A Practical Approach for SQL Injection Prevention Attacks Using IPS

Harpreet Kaur[1], Mrs. Sarika Dhingra[2]

Student, CSE, JCDMCOE, Sirsa, India [1]

Asst Professor, CSE, JCDMCOE, Sirsa, India [2]

Abstract: This paper has been proposed the IPS technique to prevent the intruder's attack. Database is the key component for store the information in organizations. Database information is the crucial part as it needs more security because this contains the password, user information, even it can be in encrypted form but intruder can alter the information or delete it. This paper has been considered the scenario of company where the multiple users can perform different operations and there is need to traverse the operations of the user. The IPS technique has been proposed to prevent the intruder's attack. The Query weight concept has been used and greater weight queries will be executed only when the user is provided by the OTP and the malicious transactions should not be executed. The malicious transaction is that transaction which the user is not authorized to perform. The history of transactions with user id is also tracked for advance analysis but it would require more memory. The SQL Injection attack has been prevented and there will be no loss of information. The proposed approach has been implemented in ASP.net using backend SQL Server 2008.

Keywords: Intruder, IPS, SQL-Injection, Database, Security.

I. Introduction

Web browsers are software applications that allow users to retrieve data and interact with content located on web pages within a website.

The web is a highly programmable environment that allows mass customization through the immediate deployment of a large and diverse range of applications, to millions of global users. Two important components of a modern website are flexible web browsers and web applications; both available to all and sundry at no expense.

As stated, websites depend on databases to deliver the required information to visitors. If web applications are not secure, i.e., vulnerable to, at least one of the various forms of hacking techniques, then your entire database of sensitive information is at serious risk.

Some hackers, for example, may maliciously inject code within vulnerable web applications to trick users and redirect them towards phishing sites. This technique is called Cross-Site Scripting and may be used even though the web servers and database engine contain no vulnerability themselves. Recent research shows that 75% of cyber attacks are done at web application level.

Database security is the system, processes, and procedures that protect a database from unintended activity. Unintended activity can be categorized as authenticated misuse, malicious attacks or inadvertent mistakes made by authorized individuals or processes. [4].

One of the techniques available to the would-be information thieves is SQLInjection (SQL-I). SQL-I attacks involve a variety of methods, but the intention of an attacker using it is to submit specially-chosen patterns when asked for the user's username and password on an internet form

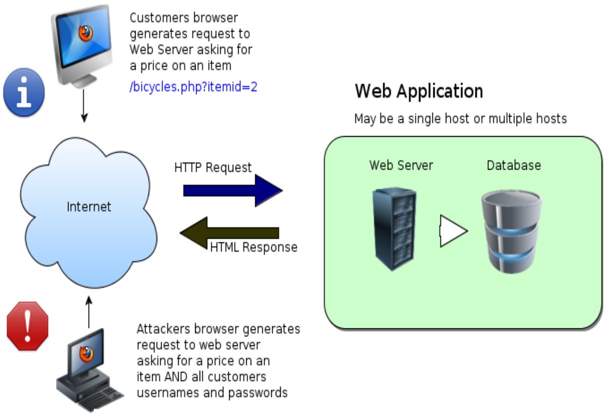

Customers browser generates request to Web Server asking for a price on an item
/bicycles.php?itemid=2
Web Application
May be a single host or multiple hosts
Internet
HTTP Request
HTML Response
Web Server
Database
Attackers browser generates request to web server asking for a price on an item AND all customers usernames and passwords

Fig 1 Web Server Attack

Security Attacks (SQL-Injection):

Once an attacker realizes that a system is vulnerable to SQL Injection, he is able to inject SQL Query / Commands through an input-form field. This is equivalent to handing the attacker your database and allowing him to execute any SQL command including DROP TABLE to the database. An attacker may execute arbitrary SQL statements on the vulnerable system. This may compromise the integrity of your database and/or expose sensitive information. Depending on the back-end database in use, SQL injection vulnerabilities lead to varying levels of data/system access for the attacker. It may be possible to manipulate existing queries, to UNION (used to select related information from two tables) arbitrary data, use sub selects, or append additional queries. In some cases, it may be possible to read in or write out to files, or to execute shell commands on the underlying operating system. Certain SQL Servers such as Microsoft SQL Server contain stored and extended procedures. If an attacker can obtain access to these procedures, it could spell disaster. The impact of SQL Injection is only uncovered when the theft is discovered. Data is being unwittingly stolen through various hack attacks all the time. The more expert of hackers rarely get caught.

