The query which is passed by the attacker to the database, the attacker can allows the query to database which is supporting element with database and our operating system. Any Sql Query that accepts the inputs from the attacker sides can harms our real web application.SQL queries in relational database tables mostly.[15]

Attacker try to inserted malicious SQL query into an entry field for execution the query so that they can dump the database or alter the database i.e. this technique is called code injection technique. So this type of attacker is also called attack vector for websites and this type of attacker is used by any type of SQL database. [2]

According the study last year , security company Imperva find that the most web application is attacked 4 times per month and other side retailers company is attacked by 2 times per month. That is not a good sign on the behalf of security.

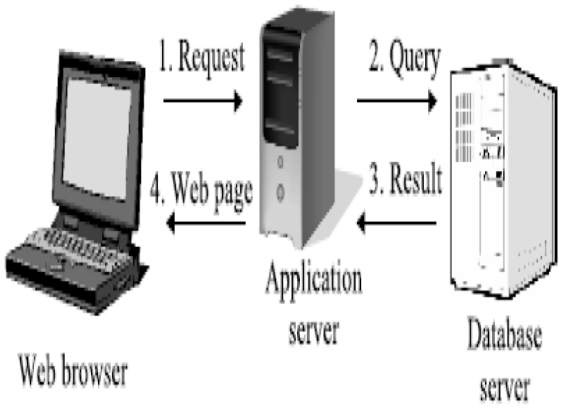



Fig: Client server architecture[10]

*1) SQL Injection Attack Steps:*

- SQL injection is a process to find the query which is enter by the user for execution the command.
- SQL attackers create crafted input data so that SQL interpreter have to accept the query and give the permission to execute the commands and give his desired results.
- SQL Injection attack breaks the security in the database layer. When attacker breaks the SQL injection flaw then attackers can drop , modify , create ,alter our sensitive database.

**2. LITERATURE REVIEW**

Our web applications allow the visitors to enter or submit or retrieve database using any web browser through the internet. These type of data have to centralized so that they can store data which is needed for websites. If any Suppliers , Employees , a host of stakeholders , customers etc. want to achieve specific content from database side then he can receive it .Company statistics , User Details , Financial and payment information etc are stored in our database which is access through custom web applications. Our Database and Web applications allow us to run our business regularly.[6]

The Process which attempts to pass commands or statements of SQL for execution by the database over the web application is called SQL Injection hacking technique. If their attempts are right then our database allow hackers to view their desire information from the database and he can hampers our database. And can do everything which he want.[7]

For example Like feedback forms , Shopping carts , Search pages , product and support request forms, shape modern websites , provide businesses and login pages etc pages are very necessary to communicate with customers for keep our customer in touch. [8]These types of pages of websites are very to use customer. These types are pages are suspicious for SQL hackers and first of all they attempts to try on these pages .We can not hide these type of pages on website. If we do it then our client cannot be touch with us. So hacking the website is becoming very easy task for Hackers.

A database is a collection of logically related tables and Distributed Database is a huge collection of such databases distributed over a computer network.[14]

*B. **For Simple Example***

For accessing the database , normal user would input their username and password to enter their profile and access his personal details and change the contents which is allow by the administrative section i.e. authenticate user are allowed to access our database. [4] In other sides, our web application which controls the authentication page will first of all communicate with our database through the specific planned commands so that they can filter that he is authenticate user or not. In the case of valid user , database allow to access the contents.

In other sides, In case of SQL Injection, Specifically craft SQL commands with the intent of passing the login form difficulty is inputted by the hacker. If client inputs are not properly sanitized then commands are directly executed and hacker enter our database. And in case of SQL Injection

vulnerabilities, Hackers are eligible to communicate with our database directly.

Dynamic script languages like PHP , JSP, ASP.NET . CGI etc are the target technologies by the hacker. By knowledge of SQL queries and creative guess the attacker attempts to hack the database in a web browser. For publicity, our website needs to be public so our security mechanisms will allow to be public with our application/s (generally over port 80/443).

