I.J. Wireless and Microwave Technologies, **2016, 6, 72-79**
Published Online November 2016 in MECS(http://www.mecs-press.net)
DOI: 10.5815/ijwmt.2016.06.08

Available online at http://www.mecs-press.net/ijwmt

A Survey on Detection and Prevention Techniques for SQL Injection Attacks

Harish Dehariya [a], Piyush Kumar Shukla [b], Manish Ahirwar [c]

[a] *Scholar, Department of Computer Science and Engineering, UIT-RGPV, Bhopal, 462036 India*
[b] *Assistant Professor, Department of Computer Science and Engineering, UIT-RGPV, Bhopal, 462036 India*
[c] *Assistant Professor, Department of Computer Science and Engineering, UIT-RGPV, Bhopal, 462036 India*

Abstract

In this current scenario web application are widely using for various purpose like online shopping, online money transfer, e-bill payment, online mobile recharges etc. As per increasing the dependency on these web applications also raises the attacks on these applications. SQL injection Attacks (SQLIA) and Cross Site Scripting (XSS) are being a major problem for web applications. SQL injection Attack (SQLIA) is the most common type of vulnerability in which a malicious mind person is inserts its own crafted query as input for retrieving personal information about others sensitive users. In this paper, for detection and prevention of SQL injection attacks various techniques are described and perform a comparison between them.

Index Terms: SQL Injection Attack, Web Application, Vulnerabilities, Detection and Prevention techniques.

1. Introduction

Web application are using for storing and retrieval for information. Various organizations use these applications for communication purpose for increasing their business and users stores there sensitive information in database. These web applications are developed in scripting languages and communication made with database through SQL queries. SQL is a communication medium between Web application and back end database. An attacker may harm the Database. So the attackers can use SQL for of breaking confidentiality purpose of database by accessing it. A SQL injection attack (SQLIA) is that in which a malicious mind person injects its own crafted query as input. The vulnerability exists within web application when a Web application does not provide proper validation or filtering for the input data entered by the user in the Input fields. The backend server executes the query statement and sends the result to the attackers. An attacker obtains the result. This SQL Injection Attack mostly affects financial web applications or secret information.

* Corresponding author.
E-mail address:dehariyah07@gmail.com, pphdwss@gmail.com, ahirwarmanish@gmail.com

In the paper list of cause of SQL Injection Attacks and list of types of SQLIAs is describe in tabular from also describe the detection and prevention, tools that have been evolved till the current time. Binary Evaluation Approach [1] based on identifying trusted info source, by using dynamic tainting to track trustworthy data with runtime and give only permission to those quires which are semantically correct. One of the tools is Vulnerability &Attack Injector Tool (VAIT) [4] that is based upon the injection of realistic vulnerabilities and the controlled exploit of those vulnerabilities in order to attack the web applications. It uses for checking the vulnerability in codes and evaluation of counter measure in IDS and effectiveness of web application vulnerability scanners.

The WebSSARI [35] tool uses predefined set of filters for filtering inputs for finding security vulnerabilities in the application. This is quite useful because it can detect stored procedure with other SQLIAs. Another approach is SQLRand [32] which is based on a randomization query algorithms. Here a strong random integer is inserted in the SQL keywords. In SQL DOM a set of classes that are strongly typed to a database schema are used to generate SQL statements instead of string manipulation.

SQL CHECK [25] is based on a key inserted at both beginning and end of user's input and invalid syntactic forms are the attacks.

SQLGAURD [30] proposed a model of expected queries at runtime. In this approaches, the model is expressed as a grammar that only accepts legal queries. Static analysis algorithm is derived from type system and type state for a secure information flow. At the time of analysis codes checks for any vulnerability without user intervention.

In the figures two processes are showing in figure 1 showing that how a normal user is accessing a web application. Here user wants to access his bank account by online banking system.

