International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 5 Issue 6 ǁ June. 2017 ǁ PP. 56-61

Devoid Web Application From SQL Injection Attack

[1]Swati Yadav, [2]Ms.Geeta
[1]M.tech CSE GITM, Gurgaon MDU University Rohtak, Haryana, India
[2]Assistant Professor GITM, Gurgaon MDU University Rohtak, Haryana, India

ABSTRACT: The entire field of web based application is controlled by the internet. In every region, World Wide Web is hugely necessary. So, network assurance is badly assuring job for us. Several kind of attacker or application programmer is attempting to split the immunity of information and destroy the instruction composed in the database. The SQL Injection Attack is very large safety measure risk in that present day. The indicated attacks allow to attacker' s unlimited access from the database or still authority of database those determine web based application. That manages conscious and secret records and put the injurious SQL query put to modify the expected function. Many database reviewer and theorist give distinct concept to avoid regarding SQL Injection Attack. But no one of the concept is completely adaptable to. This research introduces a latest framework to protecting web based application from the SQL Injection Attack. Introduced framework i.e. present in this research is based on two techniques known as SQM (SQL Query Monitor) and Sanitization Application. That is the two ways filter program which analyses the user query and generate a separate key for user before it is sent to the application server. Several aspects of SQL Injection Attack are also discussed in that research.

KEYWORDS: SQL, Database, Security, Attack,Authentication.

I. INTRODUCTION

Web application can have sensitive and confidential data which is stored in database.web applications accepts the data from the users. This data is retrieved from the database through the queries.SQL Injection attack is one of the most popular attack used in system hacking or cracking. Web applications can be harmed by SQL INJECTION ATTACK Using SQL INJECTION ATTACK attacker can gain information or have unauthorized access to the system. When attacker gains control over web application maximum damage is caused.

To insert, retrieve, update, & delete the data from database SQL language is used. When we enter data in the input fields it becomes part of the SQL query written at the backend. For example, to login in our inbox, we provide loginid and password. The loginid and password form the part of the internal SQL query. Then the SQL query is executed on the database to check whether the login credentials provided match with those present in the tables on the database. The attacker, who wants to gain access to the inbox, provides injected code instead of correct input in the input fields of the web application. This injected code changes the structure of the original SQL query and consequently, allows the attacker to gain access to the information it was not authorized for. This type of attack which allows the attacker to alter the original SQL query by adding the injected SQL code in the input field is known as SQL Injection Attack (SQLIA).[1]. In SQLIA, Attacker attempts to change SQL query by inserting new SQL keywords. The attacker modifies the original SQL query by inserting new SQL query through user input field. Injected query formed syntactically correct when concatenated with sql command. The data within the database will be altered, extracted or even dropped.

II. RELATED WORK

There are several works that have been proposed and implemented in the past by various researchers to secure web application from SQL Injection Attacks. Some of the most important attacks and their limitations are discussed in this thesis and they are compared with the techniques proposed in the thesis. The most important works are listed below.

Indrani Balasundram and E.Ramaraj [8] proposed an authentication scheme for Preventing SQL Injection Attack Using Hybrid Encryption (PSQLIAHBE).The algorithm uses both Advance Encryption Standard (AES) and Rivest-Shamir- Adleman (RSA) to prevent SQL injection attacks. A unique secret key is issued to every user and server uses combination of private key and public key for RSA encryption. The system model includes three phases i.e. registration phase, login phase and verification phase. The proposed scheme declares to be very efficient as it requires 961.88ms for encryption or decryption. In this method, two level of encryption is applied on login query:

- To encrypt user name and password, symmetric key encryption via user's secret key.
- To encrypt the query, asymmetric key encryption via server's public key.

Disadvantages:

- It doesn't work for URL based SQL injection attacks.
- Lack of secure registration phase.

Allen Pomeroy and Qing Tan [12] have proposed a technique for finding vulnerabilities in Web Application such as SQL injection attack by network recording. In this approach network forensic techniques and tools are used to analyse the network packets containing get and post requests of a web application. This approach uses network based Intrusion Detection System (IDS) to trigger network recording of suspected application attacks. The technique builds models of the typical queries and then monitors the application at runtime to identify queries that do not match the model. This new approach is very efficient in practice however; it requires more experimentation and comparison with detection method.