**3. PROBLEM DOMAIN**

With over 20% of all web vulnerabilities that is being Related to sql injection, this is the most prevalent type Of application security and also the second most common Software vulnerability which have the find and prevent Capability .sql injection always must be on high Priority for web developer and also for security basis. Generally a sql injection attack diminishes any web Application software which has not provided a proper Validation or we can say coded by a user given input Data. [9] At last that input is used as a part of query again Any back-end database. Take example, when we create Any form it asks for id. Url:http://www.somewebsite.com/id/id.asp?id=someda Ta is created after that. An attacker who is using sql injection may enter any Data or "1=1". At the particular time if the application Software of the web is not given proper validation or Incorrectly encoded the user given data that is directly Send to the database, and as well as input with the Vulnerable query will reach there in reply that will Expose all ids in the database since the condition "1=1" Is always true. The example given is a basic example but It illustrates the importance of sanitizing user input data Before using it in a sql query or sql commands.

**4. SQL INJECTION IMPACT**

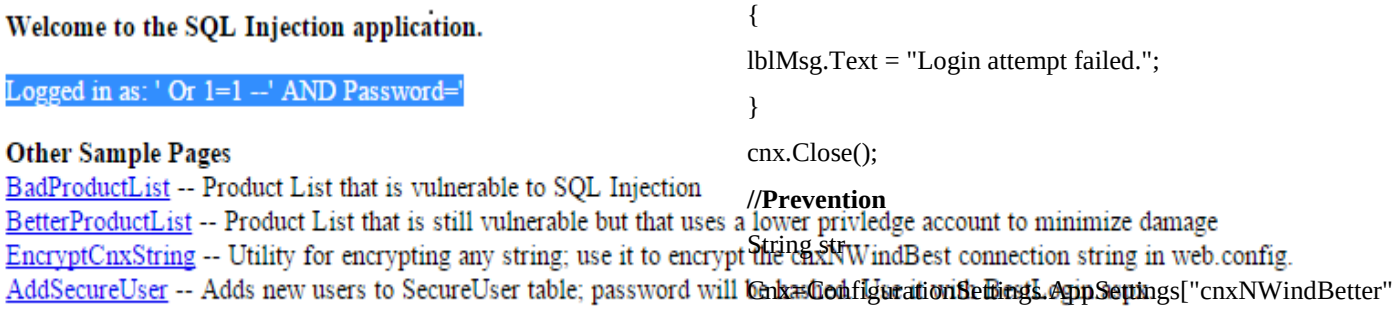
When a hacker feels that a system is ready to sql Injection attacks, he is now able to insert sql query/ Commands to the database through an input from field. This is similarly like as to say attacker comes to make Changes in our database and allowing him to do insert or Delete like drop to the database. Execution of arbitrary sql statements on the vulnerable System may be done by an attacker. This may break the Integrity of your system database and/or exposure of all sensitive information's. It depends on the back-end Database, sql injection vulnerabilities can be lead to Varying levels of data/system access to the attacker. [7] Manipulate in any existing queries, to union that is Used to select related information from two tables use Sub-selects arbitrary data, or append additional queries. Some of the sql servers like Microsoft sql server Contains stored and extended procedures for database Server functions. In certain cases, it can be possible to Read in or write out in files, and can execute shell Commands on the underlying operating system. Data is being stolen through the various attacks at all the time. Hackers rarely get caught which are more expert. Any attacker that can obtain access, it could spell Disaster. A sql injection attacks involves the Modification of sql statements that are used in a web Application through the use of attacker-input data. Unfortunately the harm of sql injection is only found when the theft is discovered. Improper validation and improper construction and incorrect input of sql Statements in web applications can lead them theft to Sql injection attacks. [6] Thus sql injection is a potentially destructive and prevalent attack that the open Web application security project (owasp) listed it as the number one threat to web applications.