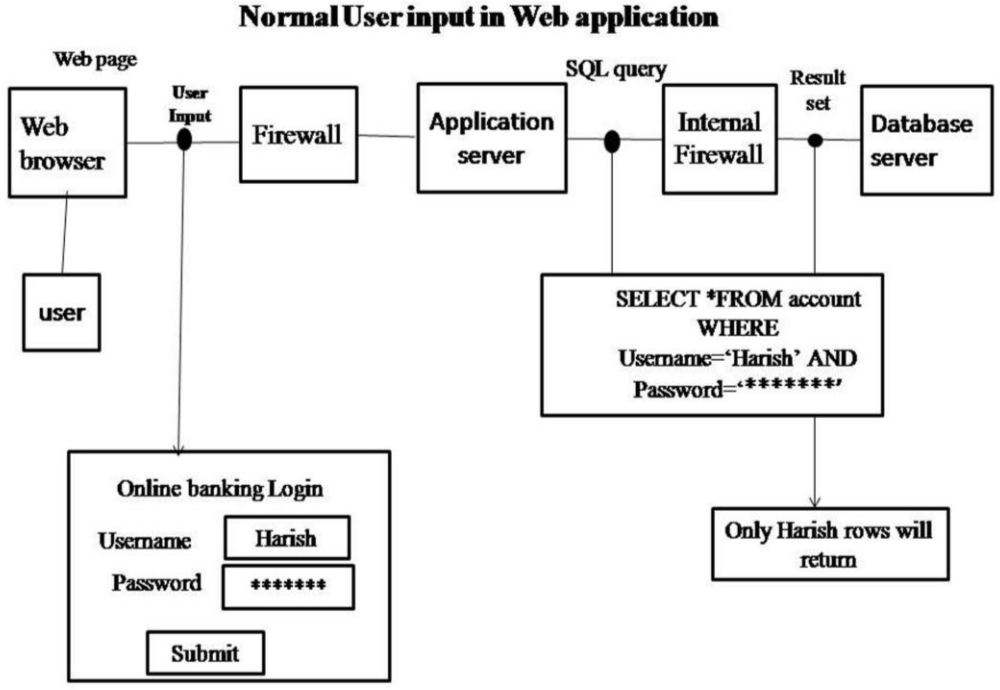

Fig.1. Normal User input in Web application et al [12]

For accessing following steps are required.

1. Normal User by using web browser uses a web application of a bank and give username and password.
2. After submission of input firewall checks formatting of given input.
3. Now Application server checks the input field is in correct or not.
4. This input converts into query form and transfer to the database server for checking authorization.
5. When database check that the user is authorised so now execute the query and generates the result.

