International Journal of Computer Engineering and Information Technology

VOL. 9, NO. 12, December 2017, 284–291
Available online at: www.ijceit.org
E-ISSN 2412-8856 (Online)

A Systematic Literature Review on SQL Injection Attacks Techniques and Common Exploited Vulnerabilities

Salem A. Faker[1], Mohamed A. Muslim[2] and Harry S. Dachlan[3]

[1] Computer Department, Tripoli University, Tripoli, Libya.

[2, 3] Electrical Engineering, Brawijaya University, Malang, Indonesia

[1]salemsamno679@gmail.com, [2]muh_aziz@ub.ac.id, [3]harrysd@ub.ac.id

ABSTRACT

Database faces several threats such as Cross Site Scripting Attack (XSS), phishing, Denial of Service (DoS) and SQL injection attack. SQL injection attack is the major concern and the most effective method for stealing the data from backend database, by the help of these attacks hacker can get access to the database and steal sensitive information. Although there are many review papers in the field, they are not systematic literature review (SLR), instead they are a normal survey or literature review. In contrast, this paper followed general guidelines for undertaking SLR in order to exploring the impacts and the types of SQLIA, illustrating the mechanisms and techniques of conducting SQLIA and exploring the most common type of SQLIA. An advanced search has been performed in most relevant digital libraries to obtain potentially relevant articles published until the end of 2016. Primary studies have been identified based on inclusion and exclusion criteria. The analytical study is mainly based on the primary studies to achieve the objectives. The results illustrate the impacts and the types of SQLIA, exploring the impacts and the types of SQLIA and illustrating the mechanisms and techniques of conducting SQLIA.

Keywords: *Database attacks, Systematic literature review, SQLIA.*

1. INTRODUCTION

Organizations use web applications with dynamic database to build a collaborative environment and for providing a better and a various service to their customers. The services could be online banking which holding very sensitive data, universities that uses a countless of students' results and different other government web applications. There are many attacks threatening database security such as Cross Site Scripting Attack (XSS), phishing, Denial of Service (DoS) and SQL injection attack [1, 2]. According to [2-7], SQL injection attack the major concern of web applications' developers because SQL injection attacks threaten the confidentiality, integrity, functionality and availability of back end-databases' of any web applications. Furthermore, SQL Injection Attacks are the most effective method for stealing the data from backend database, by the help of these attacks hacker can get access to the database and steal sensitive information. Finally, SQLIA usually ranked at the first one in the list of top 10 vulnerabilities in the Open Web Application Security Project [8]. There are several types, technique, and tools of SQL injection attacks.

This paper provides a systematic literature review on SQLIA in order to keep researchers up to date with the types, technique, and tools of SQL injection attacks. Systematic literature reviews (SLR) aims to identify, assess and combine the evidence from primary studies (PSs) using an explicit and rigorous method. In conducting this SLR, seven scientific online digital libraries were included in the search strategy, and potentially 608 relevant articles as a result of that search. After applying inclusion and exclusion criteria at the study selection stage as well as screening the selected articles at the assessing quality stage, identify 46 relevant primary studies (PSs) which were used for reporting the findings. We describe SLR methodology in Section 3 and present the finding results in Section 4. In Section 4 we answer the research questions. The conclusion is presented in Section 5.

2. RELATED WORK

There are many review papers on SQLIA. These review papers illustrate, define and explain the technique, tools and types of SQLIA. In [9] the authors survey the vulnerabilities in the context of Web Services. In addition, the authors discussed the general countermeasures for prevention and mitigation of SQLIA. Furthermore, [9] aimed to provide a web developer with the potential impact of general attacks of web applications based on attacks' category, level, spreading, size, deviation, dependencies, fundability and amplification. Whereas [10] survey the effective SQLIA detection and prevention techniques and [11] illustrates the popular input validation attacks includes SQL injection and Cross Side Scripting (XSS). Similarly, [12-16] discuss several types of SQLI attacks and Cross Site Scripting Attack, vulnerabilities, and different detection and prevention techniques. Although there are many review papers about types, technique, and tools of SQLIA, however, there is a lack of systematic literature reviews on SQLIA to keep researchers up to date with the types, technique, and tools of SQLIA. In contrast to existing review papers, this paper provides a systematic literature review (SLR) to systematically investigating the existing SQLIA techniques and briefly describe each technique by example. In addition, it aims to highlight the exploited vulnerabilities that used to inject a malicious code in order to provide web developers with a wide knowledge about the most exploited vulnerabilities. Furthermore, it aims to systematically discuss the impact of SQL Injection. Unlike to [1-5], this systematic literature review followed general guidelines for undertaking systematic literature reviews by following [17] approach which details a range of related work, provide a systematic and rigorous approach to illustrating types, technique, tools, of SQLIA.