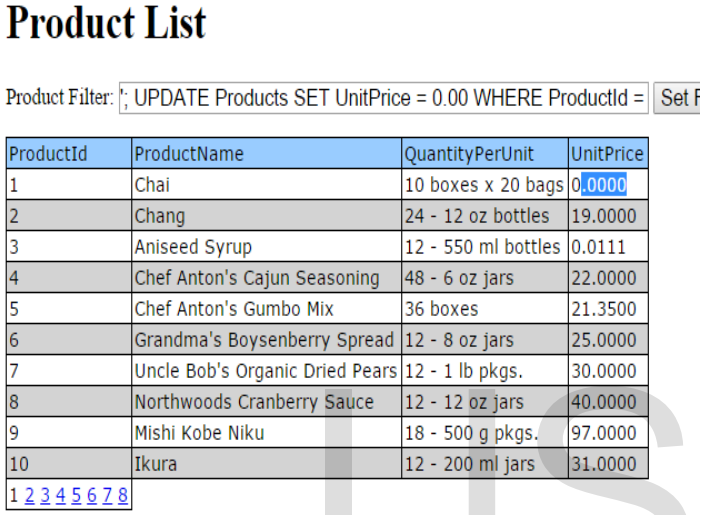
**5. SIMULATION RESULT ANALYSIS**

Sql injection can helps to retrieve sensitive information like password or credit card details, To prevent sql injection developer should has to take some measure steps like use session in place of query string to transfer value from one page to another. Store sensitive information like password or credit card to XML or file system which is not easily accessible. If using Query string is necessary try using URL Encoding technique.

Now a day's some DBMS like MS Sql server supports regular expression validation which protect data insertion

like " ' ". All DBMS doesn't supports " ' " handle it is very necessary replace it with some other character.




**OUR ALGORITHM STEPS ARE:**

```
StringstrCnx=ConfigurationSettings.AppSettings["cnxNWindBad"];
SqlConnection cnx = new SqlConnection(strCnx);
cnx.Open();
string strQry = "SELECT Count(*) FROM Users WHERE UserName='" +txtUser.Text + "' AND Password='" + txtPassword.Text + "'";
int intRecs;
SqlCommand cmd = new SqlCommand(strQry, cnx);
cmd.CommandType= CommandType.Text;
intRecs = (int) cmd.ExecuteScalar();
if (intRecs>0)
{
FormsAuthentication.RedirectFromLoginPage(txtUser.Text, false);
}
else
{
lblMsg.Text = "Login attempt failed.";
}
cnx.Close();
```

**//Prevention**

```
String str=ConfigurationSettings.AppSettings["cnxNWindBetter"];
using(SqlConnection cnx = new SqlConnection(strCnx))
{
cnx.Open();
SqlCommand cmd = new SqlCommand("procVerifyUser", cnx);
cmd.CommandType= CommandType.StoredProcedure;
SqlParameterprm=new SqlParameter ("@username",SqlDbType.VarChar,50);
prm.Direction=ParameterDirection.Input;
prm.Value = txtUser.Text;
cmd.Parameters.Add(prm);
prm = new SqlParameter ("@password",SqlDbType.VarChar,50);
prm.Direction=ParameterDirection.Input;
prm.Value = txtPassword.Text;
cmd.Parameters.Add(prm);
string strAccessLevel = (string) cmd.ExecuteScalar();
if (strAccessLevel.Length>0)
{
FormsAuthentication.RedirectFromLoginPage(txtUser.Text, false);
}
else
{ lblMsg.Text = "Login attempt failed.";
}
 }
```

**6. CONCLUSION**

SQL attackers create crafted input data so that SQL interpreter have to accept the query and give the permission to execute the commands and give his desired results. SQL Injection attack breaks the security in the database layer and can alter, steal or destroy our database through using web application.