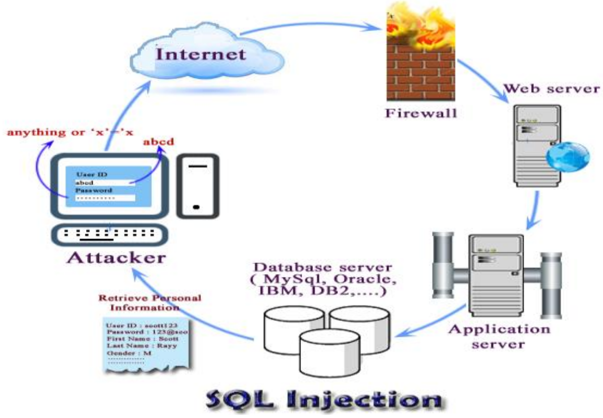


Figure 2: SQL Injection

a. AND/OR Attack

Web programmers often take string values entered by an Internet user on a form that represents user names and passwords and place them directly into the SQL statement to be run against a database. [3] A simple test SQL statement that may be used is the following example.

SELECT username, password FROM Authouser WHERE username = ‘usernameFromForm’.

AND password = ‘passwordFromForm’;

In this example, the values usernameFromForm and passwordFromForm are the literal values obtained from the form. The intent is using the username and password obtained from the form to see if there is a matching username and password in the Authouser table. If any rows are returned, the user is authenticated. However, if the web programmer is not careful and uses this method and the form values without checking them, a hacker may instead pass arbitrary values that the programmer did not originally anticipate. One such attack is the basic attack that involves the AND or OR logic in the SQL predicate. The hacker can specify a valid username such as “Narinder” and then specify the password as “' OR '1'='1” in the form. Then query will be like this.

SELECT username, password FROM Authouser WHERE username = ‘Narinder’ AND password = '' OR '1'='1';

b. Comments Attack

SQL allows inline commenting within the SQL “code”. This allows two variations of SQL-I comments attacks. One simple variation is assigning username to be a valid username followed by comment characters. For example, we assign username = “admin' --”. Then our SQL test query may look like the following [3].

SELECT username, password FROM Authouser WHERE username = 'admin' --' AND password = 'anything';

Everything after the “--” in the WHERE clause will be ignored, so this will allow the hacker to log in as “admin”. This is a method of using comments as a way of ignoring the rest of the query

Privilege elevation exploits can be defeated with a combination of query-level access control and traditional intrusion prevention systems (IPS). Query-level access control can detect a user who suddenly uses an unusual SQL operation, while an IPS can identify a specific documented threat within the operation.

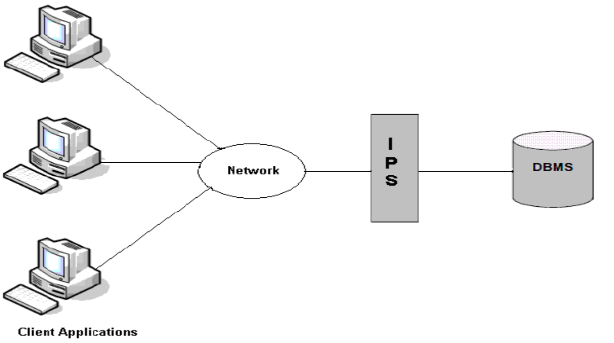


Fig 3 Database Management System with Intrusion Prevention System

Database security Techniques

There are some security techniques that may prove useful in stimulating the database [31].

a. Securing Database using Cryptography

Users are divided into two levels: Level 1 (L1) and Level 2 (L2). Level 1 user have access to their own private encrypted data and the unclassified public data, whereas Level 2 users have access to their own private data and also classified data which is stored in an encrypted form. There is also provision to perform column-wise encryption that allows the users to classify the data into sensitive data and public data. This classification helps in selecting to encrypt only that data which is critical and leaves the public data untouched thereby reducing the burden of encrypting and decrypting the whole database, as result of which the performance is not degraded. The technique involves designing a framework to encrypt the databases over the unsecured network in a diversified form that comprise of owning many keys by various parties. In the proposed framework, the data is grouped depending upon the ownership and on other conditions.