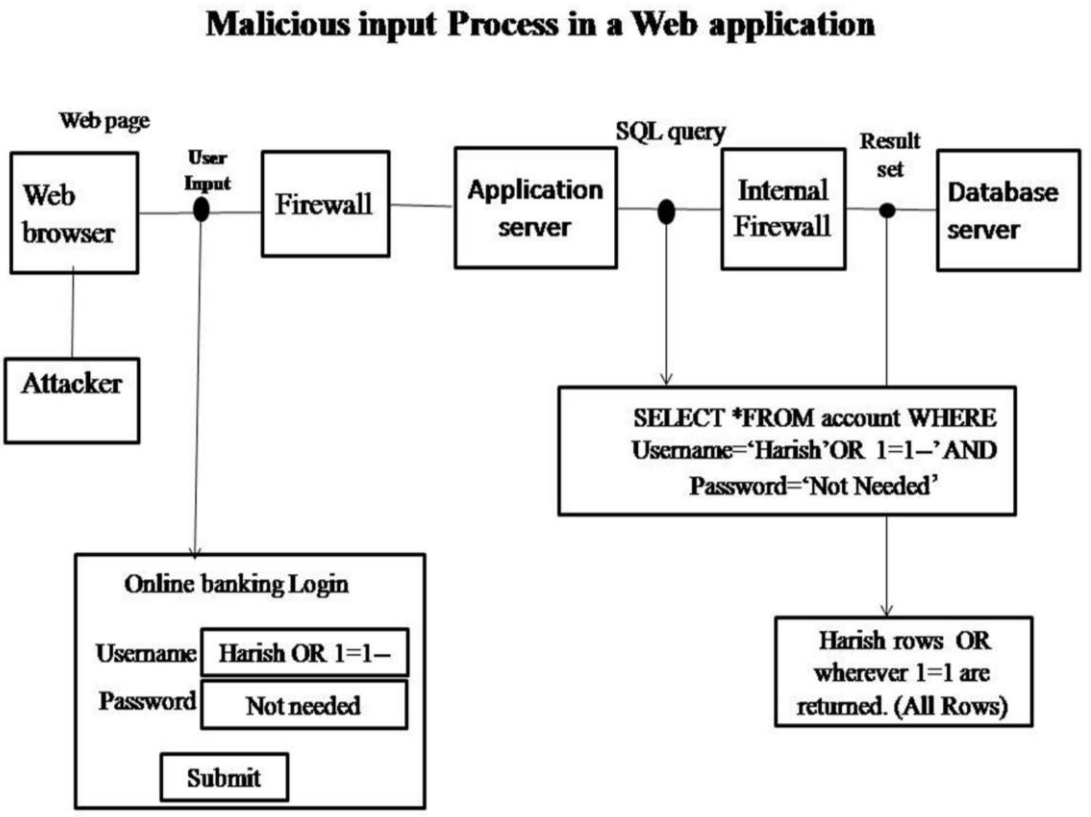

Fig.2. Process of Malicious User input in Web application et al [12]

This figure is showing SQL injection attack.

1. Attacker insert fake input commands in username, in case of above figure it is OR 1=1—this is command for Tautology attack, which means all inputs are true and upcoming conditions will be executes.
2. Tautology command gets the authorization for execution of query.
3. Application server passes this query to the database server for execution.
4. Database server generates the result of query after execution and shows to the attacker.

2. Causes of Sql Injection Attacks

Table 1. List of various causes by which SQL injection become Attacked.

CAUSES	PERPOSE OF ATTACKS
Invalidated Input	SQL query consist parameters such as INSERT, UPATE, ALTER, semicolon and quotation mark. If there is no checking for web applications so it can be abused in SQL injection.
Uncontrollable variable size	If any variable is using storage for large amount of data so some time can be possible that attacker may enter faked input values.
Error Message	Error message generates when wrong input values is inserted in web application. Attacker may get the script structure or information about database .so that attacker may create its own attack.
Clint side only control	If input validation is implemented in client side scripts only, then by using cross site scripting security function of script at client side can be override and attacker can invalidate input or accessing database.
Sub-select:	When a SQL query is inserted in WHERE clause of other SQL query so this is a weakness for Database. This weakness makes the web application more vulnerable
Into Outfile support	A text file of containing SQL query result may get by manipulating SQL query. This can be possible by using condition of INTO OUTFILE clause that is benefit of Some Relational Database.
Stored Procedure	Stored Procedure is a small program with some function that calls multiple times in execution. When these functions become calls so that stored procedure become calls in place of that functions. The problem with the stored procedure is that an attacker can execute and damage database
Generous Privileges:	Privileges are some rules for accessing some database for some object and by object which actions going to be perform. Here SELECT, INSERT, DELETE can include typical privileges. Bypass authentication an attacker gain privileges.

Table 2. List of Various Types of SQL injection Attacks with the basic commands which uses for SQLIA et al [10, 12, 17]

Name of Attacks	Basic commands	Example of Malicious query
Tautologies	Statement "**1=1"**	"SELECT * FROM the employee WHERE username = 'Harish' and password ='aaa' OR '1'='1'"
Illegal /Logically Incorrect Queries	An attacker may get the various parameters from errors message which generates on wrong query. So these parameters may help for creating new Query	URL http://www.onlinetickitbooking.it/event/?id_num=12) is original but in place type http://www.onlinetickitbooking.it/event/?id_num=12'3)
Union Query	In this type of attacker join a new query in original query by using **UNION** keyword and can get data tables from database.	*SELECT Name, Phone FROM Users WHERE Id=1 UNION ALL SELECT creditCardNumber,1 FROM CreditCardTable*
Blind Injection	Attacker can get the information of database structure by asking true/false type of questions through SQL statements, when developers hide the error details.	SELECT name FROM table WHERE login id= 'harish' and 1 =0 -- AND pass = SELECT name FROM table WHERE login id= 'harish' and 1 = 1 -- AND pass =
Timing Attacks	**WAITFOR** keyword is used for delay response by database	declare @ varchar(800) select @ = db_nameO if (ascii(substring(@, 1, 1)) & (power(3, 0))) > 0 waitfor delay '0:0:15'
Piggy-backed Queries	It uses "**;"** to append crafted query to original query	*SELECT info FROM users WHERE login='doe' AND pass=''; drop table users*—'AND pin =234
Stored Procedure	It coded by programmer it can also be uses as attack. Depend on specific stored procedure on the database there are different ways to attack.	*CREATE PROCEDURE DBO.is Authenticated @userName varchar2, @pass varchar2, @pin int AS EXEC("SELECT accounts FROM users WHERE login='" +@userName+ "' and pass='" +@password+ "' and pin=" +@pin); GO*

