ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

A Top Web Security Vulnerability: SQL Injection attack – Survey

[1] .Thirunavukkarasan ,[2]P.Maniandan,[3]S. P.Ramesh
[1][2][3] Assistant Professor
[1]Galgotias University, [2] MIET Engg College, [3]Dhanalakshmi Engg College

Abstract: Online services serve today's basic needs of people in this modern internet era. This internet based services make use of huge volumes of data for their processing. Social media is another interaction media through which huge volumes of data is stored in a daily basis. These data has to be stored in databases. SQL queries provide means for accessing, modifying and retrieval of data from databases. If attacker finds way in modifying queries with unauthorized access, the evidence of data for future purposes is under danger. Such SQL injection attack is found to be predominant in web applications. Such an attack has to be mitigated and addressed soon for protecting the huge data. The survey reveals the importance of data and effects of SQL injection attack over data and its prevention techniques.

Indexed Terms: SQL injection attack, Web Applications.

INTRODUCTION

The birth of Internet has led to many advantages in all fields. This marked the beginning of new era. Now-a-days mostly all transactions are made through online. Web Applications are an integral part of each and every one's daily activities. Each service provides authentication of the user through username and password. If the secrecy of the username and password are revealed out then the purpose of doing online transaction will be of no use. Then the attacker uses this chance to compromise the website by the username and password. Web services are vulnerable to unauthorized users, so that they can gain access and use the information of authorized users and try to modify it. This is found to be dangerous when the transaction is all about money transfer. This type of attack is called SQL injection attack. Such attack which exploits the databases connected with the applications has to be detected and prevented. Tomorrow's world revolves around the data being stored from the history. The data stored in volumes under databases has to be protected for any kind of references in future.It is mandatory to protect our stored data from SQL injection attacks. People rely on communication through social media for all their interactions. This increases the storage of data in electronic media eventually in data warehouses. The huge data in databases is vulnerable to SQL injection attack which has the ability to completely destroy the data. It is necessary to analyze the possible solutions, detection and prevention techniques. The survey describes about the SQL injection attack, its types, possible effects of SQL injection on databases, the existing solutions for detection and prevention techniques and its analysis for the better solution to protect data from SQL injection attack.

MOTIVATION

Today's modern world could not move forward without the use of internet. This is due to the rapid rise of internet users every second. The web services are becoming the most popular technology advancement among the people. All the details of each and every person are stored somewhere and are available through the internet. When the internet is used to transfer data then the critical information which should be kept secret are made available in the internet. This makes them vulnerable and the attackers can steal the unauthorized information in the internet by means of the security loops available in the websites. There are lot of possible attacks followed by the hackers. One such attack which is least aware and very dangerous among all other web applications security attacks is SQL injection attack.Open Web application Security project (OWASP) is the voluntary organization which tries to provide awareness to the public related to vulnerabilities, security loop holes, attacks, its worse effects on web applications in the top 10 list. It makes it releases for every three years from 2004. Table.1 details the survey of top 10 attacks according to their releases made in 2004, 2007, 2010 and and 2013. It is obvious from the table that SQL injection attack tops the list for the past two releases to till date.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

Top Attacks	2004	2007	2010	2013
A1	Unvalidated input	Crosssite scrlpting xss)	Injection	Injection
A2	Broken access control	Injection flaws	Cross site scrlpting (xss)	Broken authentication and session management
A3	Broken authentication and session management	Malicious file execution	Broken authentication and session management	Crosssite scrlpting xss)
A4	Crosssite scrlpting	Insecure direct object reference	Insecure direct object reference	Insecure direct object reference
A5	Buffer overflow	Cross site request forgery (csrf)	Cross site request forgery (csrf)	Security misconfiguration
A6	Injection flaws	Information leakage and improper error handling	Security misconfiguration (new)	Sensitive data exposure
A7	Improper error handling	Broken authentication and session managem	Failure to restrict url access	Missing function level access control
		ent		
A8	Insecure storage	Insecure cryptographic storage	Unvalidated redirects and forwards (new)	Cross-site request forgery (csrf)
A9	Application denial of service	Insecure communication	Insecure cryptographic storage	Using known vulnerable components
A10	Insecure configuration management	Failure to restrict url access	Insufficient transport layer protection	Unvalidated redirects and forwards.

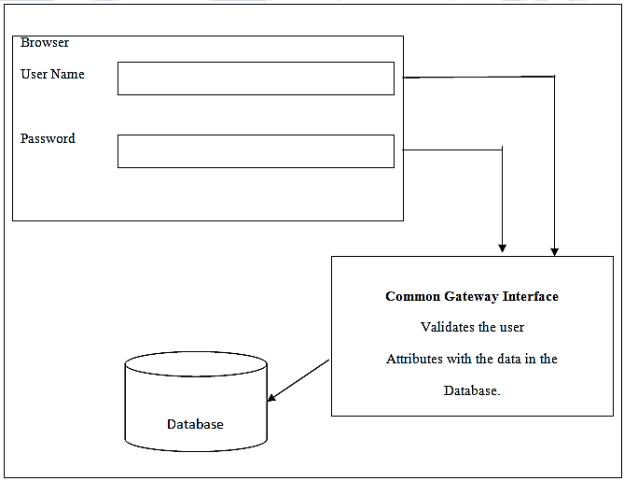

Fig.1 Web application architecture with respect to SQL injection attack

SQL INJECTION ATTACK:

SQL Injection attack is a code injection technique which exploits the vulnerability present in the application code for gaining the unauthorized access over the data. The effects of SQL injection attacks are modification of data, destroying some fields of data, unauthorized access to data, stealing of data, dropping down the entire database, etc.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

Types of SQL injection Attack:

- Tautology queries
- Illegal queries
- Union queries
- Piggy bagged queries
- Stored Procedures

Tautology queries:
These queries include tautology statements into the queries to gain the Boolean value as always true to get unauthorized access over the database.

Illegal Queries:
This kind of attack introduces some logically incorrect words into the queries to yield the information about structure of the database.

Union Queries:
This kind of attack attaches a sub query with the existing query using a keyword union in order to perform some malicious activity over the database.

Piggy bagged queries:
This kind of attack tries to insert malicious query along with the existing query by adding ';' at the end of the query through which malicious effects are created.

Stored Procedures:
Queries packed inside the stored procedures are also found vulnerable to piggy bagged queries through which unwanted queries are also allowed to be executed with the existing one.

RELATED WORK:

Some attacks try to obtain the private and sensitive data by leveraging the vulnerabilities in the web application. An overview of DOS, SQL injection,XSS attacks[2] are provided. The biggest challenge is the internet's open architecture based on mutual trust. This enables attackers to operate under numerous untraceable aliases. The other is that most techniques that increase the security also degrades the performance. In order to protect mission-critical websites, secure delivery network has been built as a shield to absorb attack traffic. Mechanisms are provided to protect DNS. The defending mechanisms at the edge by implementing a distributed web application firewall is given. Various rules are given to apply to every request.

Various methods [15] are available to overcome XSS. But the issue still occurs in anyapplications as the methods are difficult to adopt and implement. Moreover the complexity of XSS problem has further added to it. In this paper a code auditing approach is provided which comprises of two phases. First phase is the extraction of all features that could defense against XSS attacks. The second phase starts by verifying the information in first phase for the adequacy of defense methods in preventing potential cross site script attacks. With all these a tainted information flow graph is modeled. This is implemented using XSS defense extractor using seven test subjects. But the DOM based XSS attack is not considered in this model and only few applications are examined.

Several major websites have been the target of XSS attacks [17] for decades. In order to protect these, wide spectrum of solutions like simple static analysis to complex runtime protection mechanisms have been provided. The normal idea is to use special characters to make the web browser to switch from a data context to code context. The common defenses are defensive coding practices, XSS testing, vulnerability detection, and runtime attack prevention. Defensive coding practices can completely remove all XSS vulnerabilities in Web applica¬tions if they are applied appropriately. But they are labor-intensive, prone to human error, and difficult to enforce in deployed applications. Input Validation methods like Specification-based IVT, Code-based IVT exists. But generation of mutants is not automated and hence labor intensive. Vulnerability detection includes static analysis, static string analysis, combined static and dynamic analysis. To prevent the real time attacks using IDS or runtime monitors are used. Many approaches are used for server side prevention, client side prevention. In spite of all these weakness occurs. Hence form the development perspective and program verification perspective many developments need to be made.

Improperly coded applications is a great advantage for injection attacks to insert and execute attacker-specified commands which in turn enables to access the critical data and resources. To avoid and reduce the security vulnerabilities 16] a defense-in-depth approach is provided. According to this approach, security depends on several layers of mechanisms to cover the failure of each other. To ensure security in all phases of the software product's development life cycle various techniques and tools are needed. The three important phases, out of the phases in the life cycle, to be focussed are implementation, testing, deployment. The defense-in-depth approach assumes that each security precaution can

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

fail. Three unique lines of defense required are input validation, hotspot protection, and output validation. It is also important to find the dangerous hidden flaws in the code . The main approaches for detecting vulner-abilities are white-box analysis and black-box testing. Further this paper provides an overview of the intrinsic limitations penetration testing and static code analysis. Attack detection mechanisms like intrusion detection systems (IDSs) or Web application firewalls (WAFs) are provided. Their limitations are also highlighted.

There are many existing methods for the detection and prevention of SQL injection attacks. In this paper, JDBC checker[37] is capable of analyzing statically the correctness of queries being generated dynamically. Java can itself detect the type errors generated statically.

In this paper, static analysis algorithm based on lattice is created for checking type errors and vulnerable codes are protected using guards at run time. WebSSARI[34] (Web application Security by static Analysis and Runtime Inspection) is developed to test the algorithm which verified the correctness of websites from SourceForge.net. There are several security issues in an application developed using Java.

A new lightweight static analyzer[28] which is capable of detecting security errors like possibility of Bad Session stores and SQL Injections has been developed and verified with many applications.