b. Securing Database using Steganography

There are various techniques in steganography that can be implemented to hide critical data and prevent them from unauthorized and direct access. The various techniques include still image steganography, audio steganography, video steganography, IP Datagram steganography. In the proposed scheme the data is embedded in the LSB’s of the pixel values. The pixels values are categorized into different ranges and depending on the range certain number of bits is allocated to hide the sensitive data. The different scheme is that the image is divided into fixed number of blocks. Histogram of each block is calculated along with the maximum and minimum points to mask the data. This mechanism increases the hiding capacity of the data. Another approach is that the technique involves using prime numbers and natural numbers to enhance the number of bit planes to cloak the data in the images.

c. Securing Database using Access Control

Another is the authorization technique for video databases. In the this scheme, the access to the database and to a particular stream of the video is granted only after verifying the credentials of that user. The credentials may not just be the user-id but it may be the characteristics that define the user and only after successful verification of the credentials the user is granted the permission to access the database. There is generalized authorization model for multimedia digital libraries. The scheme involves integrating the three most common and widely used access control mechanisms namely: mandatory, discretionary and role-based models into a single framework to allow a unified access to the protected data. The technique also addresses the need of continuous media data while supporting the QoS constraints alongside preserving the operational semantics. In the explained technique is based on authorization views which enable authorization transparent querying in which the user queries are formed and represented in terms of database relations and are acceptable only when the queries can be verified using the information contained in the authorization rules. The work presents the new techniques of validity and conditional validity which is an extension of the earlier work done in the same area.

II. LITERATURE REVIEW

Author has been proposed the attacks using SQL. The traditional solution to database security has a drawback that it cannot deal with malicious attacks by persons with legal identity, and that, it is in general not cost-effective to users who have different security requirements for it only offers fixed security level. By adopting multi-layer security model, namely "user +OS +DBMS +transaction-level intrusion tolerance", it integrates redundancy and variety technology; by adopting integral security strategy and server-oriented intrusion tolerance technology, it realizes the survivability and availability of database, and the confidentiality and integrity of sensitive data. In this way, it can effectively resist malicious attacks by persons with legal identity and reduce the cost of security [1].

Databases are the repositories of the most important and expensive information in the enterprise. With the increase in access to data stored in databases, the frequency of attacks against those databases has also increased. A database threat refers to an object, person or other entity that represents a risk of loss or corruption of sensitive data to an asset. Today, in many business organizations, the databases and data assets are poorly protected. Databases should be secured more than any systems in the organization. To secure a database environment, many database security models need to be developed. The purpose of the paper is to highlight and threat types and their impacts on sensitive data, and presents different security models. The assumption underlying this study is that by understanding the weaknesses and the threats facing databases,
database administrators can then begin to create a security plan to better protect their databases [2].

The security enhanced module is designed to improve the security of the database. The design ideas and operating principle are studied. The realizing of the security checked module, including the optimization of security rule base, the realizing of security checking and the management of securityrules are explained. The experiment shows that although the response time is increased, the security of the database is enhanced enormously [3].

With widespread adoption of the Web as an instant means of information dissemination and various other transactions, including those having financial consequences like e-banking, e-shopping, online payment of bills etc, they are becoming more and more dependent on web applications. An unauthorized access to this much of confidential data by a crafted user can threat their confidentiality, integrity, and authority. As a result, the system could bear heavy loss in giving proper services to its users or it may face complete destruction. Sometimes such type of collapse of a system can threaten the existence of a company or a bank or an industry. SQL Injection attacks are one of the most dangerous security threats to web applications. Several researchers have proposed several ways to prevent SQL injection attacks in the application layer but very little emphasis is laid on preventing SQL Injection attacks in stored procedures in the database layer. In this paper a novel technique to prevent SQLIA in stored procedures is proposed. This technique provides a two phase security to the application, so that, if one phase is compromised, the second phase can still prevent the attack [4].

SQL injection is one of the biggest challenges for the web application security. Based on the studies by OWASP, SQL injection has the highest rank in the web based vulnerabilities. In case of a successful SQL injection attack, the attacker can have access to the web application database. With the rapid rise of SQL injection based attacks, researchers start to provide different security solutions to protect web application against them. One of the most common solutions is the using of web application firewalls. Usually these firewalls use signature based technique as the main core for the detection. In this technique the firewall checks each packet against a list of predefined SQL injection attacks known as signatures. The problem with this technique is that, an attacker with a good knowledge of SQL language can change the look of the SQL queries in a way that firewall cannot detect them but still they lead to the same malicious results. In this paper first they described the nature of SQL injection attack, then they analyzed current SQL injection detection evasion techniques and how they can bypass the detection filters, afterward they proposed a combination of solutions which helps to mitigate the risk of SQL injection attack [5].