Table 3. A Comparison between different tools and used methodology.

TOOLS	TECHNIQUE FOR DETETCION AND PREVENTION	DRWABAKES
VAIT[4]	Create Malicious web page of application and then monitor by VAIT tool	For some applications it can detect all attacks but not successful for all web applications.
IDS[12]	A specification based approach to detect malicious intrusions	Only helpful for typical set of queries
DIWeDa[20]	To detect various types of intrusions in Web Databases applications	It is not useful for detection of all attacks
CSSE[24]	Context Sensitive Analysis	They require modification to the runtime environment which affect portability
SQL CHECK[25]	A key is inserted at both beginning and end of user's input. Invalid syntactic forms are the attacks. The key strength is a major issue	Discovery of Secret key by attacker
SQL DOM[26]	A set of classes that are strongly-typed to a database schema are used to generate SQL statements instead of string manipulation	Detection of Stored procedure cannot possible
SecuriFly[27]	Check input string instead of Character	Numeric fields cannot stop
Java Static Tainting[28]	It find error in JAVA code with static analysis	Generates high amount of false positive
Java Dynamic[29] Tainting	Track taint information on per-string basis	This technique still not applies on all realistic web applications.
SQL GUARD[30]	The parse trees of the SQL statement before and after user input are compared at a run time. The Web script has to be modified	Discovery of Secret key by attacker can be possible
Web Application Hardening[31]	Context Sensitive Analysis	They require modification to the runtime environment which affect portability
SQLRand[32]	A strong random integer is inserted in the SQL keywords.	Implementations of proxy server for randomize and de-randomization adds to performance overhead.
Tautology Checker[33]	Static analysis combined with automated reasoning for verify tautology	It can only detect tautology attacks
JDBC-CHECKER[34]	Type mismatch in a Dynamically generated query string	Mostly SQL injection Attack queries become syntactically correct
WebSSARI[35]	Check Input Validation against preconditioning	Some precondition may not be accurately expressed for some filter.
WAVES[36]	Uses Crawler for identification various points in web application.	Some point may not be identify which attacker use for attacks
Security Gateway[37]	use proxy filter system	If input validation rules breaks

3. Conclusion

SQL injection attack is a major attack for security of web applications. For security of web application many tools have been developed for detection and prevention. In these most of the tools can attack some types of SQLIAs. Tautology checker, SQL DOM [26] SQLRand [32] CSSE [24], SQL Check [25] SQLGaurd [28] etc. are the tools which can detect Attacks but not all. Stored procedure is one type which can detect by only one tool that is WebSSARI [35]. VAIT [4] tool is new one which tested on only three open sources applications. But it need more test cases for improvements. Security of web application by SQLIA is an area where a tool requires by which all types of attacks may detect and prevent.

Reference

[1] Venkatramulu Sunkari, Dr. C.V.Guru Rao: Protect Web Applications against SQL Injection Attacks

Using Binary Evaluation Approach, International Journal of Innovations in Engineering and Technology (IJIET), Volume 6 Issue 4 April 2016.

[2] Chandershekhar Sharma, Dr. S. C. Jain, Dr. Arvind K Sharma: Explorative Study of SQL Injection attacks and Mechanisms to secure web application database –A Review International Journal of Advanced Computer Science and Applications, (IJACSA) Vol. 7, No. 3, 2016.

[3] Anuj Dakwala, Kruti Lavingia: A Machine learning approach to improve the efficiency of Fake websites detection Techniques, International journal of computer Science and Communication (IJCSC) Vol. 7, no. 1, PP 236-243, March 2016.