3. RESEARCH METHOD

This systematic literature review performs four main phases; planning, conducting, reporting the result and discussing and interpret the results. Figure 1 graphically illustrate the phases involved in this SLR and the activities of each phase.

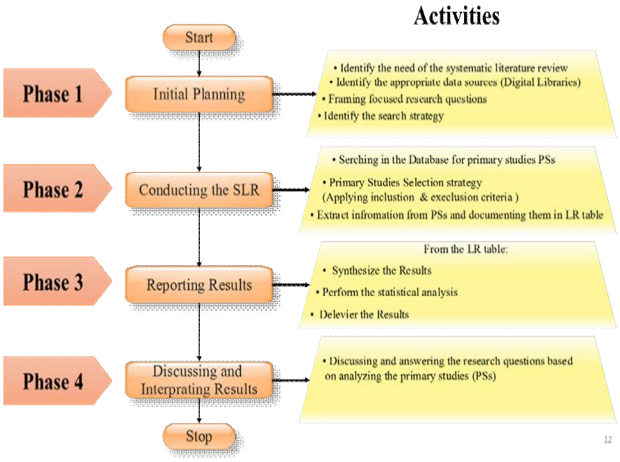
Activities
Start
Phase 1
Initial Planning
• Identify the need of the systematic literature review
• Identify the appropriate data sources (Digital Libraries)
• Framing focused research questions
• Identify the search strategy
Phase 2
Conducting the SLR
• Serching in the Database for primary studies PSs
• Primary Studies Selection strategy (Applying inclustion & execlusion criteria)
• Extract infromation from PSs and documenting them in LR table
Phase 3
Reporting Results
From the LR table:
• Synthesize the Results
• Perform the statistical analysis
• Delevier the Results
Phase 4
Discussing and Interprating Results
• Discussing and answering the research questions based on analyzing the primary studies (PSs)
Stop
12

Fig. 1. Research Phases

In planning phase, initially we identify the need for a systematic literature review, then framing focused research questions. While in the second phase, we initially searched the databases for primary studies, after that the articles are evaluated for relevance and quality, then extracted data from the primary studies. In the reporting phase, the results are synthesized, analyzed and reported. The reporting phase explores the protocol that has been followed to review and reduce the chances of researcher bias. The protocol includes identifying the research questions, defining the search strategy, determining the study selection, and the study quality assessments.

3.1 Research Questions

Based on the primary studies that have been determined by the study selection process, this SLR attempts to answer the following research questions (RQs):

RQ1: What are the impacts of SQLIA?
RQ2: What are the types of SQLIA?
RQ3: What are the Mechanisms of SQLIA?

The first RQ1 and RQ2 are motivated by the desire of exploring the impacts and the types of SQLIA. While RQ2 is motivated by the desire of illustrating the mechanisms and techniques of conducting SQLIA. Whereas, the RQ3 is motivated by the desire of exploring the most common type of SQLIA.

3.2. Research Strategy

To obtain a comprehensive list of articles about SQLIA we conducted an advanced search in most popular and relevant digital libraries that contain peer-reviewed journal articles, conference proceedings, and book chapters. The selected databases include PubMed, ACM, Springer, Scopus, IEEE, ISI, Google scholar, and Science Direct. The selection of these libraries increases confidence in the completeness of the review. In addition, to increase the comprehensiveness of this systematic literature review the searching years were specified from the year 2010 up to the end of 2017. Besides that, the search string was constructed based on the following factors:
The major terms extracted from the research questions.
Alternative spellings and synonyms of the major terms.
Research keywords appeared in other relevant papers (e.g. [12-16]).
Boolean (AND) was used to connect the major research terms and Boolean (OR) used to connect alternative spellings and synonyms of the keywords.

3.3 Study Selection

This section explains the method of selecting PSs from relevant articles. An inclusion and exclusion criteria were applied to focus on reviewing types, technique, and tools of SQLIA only. In inclusion criteria, the PSs have to be published in peer-reviewed journals or conference proceeding. In addition, the articles should be related to SQLIA. Meanwhile, all articles that did not fulfill the inclusion criteria were excluded. Inclusion and exclusion criteria are important requirements and must be satisfied in all selected articles to ensure that the selected PSs were within the related and targeted area of research.