III.OBJECTIVES

The main Aim of this research is to understand the security threats and identify the appropriate security techniques used to detect and prevent database from malicious attack. Intrusion prevention system comes in between the network and DBMS as shown. Thus it checks all the transactions before they are committed and prevents the malicious ones from getting committed.

1. This model will even help us to avoid damage from those malicious transactions which cannot be rolled back.

2. This model shown acts before the database and prevents the transactions from getting committed in database.
3.Instant detect and prevent database from malicious transactions using algorithm for instant detection and prevention.
4. Implementation of this research will be demonstrated by Developing the Web Application in ASP.Net 4.0 with Backend SQL Server 2008.
Setting up an intrusion prevention system (IPS) along with a database management system comprises of three phases namely.

a. Profiling the transactions and assigning to users
b. Giving weight to commands and
c. Instant detection and prevention.

IV. ALGORITHM

These are the steps which have been followed for database security and this will protect the database from the crucial SQL queries. The weight age concept has been considered with the queries for identify the crucial queries.
1. Worked on above mentioned Phases
2.Designed an Algorithm for INSTANT detection and Prevention.
3. Implemented the algorithm with help of programming.
4. Performed Experimental Evaluation.
5. Introduced any hashing technique to improve data security.
6. Perform different Malicious Transactions to verify success rate of method.
7. Experimental results and analysis of the application.
Algorithm is used for solving the problem step by step which can be implemented with help of programming in any language. In the IPS Algorithm, the weight age has been given to the commands according the crucial level. The more crucial command, the more weight age it will contain and accordingly, OTP (One type Password) will be required and generated by the admin and then reset for the purpose of security. The complete flow chart has been written as which explains the concept in steps.

```
1. Start
2. Get User Id (ID) & Password (P)
3. Analyse Query (Select |Insert |Delete |Update)
4. Identify Weight age of Query
5. If Weight age>=3
    Move to step6
   Else
   Move to step 9
6. Command Nature Computation
7. If New
     Then Move to Step 8
Else
Move to Step 11
8. If OTP Matched
Then Move to Step 9
Else
Move to Step 11
9. Execute Query
10. Save History
11. End
```

V. FLOW CHART

A flowchart is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analysing, designing, documenting or managing a process or program in various fields.

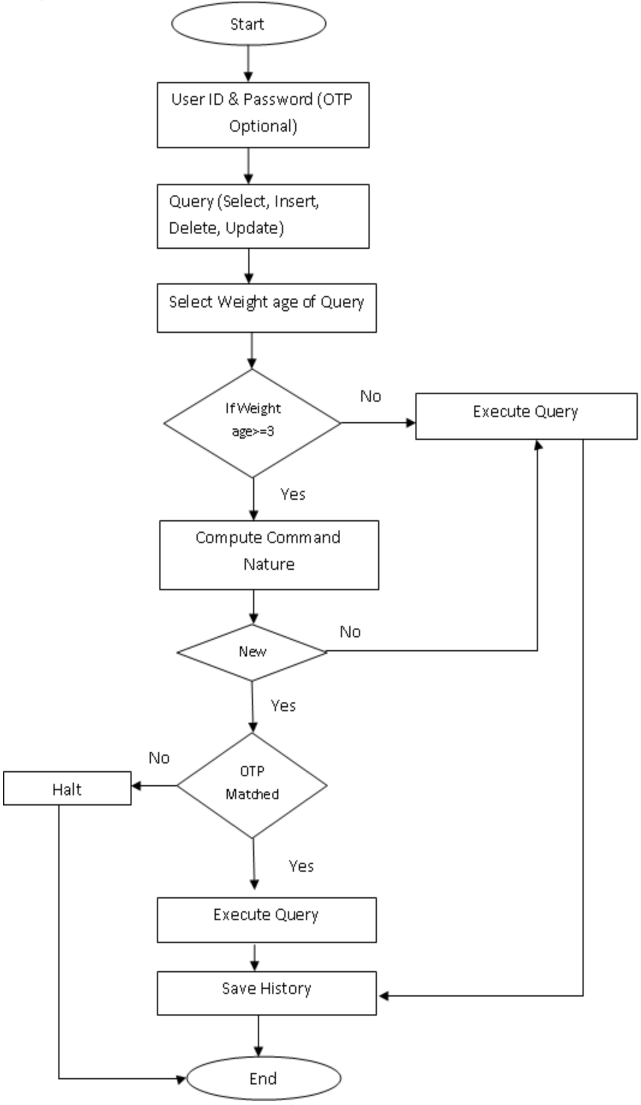


Figure 4: Flow Chart

VI. RESULTS

The intrusion Prevention has been developed by using ASP.Net 4.0 using SQL Server 2008 as Backend for keep the record of previous queries. The users have been created along with the OTP provided by the admin. The update and delete have more weight age and select, insert has less weight age.
Initially the database has been designed.