Disadvantages:

- Not efficient in case of high volume traffic.
- Packet fragmentation attack could bypass detection.

X. Fu, X. Lu, B. Peltsverger, S. Chen, K. Qian and L. Tao [20] proposed SAFELI a Static Analysis framework for Detecting SQL Injection Vulnerabilities in web application. SAFELI framework aims at identifying the SQL Injection attacks during the compile-time. This static analysis tool has two main advantages. Firstly, it does a White-box Static Analysis and secondly, it uses a Hybrid-Constraint Solver. For the White-box Static Analysis, the proposed approach considers the byte-code and deals mainly with strings. For the Hybrid-Constraint Solver, the method implements an efficient string analysis tool which is able to deal with Boolean, integer and string variables. Its implementation was done on ASP.NET Web applications and it was able to detect vulnerabilities that were ignored by the black-box vulnerability scanners. This approach is an efficient approximation mechanism to deal with string constraints. However, the approach is only dedicated to ASP.NET vulnerabilities.

Disadvantages:

- This framework does not work for dynamic based approach.

III. TOOLS

1.FRONT END TOOL:

JSP

JSP technology is used to create web application just like Servlet technology. It can be thought of as an extension to servlet because it provides more functionality than servlet such as expression language, jstl etc.

A JSP page consists of HTML tags and JSP tags. The jsp pages are easier to maintain than servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tag etc.

2. BACK END TOOL:

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius' daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.[9] For proprietary use, several paid editions are available, and offer additional functionality.

MY SQL WORKBENCH

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more. MySQL Workbench is available on Windows, Linux and Mac OS X.

ECLIPSE

Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE.[6] It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins, including Ada, ABAP, C, C++, COBOL, D, Fortran.

JDK

The **Java Development Kit** (**JDK**) is an implementation of either one of the Java Platform, Standard Edition, Java Platform, Enterprise Edition, or Java Platform, Micro Edition platforms released by Oracle

Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, macOS or Windows. The JDK includes a private JVM and a few other resources to finish the development of a Java Application.Since the introduction of the Java platform, it has been by far the most widely used Software Development Kit (SDK). On 17 November 2006, Sun announced that they would release it under the GNU General Public License (GPL), thus making it free software

TOMCAT

Apache Tomcat, often referred to as **Tomcat Server**, is an open-source Java Servlet Container developed by the Apache Software Foundation (ASF). Tomcat implements several Java EE specifications including Java Servlet, JavaServer Pages (JSP), Java EL, and WebSocket, and provides a "pure Java" HTTP web server environment in which Java code can run.

IV. RESULTS

The adminlogin table appears in the database, which shows admin can perform like add, delete or edit the customer system. Using SQL Server database as a backend database for the credentials i.e. username and password. This adminlogin is responsible for maintaining the database/system, has specific controls to a website that allows them to control website.