4. RESULT

In this section, we provide an overview on the SQLIA then answering the research questions based on analyzing the primary studies (46 articles) that identified in accordance with the review protocol.

4.1 Overview on the SQLIA

SQL is stand for Structure Query Language and originally developed in the early 1970’s by Edgar. SQL is the most common high level language used in various relational Database Management Systems (DBMS) for control and build the desired query commands [4]. SQL is used for accessing database servers, including MySQL, Oracle, and SQL Server. Web programming languages such as Java, ASP.NET, and PHP provide various methods for constructing and executing SQL statements [18]. SQL language is a communication way between users and database in order to allow the user to interaction with a database for managing data held in a relational database management system [19]. SQL statements can modify the structure of databases (using Data Definition Language statements or 'DDL') and manipulate the contents of databases (using Data Manipulation Language statements, or 'DML') [20]. According to [2] web applications could be a target of SQL Injection Attack when (i) accept inputs from user or system, (ii) concatenate input with SQL statement and builds complete query structure, and (iii) executed query concatenated with HTML code.

4.2 RQ1: What are the impacts of SQLIA?

SQL injection attack occurs when data comes from untrusted source or use incoming data dynamically to construct SQL query to gain access to a database for viewing or manipulating data for different goals [4] [3] [21].

SQL injection can result in authentication bypass, data loss, denial of access, and also it may lead to destruction of whole database or the host takeover.

A successful SQL injection attacks can be disastrous, because attacks can read a sensitive data from the database, modify database by (insert/ modify/delete), change database structure, execute administrative operation (such as shutdown the database) or recover the contents of the DBMS files system and execute commands (such as XP cmd shell) to the operating system [22]. The main consequence of these vulnerabilities are:

1. Confidentiality: Loss of confidentiality is a major problem caused by SQL Injection attacks since databases hold sensitive and crucial information which could be viewed by unauthorized users as a consequence of successful SQL injection attack [4] [5].
2. Integrity: Successful SQL injection attack allows external source to make unauthorized modifications such as altering or even deleting information from targeted databases [23] [24] .
3. Authentication: Poorly written SQL queries do not properly validate user names and passwords, which allows unauthenticated entity or attacker to access to the database or application as an authenticated user, without initial knowledge of the password or even user name [3].
4. Authorization: Successful exploitation of SQL injection vulnerability, allows attacker to change information and gain elevated privileges if the authorization information is stored in the affected database [2].
5. Functionality: Concurrent processing is an important feature in any database processing for enables users’ simultaneous access, sharing data, and consistent updates of databases. SQL injection attack could partially or fully corrupting Database intended function [6] [25].

4.3 RQ2: What are the types of SQLIA?

Several types of SQL injection attack can be characterized based on the goal, or intent of the attacker. The most common and well-known SQL Injection attacks based on the goals of the attackers are summarized in Table 2.

Table 2: SQL injection attack based on attacks' intend

Ref.	Classification	Techniques/ implementation
[26] [6] [3] [26] [27] [28]	Identifying injectable parameter	To discover which parameters and user-input fields are vulnerable to SQLIA
	Extracting data	To extract data values from the database
	Adding or modifying data	To add or change information in a database
	Performing DOS	To shut down, locking or dropping database
	Evading detection	To avoid auditing and detection by system protection mechanisms
	Bypassing authentication	To bypass database and application authentication mechanisms
	Executing remote commands	To execute arbitrary commands on the database including stored procedure or functions
	Performing privilege escalation	To escalate the privileges by implementation errors or logical flaws in the database
	Determining database schema	To obtain information about (table names, column names, and column data types, etc.) using penetration testers and vulnerabilities scanners

Whereas Table 3 summarizes the most common SQLIA based on input mechanisms.