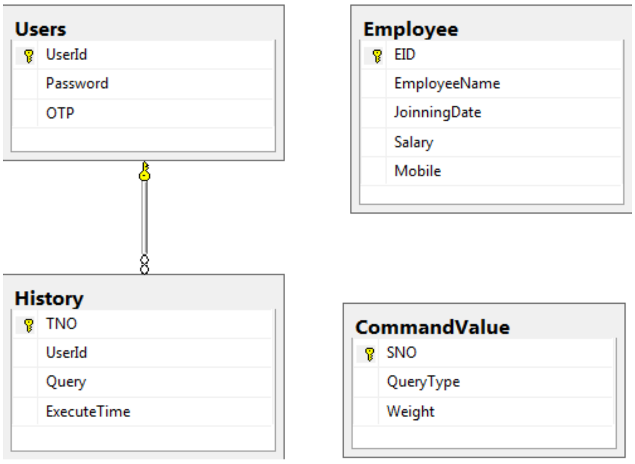
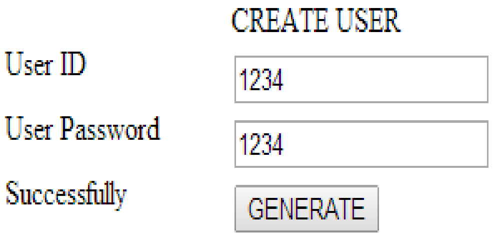


Figure 6: Database Diagram

In the create user step, the users will be create by enter the user name and password. This user can apply the queries based on the requirement.



CREATE USER

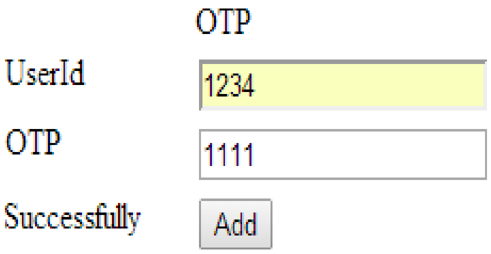
User ID 1234

User Password 1234

Successfully

Figure 7: Create Users

The OTP Password required to be generated by the admin required for update and delete the information because they have more weight age than the other queries.



OTP

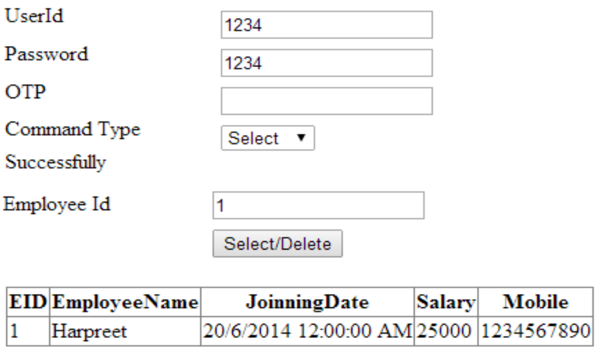
UserId 1234

OTP 1111

Successfully

Figure 8: OTP Password

For select the information there is no need of OTP password but required to enter the normal username and password provided by the admin at the time of registration.



UserId 1234

Password 1234

OTP

Command Type Select

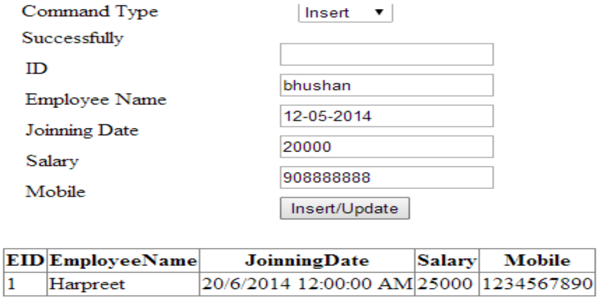
Successfully

Employee Id 1

EID	EmployeeName	JoinningDate	Salary	Mobile
1	Harpreet	20/6/2014 12:00:00 AM	25000	1234567890

Figure9: Select Information

The information will be retrieved from the database without any restriction and the intruder cannot apply any operation.