[4] Jose Fonseca, Marco Vieira and Henrique Madeira: Evaluation of web security Mechanism using Vulnerabilities and Attack Injection IEEE Transactions on Dependable and secure computing Vol. 11 no 5 September/October 2014.

[5] Nuno Seixas, Marco Vieira, Jose Fonseca, Henqirque Madeira*:* Analysis of field data on web security vulnerabilities, IEEE Transactions on Dependable and secure computing Vol. 11 No.2 March/Aril 2014.

[6] www.owasp.org/index.php/SQL_Injection

[7] Hussein AlNabulsi, Izzat Alsmadi, Mohammad Al-Jarrah: Textual Manipulation for SQL Injection attack, I.J. computer Network and Information Security, 2014.

[8] Hossaian Shahriar, Mohammad Zulkernine: Information Theoretic Detection of SQL Injection Attacks, International Symposium on high-Assurance system s Engineering, IEEE 2014.

[9] Jaskanwal Minhas Raman Kumar: Blocking of SQL Injection attack by Comparing Static and Dynamic queries, International Journal of computer network and Information Security 2013.

[10] Monali R. Borade, Neeta A. Despande: Extensive Review of SQLIA's Detection and Prevention Techniques, International Journal of Emerging Technology and Advanced Engineering ISSN 2250-2459, ISO 9001:2008 Certified Journal, Vol;ume3, Issue 10, October 2013.

[11] Iyano Alessandro Elia, Jose Fonseca and Marco Vieira: Computing SQL Injection Detection Tools Using Attack Injection: An Experimental study IEEE International Symposium on software reliability Engineering 2012.

[12] Atefeh Tajpour, Suhaimi Ibrahim, Mohammad Sharifi: Web Application security by SQL Injection Detection tools, International Journal of Computer science, Issue, Volume 9 Issue 2 No 3 March 2012.

[13] Srinivas Avireddy, Varalaxhmiperumal, Narayan Gowraj, Ram Srivastava Kannan, Random4: An Application Specific Randomized Encryption Algorithm to prevent SQL Injection" 11th International conference on trust, Security and privacy in computing and communications, IEEE 2012.

[14] Inyong Lee, Soonki Jeong, Sangsoo Yeo, Jongsub Moon: A novel method for SQL Injection attack detection based on removing SQL Query attribute values, ELSEVIER 2012.

[15] Kanchana Natrajan, Sarala Subramani: Generation of SQL injection free secure algorithm to detect and prevent SQL Injection attack, ELESEVIER C3IT-2012.

[16] I. Elia, J. Fonseca and M. Vieira: Comparing SQLi Detection Tools using attack Injection: An Experimental Study, IEEE Symp. Software Reliability engineering, November 2010.

[17] Atefeh Tajpour, Maslin Masrom, Suhaimi Ibrahim, Mohammad Sharifi: SQL injection detection and prevention Tools Assessments, IEEE 2010.

[18] Ntagwabira Lambert, Kang Song Lin: Use of Query Tokenization to detect and prevent SQL Injection attacks, IEEE 2010.

[19] Michelle Ruse, Tanmoy Sarkar, Samik Basu: Analysis and Detection of SQL Injection Vulnerabilities via Automatic Test Case Generation of Programs, Annual International Symposium on application and the Internet. 2010.

[20] A. Roichman E. Gudes: DIWeDa –Detecting Intrusions in Web Databases, Vol. 5094, pp. 313-329 Springer Heidelberg 2008.

[21] J. Fonseca and Marco Vieira: Mapping software fault with web security vulnerabilities, IEEE conference on dependable system and network, June 2008.

[22] P. Grazie: SQL Prevent Thesis, University of Columbia, Vancouver, Canada 2008.