Table 3: SQL injection classified based on input mechanisms

Ref.	Classification		Mechanisms
[3] [27] [28] [29] [30] [31]	First order injection	Through user input	Injects malicious SQL commands into user input query in web form based on GET and POST.
		Through cookies	Modified cookies fields containing SQLIA
		Through server variables	Manipulated header containing SQLIA
	Second-order injection	occurs when data input stored in a place and then used in a different SQL Query without correct filtering or without using parameterized queries	Frequency-based primary applications
			Frequency-based secondary applications
			Secondary support application
			Cascaded submission application

Table 4: summarize SQLIA based on the mechanisms of attacks

Ref.	Classification type	Techniques/ implementation	
[4] [6] [32] [3] [7] [2]	Classic SQLIA	Piggy-backed	Insert additional queries to be executed
		Tautologies	Create a query that always evaluates to true
		Alternate encodings	Encode attacks in such a way as to avoid naive input filtering e.g. **user= 41444d494e** instead of user= ADMIN
		Illegal/Logical	Using error messages rejected by the database to find useful data
		Union	Injected query is joined with a safe query using the keyword UNION
		Stored procedures	executes built-in procedure or functions
	Inference	Blind SQLIA (True/False)	Conditional response
			Conditional error
			Out-of-band channeling
		Timing SQLIA (If/Then)	Double Blind (Time-Delays)
			Deep Blind (multiple statements)
	DBMS specific SQLIA	DB fingerprinting (e.g. DMBS version and host OS)	
		DB mapping	
	Compounded SQLIA	Fast-Fluxing SQLIA	

4.4 RQ3: What are the Mechanisms of SQLIA?

There are varieties of SQL injection mechanism in Table 4 summarize SQLIA based on the mechanism attacks that hackers can try to achieve the purpose of the hacking.

In this question, most of them are explained in wide details as following:

1. Tautologies: used to inject code in one or more conditional statements so that they always evaluate to true. This method occurs in the Absence of checking inputted data that goes to database. The following code is an example of such a dynamic SQL statement. query = "SELECT info FROM user WHERE name = "name" AND pwd = "pwd"; Attackers can use tautologies to exploit this peace of code by supplying the value (x' OR '1'='1') to the input parameter name. An attacker could access user information without a valid account because the WHERE clause condition becomes (WHERE name = 'x' OR '1'='1' --;) this make the system evaluates the result to be true and to terminates the rest of the Query using (--;).
2. Piggy-backed: In this case, the attackers insert additional queries to be executed by the database in order to extracting data, adding or modifying data, performing denial of service or executing remote commands [28]. In this case, attackers are not trying to modify the original query. In fact, they are trying to add extra and distinct query that added to the original query using private words based on SQL language such as OR, AND, INSERT, UPDATE, DROP or DELETE in order to allow database to receive multiple SQL queries [10].
3. Alternate Encodings: the main goal of hackers when they use this method is to evading detection. In addition, this attack type is used for encoding the attack strings (e.g., using hexadecimal, ASCII, and Unicode character encoding) in order to escape from developer filtering. In fact, alternate encodings usually combined with other attack techniques and targets different layers in the application [4]. For example, attacker use char (44) instead of single quote that is a bad character. The char () function and ASCII hexadecimal encoding are combined in this attack. The char () function takes hexadecimal encoding of character(s) and returns the actual character(s).
4. Illegal/Logical: In this attack, an attacker tries to inject statements that cause a syntax error page returned by application servers for identifying injectable parameters, performing database finger-printing and extracting data of the back-end database of a Web application [33]. In fact, error page provides hackers with information about few details name of tables as instance or reveal vulnerable/injectable parameters to an attacker,

which may those information used for conducting next step of attack [34].

5. Union query: In this method of attack Injected, query is joined with original query using the keyword UNION in order to get information related to other tables from the database. With the help of this type of attack, attacker can extract data type or information about the columns [35]. By default, Most SQL-compliant databases, including SQL Server, store metadata in a series of system tables with the names sysobjects, syscolumns, sysindexes, and so on. This means that a hacker could use the system tables to ascertain schema information for a database to assist hackers for further compromise to the database.
6. Stored Procedures: In this method, the attacker executes built-in functions using malicious SQL codes for performing privilege escalation, performing denial of service or executing remote commands. In fact, most database providers develop databases with a standard set of stored procedures and functions for extending the functionality of the database and allow to them to interact with the operating system. Therefore, once an attacker determines which backend-database is in use, SQLIAs can be crafted to execute stored procedures provided by that specific database [28] [12].
7. Inference: An attacker derives logical conclusions from the answer to a true/false question concerning the response of database server. This method consist of two type Blind injection and timing injection [3]. In Blind injection, hackers collect information about database by inferring from the replies of the page after questioning the server true/false questions. If the answer is true then the application behaves correctly and if the answer is false then it cause an error. Therefore, attacker can get indirect response from database [32].

5. DISCUSSION

This section discusses and interprets the results reported in Section 4.