Command Type Insert

Successfully

ID

Employee Name bhushan

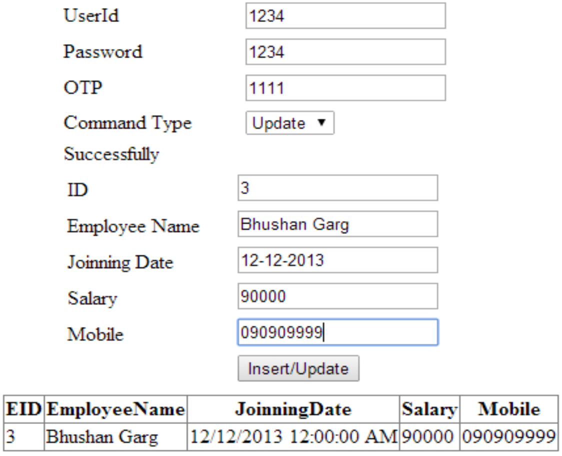
Joinning Date 12-05-2014

Salary 20000

Mobile 908888888

EID	EmployeeName	JoinningDate	Salary	Mobile
1	Harpreet	20/6/2014 12:00:00 AM	25000	1234567890

Figure 10: Insert new information



UserId 1234

Password 1234

OTP 1111

Command Type Update

Successfully

ID 3

Employee Name Bhushan Garg

Joinning Date 12-12-2013

Salary 90000

Mobile 090909999

EID	EmployeeName	JoinningDate	Salary	Mobile
3	Bhushan Garg	12/12/2013 12:00:00 AM	90000	090909999

Figure 11: Update User by OTP

VII. CONCLUSION AND FUTURE WORK

The research has been shown that IPS Techniques work good with database operation and the query will be execute after the permission of Admin OTP. This proves that IPS mechanism can be used for detection of malicious transactions in DBMS. The IPS mechanism uses query weightage and the assigned users can do operation accordingly. The database has been created for the prevention analysis. The IPS Mechanism works on different phases such identify the different transactions, assign the weights to the queries such as the update has greater weight and select has less weight, and then prevent from the intruder by OTP Mechanism provided by the Admin to the users. The transaction is Insert, Update, Delete, and Select. The proposed approach prevent from the intruder because complete work in under the surveillance of Admin and the OTP will be generated by the admin.

In future, this can be applied to the distributed environment where the multiple roles of users are present and different kind of queries can be executed in real time environment. The log based analysis approach can also be applied to detect that the users has been tried to inject the query and then, the prevention system automatically resolves the issues in the distributed real time environment.

REFERNCES

[1]Jianhua Lu et. al. (2012), "A Design of Solution to Database Security Based on Multi-Layer Intrusion Tolerance", Industrial Control and Electronics Engineering (ICICEE), Page(s): 1571 – 1574.

[2] Nedhal A. Al-Sayid (2013), "Database Security Threats: A Survey Study", International Conference on Computer Science and Information Technology (CSIT), IEEE.

[3]Peng Wang et. al (2013), "Design and Implementation of Security Enhanced Module in Database", Internet Computing for Engineering and Science (ICICSE), Page(s): 60 – 62.

[4] Shelly Rohilla (2013), "Database Security by Preventing SQL Injection Attacks in Stored Procedures", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 11.

[5]Sadeghian, A. ; Zamani, M. ; Ibrahim, S. (2013), "SQL Injection Is Still Alive: A Study on SQL Injection Signature Evasion Techniques",Informatics and Creative Multimedia (ICICM), IEEE, Page(s): 265 – 268

[6]Sadeghian, A. (2013), "A Taxonomy of SQL Injection Detection and Prevention Techniques", Informatics and Creative Multimedia (ICICM), Page(s): 53 – 56.

[7] Al-Sayid, N.A. and Aldlaeen D. (2013), "Database security threats: A survey study", IEEE, Computer Science and Information Technology (CSIT), 2013 5th International Conference, pp.60 - 64.

[8]Sadeghian, A., Zamani, M. Abdullah, S.M. (2013), "A Taxonomy of SQL Injection Attacks" Informatics and Creative Multimedia (ICICM), International Conference,IEEE,Page(s): 269 – 273

[9] Harshavardhan Kayarkar, "Classification of Various Security Techniques in Databases and their Comparative Analysis".