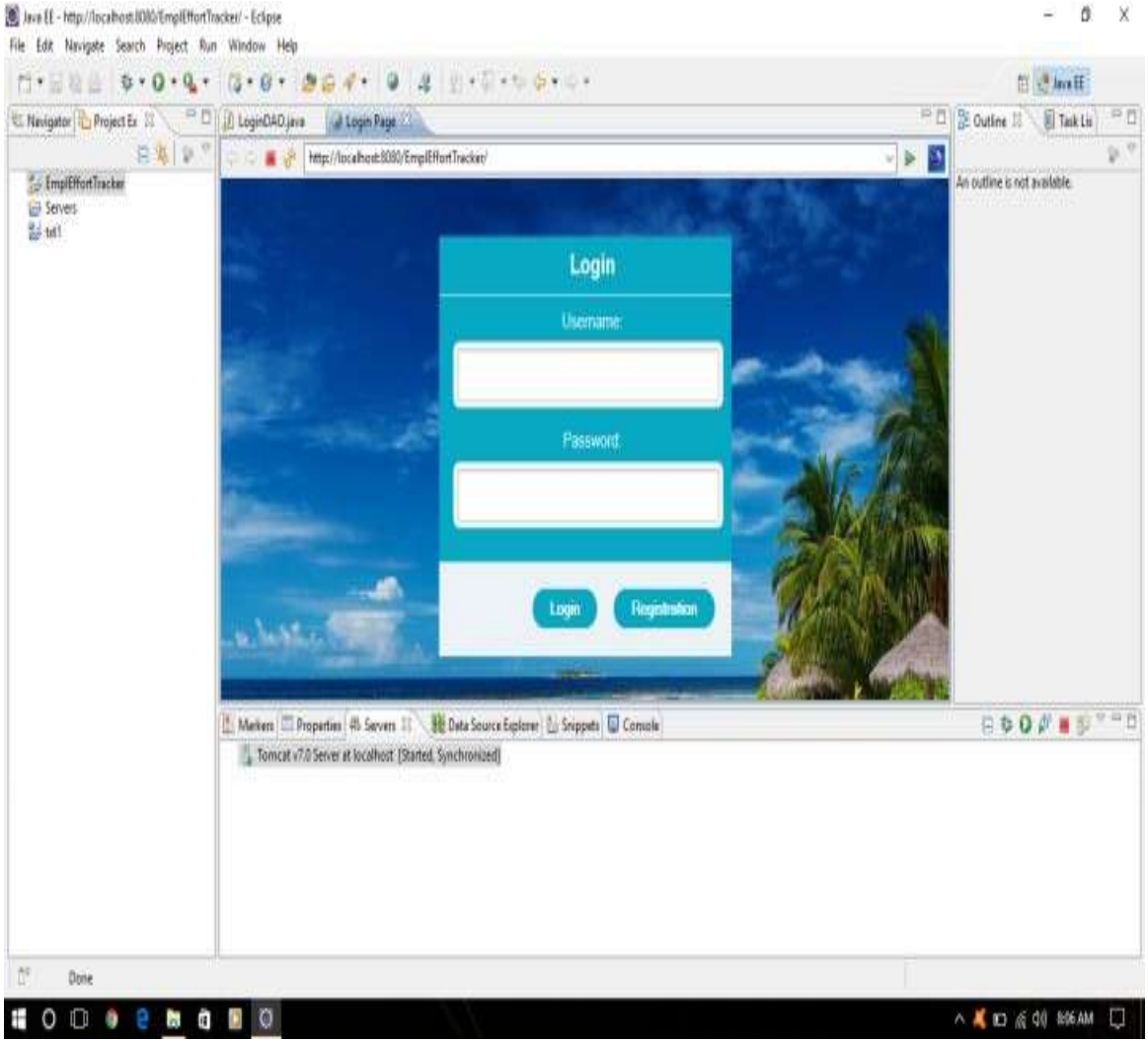

Figure 1

OUTPUT

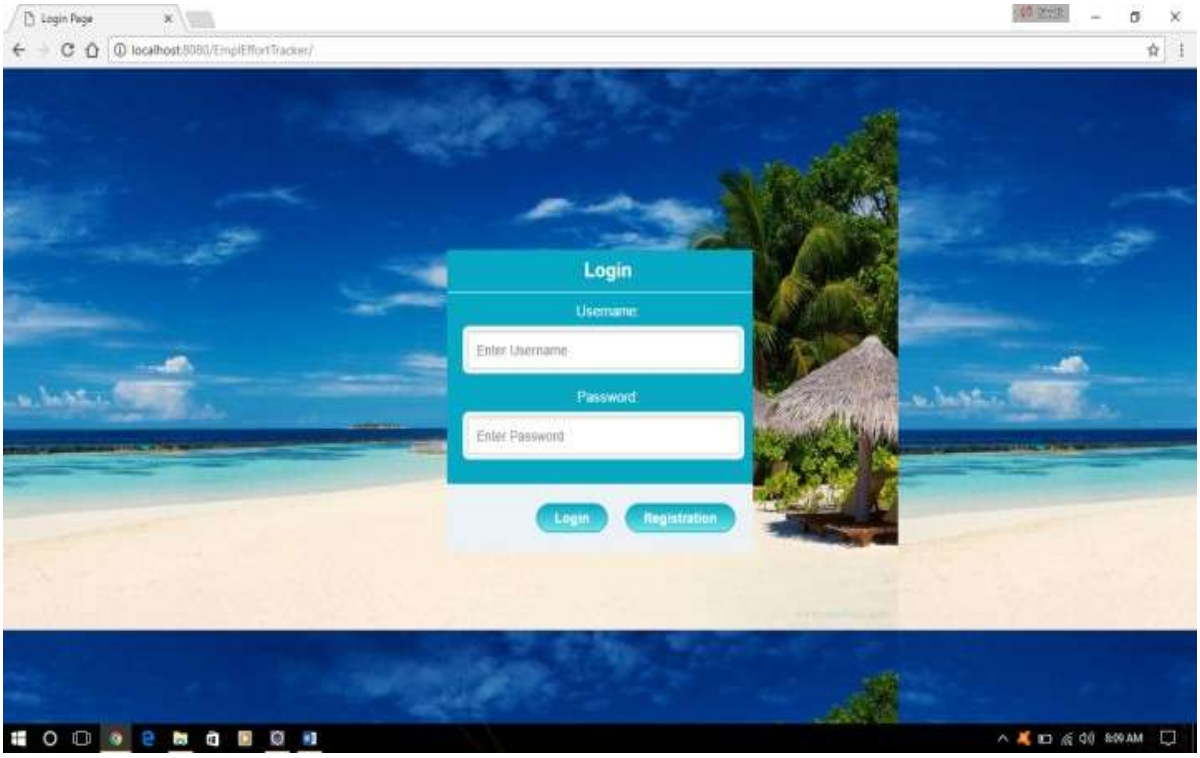

Figure 2. Login Page

Register Page

If User have no account then he/she will be create a account by registration .For registration he can fill the following details and then register the page.