[23] J. Fonseca and Marco Vieira and Henrique Madeira: Training Security Assurance Team using Vulnerability Injection, IEEE Pacific Rim Dependable Computing, December 2008.
[24] T. Pietraszek, C. V. Bergh: Defending against Injection Attacks Trough Context-Sensitive String Evaluation, Recent Advanced in Intrusion Detection Volume: 3858, 2006.
[25] Z. Su and G. Wassermann The Essence of Command Injection Attacks in Web Applications The 33rd Annual Symposium on Principles of Programming Languages, 2006.
[26] McClure and I. H. Kruger: SQL DOM: Compile time checking and dynamic SQL statements, Software Engineering ICSE 2005.
[27] M. Martin, B. Livshits, and M. S. Lam. Finding Application Errors and Security Flaws Using PQL: A Program Query Language. *ACM* SIGPLAN Notices, Volume: 40, Issue: 10, pp: 365-383, 2005.
[28] V. B. Livshits and M. S. Lam: Finding Security Errors in Java Programs with Static Analysis. In Proceedings of the 14th Usenix Security Symposium, pages 271–286, Aug. 2005.
[29] V. Haldar, D. Chandra, and M. Franz: Dynamic Taint Propagation for Java in Proceedings 21st Annual Computer Security Applications Conference, Dec. 2005.
[30] G. T. Buehrer, B. W. Weide, and P. A. G. Sivilotti: Using Parse Tree Validation to Prevent SQL Injection Attacks International Workshop on Software Engineering and Middleware (SEM), 2005.
[31] A. Nguyen-Tuong, S. Guarnieri, D. Greene, J. Shirley, and D. Evans: Automatically Hardening Web Applications Using Precise Tainting Information. In Twentieth IFIP International Information Security Conference May 2005.
[32] S. W. Boyd and A. D. Keromytis "SQL Rand Preventing" Cryptography and network security conference pages 292-302 June 2004.
[33] G. Wassermann and Z. Su: An Analysis Framework for Security in Web Applications in Proceedings of the FSE Workshop on Specification and Verification of Component-Based Systems, pages 70–78, 2004.
[34] C. Gould, Z. Su, and P. Devanbu JDBC Checker: A Static Analysis Tool for SQL/JDBC Applications. In Proceedings of the 26th International Conference on Software Engineering pages 697–698, 2004.
[35] Yao-Wen Huang, Fang Yu, Christian Hang, Chuang Hang, Tsai, D.T. Lee, Sy-Yen Kuo "Securing Web Application Code by Static Analysis and Runtime Protection" 13th conference on World wide web in ACM New York USA 2004.
[36] Y. Huang S. Huang T. Lin and C. Tsai: Web Application security Assessment by Fault Injection and Behaviour, In Proceeding of the 11th International World Wide Web Conference, May 2003.
[37] D. Scott and R. Sharp: Abstracting Application-level Web Security. In Proceedings of the 11th International Conference on the World Wide Web, pages 396–407, 2002.

Authors' Profiles

Harish Dehariya: Received his Bachelor's degree in computer Science and Engineering from BIST, Bhopal, India in 2012. At present he is pursuing his Master of Engineering in Computer Science and Engineering from UIT, RGPV, Bhopal, Madhya Pradesh, India.

Dr. Piyush Kumar Shukla: received his Ph.D. in Computer Science & Engineering in 2013 from RGPV, Bhopal. M.P. India, M. Tech in Computer Science & Engineering in 2005 from SATI, Vidisha, Madhya Pradesh Bachelor's degree in Electronics & Communication Engineering, LNCT, Bhopal Madhya Pradesh in 2001. He is a member of IEEE and IACSIT. Currently he is working as an Assistant Professor in Department of Computer Science & Engineering, UIT-RGPV Bhopal, Madhya Pradesh, India. He has published more than 40 Research Papers in various International & National Journals & Conferences.

Prof. Manish Ahirwar: received his Master degree in M. Tech in 2006 in computer Science and Engineering from Atal Bihari Vajpeyi IIITM Gwalior, Madhya Pradesh India and Bachelor of Engineering in Computer Science and Engineering from M.I.T.S. Gwalior, Madhya Pradesh in 2004. Currently he is working as an Assistant Professor in Department of Computer Science & Engineering, UIT-RGPV Bhopal Madhya Pradesh India. He is a member of IEEE Cloud Computing, IACSIT and IAENG. He has published more than 10 Research Papers in various International & National Journals & Conferences.

How to cite this paper: Harish Dehariya, Piyush Kumar Shukla, Manish Ahirwar,"A Survey on Detection and Prevention Techniques for SQL Injection Attacks", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.6, No.6, pp.72-79, 2016.DOI: 10.5815/ijwmt.2016.06.08