5.1 The impacts of SQLIA (related to RQ1)

In real world scenario, it is very hard to detect the SQL injection prior to its impact. Most time, unauthorized activity is performed by the attacker through valid user credentials or by using inherent features of database application such as malicious modification of existing SQL Queries of web application that are accessing critical sections of the databases [2]. SQL injection attacks can be disastrous, because attacks can read a sensitive data from the database, modify database by (insert/ modify/delete), change database structure, execute administrative operation. However, SQLIA can extremely affects confidentiality, Integrity, Authentication, Authorization, Functionality, usability and availability.

5.2 The types of SQLIA (Related to RQ2)

The result reported in section 4.3 show that there are several types of SQLIA can be used to achieve malicious aims. Since 2002 until 2014 there are numerous of methods have been installed and used to protect and prevent SQL injection attack on database with different analyzing approach either in static, dynamic mode or in hybrid mode. However, SQLIA still reported by many organization.

5.3 The Mechanisms of SQLIA (Related to RQ3)

The result reported in Section 4.4 show that, different mechanisms can be used for conducting SQLIA. However, in this paper seven common mechanisms were highlighted and explained.

6. CONCLUSION

Systematic literature reviews aim to identify, assess and combine the evidence from primary research studies using an explicit and rigorous method. This method has been widely implemented in software engineering and computer science. In this paper, a systematic literature review conducted to investigate the current state of knowledge about SQLIA. 46 primary studies have been identified in accordance with our review protocol and published between 2010 to the end of 2017. The major contributions of this paper can be concluded as:

- Detailing an obvious range of related work, search strategy and study selection for relevant studies in the field of SQLIA.
- A systematic, evidence-based, and rigorous approach to conducting and reporting the result of the research question.
- Providing a systematic literature review instead of a normal review.
- Exploring the impacts and the types of SQLIA
- illustrating the mechanisms and techniques of conducting SQLIA
- Exploring the most common type of SQLIA

A lack of systematic literature reviews in this field to keep researchers up to date with the state of research in the area encourage authors to continue the evaluation and improvement of this approach by conducting SLR on the techniques that used in preventing and detecting SQLIA.

7. ACKNOWLEDGMENT

The authors would like to express greatest appreciation to Electrical Engineering, Faculty of Engineering, Brawijaya University, Malang, Indonesia for support. Furthermore, full thanks to be supported by Ministry of Defense, Libya under the MSc. scholarship programs.

REFERENCES

[1] Sarasan, S., Detection and Prevention of Web Application Security Attacks. International Journal of Advanced Electrical and Electronics Engineering, (IJAEEE), 2013. 2.

[2] Johari, R. and P. Sharma. A Survey on Web Application Vulnerabilities (SQLIA, XSS) Exploitation and Security Engine for SQL Injection. in Communication Systems and Network Technologies (CSNT), 2012 International Conference on. 2012.

[3] Borade, M.R. and N.A. Deshpande, Extensive Review of SQLIA‘s Detection and Prevention Techniques. International Journal of Emerging Technology and Advanced Engineering, 2013. 3(10).

[4] Kindy, D.A. and A.-S.K. Pathan, A Detailed Survey on various aspects of SQL Injection in Web Applications: Vulnerabilities, Innovative Attacks and Remedies. International Journal of Communication Networks & Information Security, 2013. 5(2).

[5] Kindy, D.A. and A.-S.K. Pathan, A survey on SQL injection: Vulnerabilities, attacks, and prevention techniques. 2011.

[6] Nithya, V., R.Regan, and J.vijayaraghavan, A Survey on SQL Injection attacks, their Detection and Prevention Techniques. International Journal Of Engineering And Computer Science, 2013. 2(4): p. 886-905.

[7] Sadeghian, A., M. Zamani, and A.A. Manaf, A Taxonomy of SQL Injection Detection and Prevention Techniques. International Conference on Informatics and Creative Multimedia, 2013: p. 53-56.

[8] (OWASP), T.O.W.A.S.P. The Ten Most Critical Web Application Security Vulnerabilities. 2013 [cited 2014 april]; Available from: https://www.owasp.org/index.php/Main_Page

[9] Jensen, M., N. Gruschka, and R. Herkenhöner, A survey of attacks on web services. Computer Science-Research and Development, 2009. 24(4): p. 185-197.

[10] Kumar, P. and R. Pateriya. A survey on SQL injection attacks, detection and prevention techniques in Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference on. 2012. IEEE.