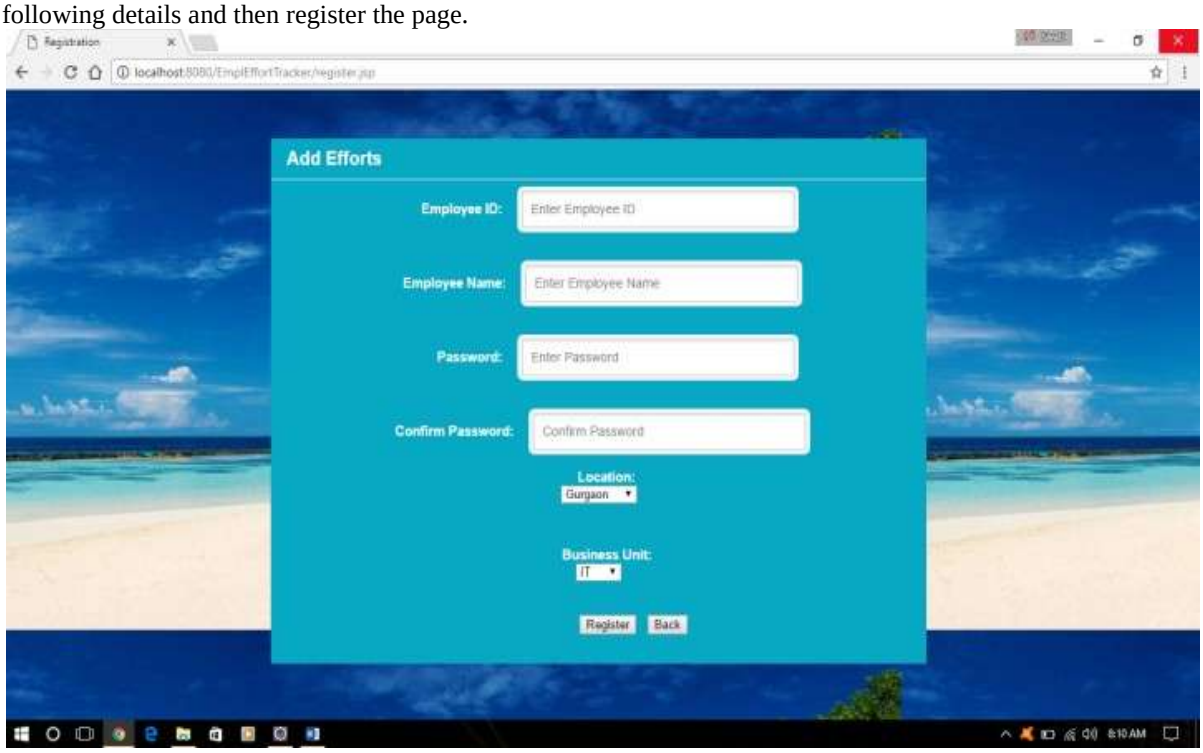

Figure 3. Register Page

View Task

When the user completed the registration process and the login process.When the user successfully login the welcome page he can add task and then can view these task in the view task.The page will look like this.

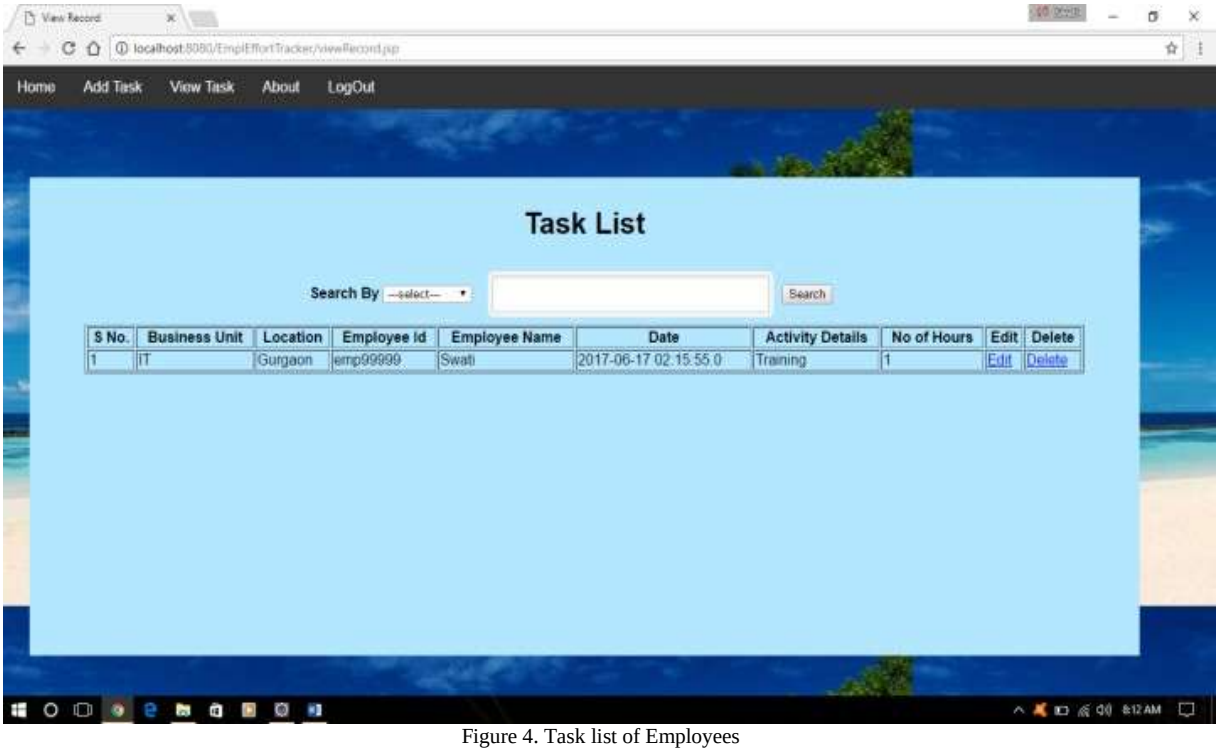

Figure 4. Task list of Employees

Invaild User

When the user want to login the welcome page and if the user fill wrong username or password then the user cvan not login the welcome page and the page show invalid user id or password.