Sql injection can helps to retrieve sensitive information like password or credit card details, To prevent sql injection developer should has to take some measure steps like use session in place of query string to transfer value from one page to another. Store sensitive information like password or credit card to XML or file system which is not easily accessible. If using Query string is necessary try using URL Encoding technique.

Now a day's some DBMS like MS Sql server supports regular expression validation which protect data insertion like " ' ". All DBMS doesn't supports " ' " handle it is very necessary replace it with some other character.

**7. REFERENCES**

[1] W. G. Halfond, J. Viegas and A. Orso, "A Classification of SQL Injection Attacks and Countermeasures," College of Computing Georgia Institute of Technology IEEE, 2006.

[2] Sruthi Bandhakavi, Prithvi Bisht, P. Madhusudan, CANDID: Preventing SQL Injection Attacks using Dynamic Candidate Evaluation. Proceedings of the 14th ACM conference on Computer and communications security. ACM, Alexandria, Virginia, USA.page:12-24.

[3] Marco Cova, Davide Balzarotti. Swaddler: An Approach for the Anomaly-based Detection of State Violations in Web Applications. Recent Advances in Intrusion Detection, Proceedings, Volume: 4637 Pages: 63-86 Published: 2007.

[4] William G.J. Halfond, Jeremy Viegas and Alessandro Orso, "A Classification of SQL Injection Attacks and Countermeasures," College of Computing Georgia Institute of Technology IEEE, 2006.

[5] Z. Su and G. Wassermann. The Essence of Command Injection Attacks in Web Applications. ACM SIGPLAN Notices. Volume: 41, pp: 372-382, 2006.

[6] Ke Wei, M. Muthuprasanna, Suraj Kothari , Dept. of Electrical and Computer Engineering , Iowa State University Ames, IA – 50011 ,Email: {weike,muthu,kothari}@iastate.edu

[7] William G.J. Halfond, Jeremy Viegas, and Alessandro Orso College of Computing Georgia Institute of Technology {whalfond|jeremyv|orso}@cc.gatech.edu

[8] Z. Su and G. Wassermann. The Essence of Command Injection Attacks in Web Applications. In The 33rd Annual Symposium on Principles of Programming Languages (POPL 2006), Jan. 2006.

[9] F. Valeur, D. Mutz, and G. Vigna. A Learning-Based Approach to the Detection of SQL Attacks. In Proceedings of the Conference on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA), Vienna, Austria, July 2005.

[10] T. M. D. Network. Request.servervariables collection. Technical report, Microsoft Corporation, 2005. http://msdn.microsoft.com/library/default. asp?url=/library/en-us/iissdk/html/9768ecfe-8280-4407-b9c0-844f75508752.asp.

[11] José Fonseca CISUC - Polithecnic Institute of Guarda, Marco Vieira, Henrique Madeira DEI/CISUC - University of Coimbra. Testing and comparing web vulnerability scanning tools for SQL injection and XSS attacks. Retrieved July 10, 2007, from http://ieeexplore.ieee.org

[12] Yuji Kosuga, Kenji Kono, Miyuki Hanaoka Department of Information and Computer Science Keio University. Sania: Syntactic and Semantic Analysis for Automated Testing against SQL Injection. Retrieved November 12, 2007, from IEEE Computer Society. http://ieeexplore.ieee.org

[13] Benjamin Livshits and U´ lfar Erlingsson. Microsoft Research. Using Web Application Construction Frameworks to Protect Against Code Injection Attacks. Retrieved June 14, 2007, from http://ieeexplore.ieee.org

[14] Tapan Kumar and Tapesh Kumar (20 August 2016). "WEB USAGE MINING USING SEMANTIC WEB APPROACH: A STUDY, SURVEY AND ANALYSIS."

[15] Tapan Kumar and DR. R. Rama Kishore (July, 2016). "SPARQL Execution of Semantic Web Data: A STUDY AND ANALYSIS."