[11] Chaudhari, X.G.R. and M.V. Vaidya, A Survey on Security and Vulnerabilities of Web Application. 2014 IJCSIT, 2014.

[12] Srivastava, S., A Survey On: Attacks due to SQL injection and their prevention method for web application. 2012.

[13] Jensen, M., N. Gruschka, and R. Herkenhöner, A survey of attacks on web services. Computer Science - Research and Development, 2009. 24(4): p. 185-197.

[14] Johari, R. and P. Sharma. A Survey on Web Application Vulnerabilities (SQLIA, XSS) Exploitation and Security Engine for SQL Injection. 2012: p. 453-458.

[15] Li, X. and Y. Xue, A survey on server-side approaches to securing web applications. ACM Computing Surveys, 2014. 46(4): p. 1-29.

[16] Agarwal, U., M. Saxena, and K.S. Rana, A Survey of SQL Injection Attacks. International Journal of Advanced Research in Computer Science and Software Engineering, 2015. 5(3).

[17] Kitchenham, B. and S. Charters, Guidelines for performing systematic literature reviews in software engineering. 2007, EBSE Technical Report EBSE-2007-01: Keele University, UK, 2007.

[18] Shar, L.K. and T. Hee Beng Kuan, Defeating SQL Injection. Computer, 2013. 46(3): p. 69-77.

[19] Katsaropolous, C. and H. Scherer, SQL Injection Attacks and Defense 2nd edition, ed. 2. 2012, USA.

[20] Clarke, J., SQL injection attacks and defense. 6 ed. 2012: Elsevier. 494.

[21] Awang, N.F. and A.A. Manaf, Detecting Vulnerabilities in Web Applications Using Automated Black Box and Manual Penetration Testing, in Advances in Security of Information and Communication Networks. 2013, Springer. p. 230-239.

[22] Tajpour, A., S. Ibrahim, and M. Masrom, SQL Injection Detection and Prevention Techniques. International Journal of Advancements in Computing Technology, 2011. 3(7): p. 82-91.

[23] Huang, Y.-W., et al. Securing web application code by static analysis and runtime protection. in Proceedings of the 13th international conference on World Wide Web. 2004. ACM.

[24] Sajjadi, S.M.S. and B. Tajalli Pour, Study of SQL Injection Attacks and Countermeasures. International Journal of Computer and Communication Engineering, 2013: p. 539-542.

[25] Al-Khashab, E., F.S. Al-Anzi, and A.A. Salman. Preventing SQL injection attacks based on query optimization process. in Second Kuwait Conf. on E-Services and E-Systems. 2011. Kuwait ACM.

[26] Halfond, W.G.J., J. Viegas, and A. Orso, A Classification of SQL Injection Attacks and Countermeasures. IEEE, 2006.

[27] Halder, R. and A. Cortesi. Obfuscation-based analysis of sql injection attacks. in Computers and Communications (ISCC), 2010 IEEE Symposium on. 2010. IEEE.

[28] Halfond, W.G.J., J. Viegas, and A. Orso. A classification of SQL injection attacks and countermeasures. in Proceedings of the IEEE International Symposium on Secure Software Engineering, Arlington, VA, USA. 2006.

[29] Bhagat, M.A. and V. Mane, PROTECTION OF WEB APPLICATION AGAINST SQL INJECTION ATTACK. International Journal of Scientific and Research Publications, 2013. 3(10).

[30] Swetha, D.N. and B.S. Kumar, Protocol Based Approach on Vulnerability Detection Tools of SQLIA along with Monitoring Tools. IJCSET, 2013. 2(11).

[31] Bravenboer, M., E. Dolstra, and E. Visser. Preventing injection attacks with syntax embeddings. in Proceedings of the 6th international conference on Generative programming and component engineering. 2007. ACM.

[32] Khochare, N., et al., Survey on SQL Injection attacks and their Countermeasures. IJCEM International Journal of Computational Engineering & Management, 2011. 14.

[33] Khari, M. and N. Kumar, SQLIA Detection And Prevention Approaches A Survey. International Journal of Computer Science & Information Technology, 2013. 3(5).

[34] K, V.K. and J.D. .D, Advanced Detecting and Defensive Coding Techniques to prevent SQLIAs in Web Applications A Survey. International Journal of Science and Modern Engineering (IJISME), 2013. 1(6).

[35] Roy, S., A.K. Singh, and Ashok Singh Sairam, Detecting and Defeating SQL Injection Attacks. International Journal of Information and Electronics Engineering, 2011. 1(1).