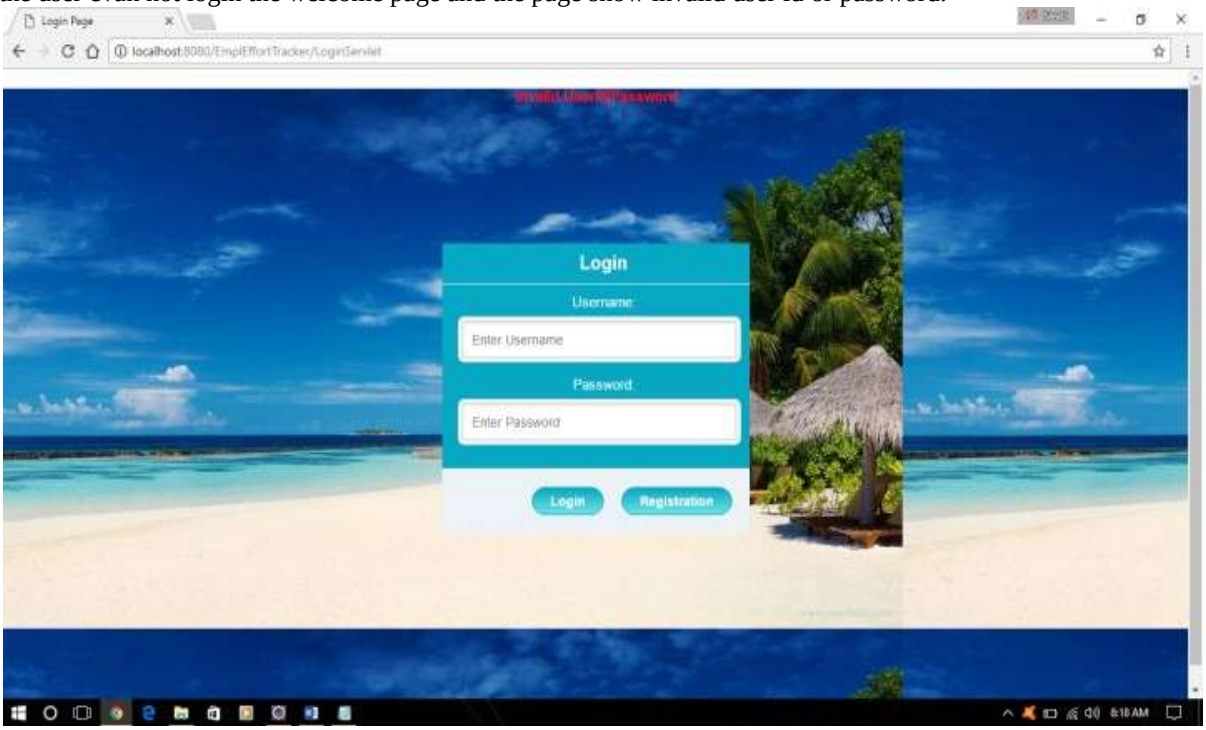

Figure 5. Invaild Login id and password

V. CONCLUSION AND FUTURE WORK

The SQL Injection Attack is the largest accessible security risk in the network based computer database in today because all attacker or application programmer attempt to crack the information safety measure accepting similar form of violation. SQL Injection Attack like one of the top-10 to make threats or vulnerabilities to web application, ambition the end part information/records. Generally attacker tries to confuse the intermediate layer technology by reshaping the SQL queries. This research introduces the framework to save web based application or database from the SQL Injection Attack. In such a manner this proposed scheme regarding security against SQLIA, is too sensitive. SQL Injection Attack abuse security sensitivity present in computer Database of a function through inserting few malicious codes. Several clarifications are disposed through distinct experimenters although no one resolution is completely capable in order to anticipate the information against the particular initiative.

Throughout the literature of research work, we get many approaches/ methods were not able or helpful to identify and avoid SQL Injection Attack and could not make much secure our database. This scheme is able to secure information against SQL injection attacks, but consumes more time for analyzing or filtering process as compared to single technique. So, in future will trying to develop this framework via composing it well active, protect and capable for any kind of attacks and will more focus on reducing time complexity of proposed algorithm. So that this modified scheme will be ready to stop SQL Injection Attack effectively and will secure web applications efficiently. This scheme can be applied to real time systems and applications to increase more security and vulnerability to real life systems.

REFERENCES

[1]. **McClure, R.A. and Kruger, I.H.**, " SQL DOM: compile time checking of dynamic SQL statements. 27th International Conference on Software Engineering(ICSE 2005)" , 15-21 May 2005, pp.

[2]. **Ettore Merlo et al.** " Insider and outsider threat sensitive SQL injection vulnerability analysis in PHP" IEEE 2006.

[3]. **William G.J. Halfond and Alessandro Orso**" Preventing SQL Injection Attacks Using AMNESIA," ICSE' 06, Shanghai, China, 2006.

[4]. **W. G.J. Halfond, J. Viegas, and A. Orso**, " A classification of SQL injection attacks and countermeasures" , In Proceedings of the international Symposium on secure Software Engineering (ISSSE), 2006.

[5]. **X. Fu, X. Lu, B. Peltsverger, S. Chen, K. Qian, and L. Tao.,**" A Static Analysis Framework for Detecting SQL Injection Vulnerabilities" , COMPSAC July 2007.

[6]. **Amirtahmasebi, K, Jalalinia, S.R., and Khadem, S.,** "A Survey of SQL Injection defense mechanism, International Conference for Internet Technology and Secured Transaction" (ICITST 2009), 9-12 Nov. (2009),

[7]. **Atefeh Tajpour, Suhaimi Ibrahim, Maslin Masrom**, **"** SQL Injection Detection and Prevention Techniques" , International Journal of Advancements in Computing August 2011.

[8]. **Indrani Balasundaram, E.Ramaraj**" An Authentication Scheme for Preventing SQL Injection Attack Using Hybrid Encryption(PSQLIAHBE)" ,(ISSN 1450-216X Vol.53 No.3 (2011).

[9]. **Pomeroy, A Qing Tan Sch. of Comput. & Inf. Syst., Athabasca Univ., Athabasca, AB, Canada** "Effective SQL Injection Attack Reconstruction Using Network Recording" in Computer and Information Technology (CIT), 2011 IEEE 11th International conference Issue Date: Aug. 31 2011-Sept. 2 2011 On page(s): 552 – 556.

[10]. **Prasant Singh Yadav, Pankaj Yadav, K.P.Yadav**" A Modern Mechanism to Avoid SQL Injection Attacks in Web Applications" , IJRREST: International Journal of Research Review in Engineering Science and Technology, Volume-1 Issue-1, June 2012.

[11]. **Veera Venkateswaramma P**, " An Effective Approach for Protecting Web from SQL Injection Attacks" , International Journal of Scientific & Engineering Research, Volume 3, 2012.

[12]. **Neha Singh, Ravindra Kumar Purwar, "** SQL Injection – A Hazard To web applications, International Journal of Advanced Research in computer Science and Software Engineering" ,vol.2,Issue 6,June 2012.

[13]. **V. Nithya, R.Regan, J.vijayaraghavan**" International Journal Of Engineering And Computer Science" ISSN:2319-7242 Volume 2 Issue 4 April, 2013 Page No.

[14]. **Ericka Chickowski**, Contributing Writer " Dark Reading" May, 2013.

[15]. **Debabrata Kar, Suvasini Panigrahi**, " Prevention of SQL Injection Attack Using Query Transformation and Hashing, IEEE International Advance Computing Conference" (IACC), 2013.

[16]. **Surya Pratap Singh, Avinash Singh, Upendra Nath Tripath, Manish Mishra**" Proactive Mechanism of Protection against SQL Injection Attack" , Gorakhpur, Vol. 3, Issue 5, 2015.