One approach of converting vulnerable SQL statements[23] into prepared statements can protect the effect of input during execution. Automated fix generation algorithm is able to remove SQL Injection vulnerabilities (SQLIV's) by performing SQLChecks over the web applications.

Abstract model[36] of source code with all possible inputs constructs dynamic SQL queries. This query set is being utilized for generating a finite state automata. This FSA is capable of verifying the presence of security violations in the source code.

At the time of development and debugging phases, Sania(Syntactic and Semantic Analysis for Automated Testing against SQL Injection) detects SQLIV by generating parse trees for both intended and attacked SQL Query. Sania compares the parse trees to find the spots being vulnerable to the SQLIA.

In this paper, the comparison between parse trees generated using SQL queries with user input and without user input reveals SQLIV present in the Web application. The author had written a specific grammar called BISON used for SQLIA.

AMNESIA[29] Analysing and Monitoring for Neutralizing SQL Injection Attack uses a model built statically comprising of the legitimate queries and checks it with the dynamic queries generated after the user's input in order to analyse the presence of SQLIV in the application code. The formal definition of all possible command injection attacks is explained in detail in this paper.

The author has developed SQLCHECK[25] for preventing all kinds of command injection attacks which are capable of changing the context of the intentions of those commands in the Web application.

A method is developed to have concern on the security[26] over the stored procedures in the database layer. A parser for stored procedure is developed for code analysis and validation is done during runtime to eliminate such attacks.

A learning based[27] anomaly approach is being developed by learning the profiles of access to the database by normal users which can detect the abnormal behaviours by notifying the change in access to the database.

A Database parser[35] SQLrand capable of translating random SQL queries to standard language , is used for creating unpredictive SQL commands which are not understandable by the attackers. This paper have given a detailed survey over all possible kinds of SQLIA and existing detection and prevention techniques for SQLIAs and the evaluation of those techniques are also discussed.
Context- Sensitive String Evaluation (CSSE)[24] performs checks on the area provided to the user and area of the developer in the code to analyse the context information present in the code. CSSE is platform specific and does not modify the available application code.

A taint[30] propagation scheme is utilized in validating the user input being submitted at the runtime. The scheme mainly focuses on the source method from which user strings are manipulated, Propagation which are the dynamically generated queries after the submission of user inputs, and sink method which executes the dynamic queries.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

A Remote[32] execution of the SQL query by the database server is done by creating objects namely Safe query objects which are capable of translating classes into executable queries.

Programmer Query language is analyzed statically to find the matches corresponding to the context, flow and alias analysis. The dynamic analysis finds out the violations and performs actions as specified by the user query.

MEASUREMENTS AND EVALUATION:

SQL Injection attack has to be detected and prevented for protecting the databases from it's worse effects. The possible solutions against SQL Injection attack are evaluated using the detection rate.
Detection rate is defined to the number of SQL Injection attacks detected to number of SQL Injection attacks performed.
This determines the performance of the detection technique against SQL Injection attack.

RESEARCH DIRECTIONS:

After analyzing the existing solutions to prevent and detect SQL injection attacks, it is understood that no existing solutions is primarily utilized in real time for detecting the SQL Injection attack. To make people aware about such a dangerous attack, the detailed survey is being written. With these solutions, a proper hybrid variety has to described for complete protection of web applications from SQL Injection attack which could be implemented in real time situations on the internet.

CONCLUSION:

Thus, SQL injection attack is still found unattended. Web services suffer a lot of security issues due to SQL injection attack. This attack almost finds its way in destroying the entire database behind the targeted web application. So, the possible solutions in the survey have to be implemented in real time to protect web applications from severe damage.

REFERENCES

[1] OWASP, (Open Web Application Security Project) Top 10 Security threats, http://www.owasp.org/.

[2] David Gillman, Yin Lin,Bruce Maggs,Ramesh K. Sitaraman," Protecting Websites from Attack with Secure Delivery Networks",IEEEComputer Society , Vol. 48 , Issue 4 , April 2015.

[3] Mahima Srivastava, "Algorithm to Prevent Back End Database against SQL Injection Attacks ", IEEE International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 2014.

[4] Seokmo Kim, Young B. Park, "An object pool realization of Whitelist strategies to neutralize Injection flaws ",IEEE International Conference on IT Convergence and Security (ICITCS), Beijing,October 2014.

[5] Jos_e Fonseca, Marco Vieira, and Henrique Madeira, "Evaluation of Web Security Mechanisms Using Vulnerability & Attack Injection ",IEEE Transactions On Dependable And Secure Computing, Vol. 11, No. 5, September/October 2014 .

[7] Hussein Alnabulsi, MdRafiqul Islam, QuaziMamun, "Detecting SQL Injection Attacks Using SNORT IDS ",Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE), Nadi , November 2014.

[9] AbdelhamidMakiou, YoucefBegriche ,Ahmed Serhrouchni , "Improving Web Application Firewalls to Detect Advanced SQL Injection Attacks" , 10th International Conference on Information Assurance and Security (IAS),November 2014.

[10] Lwin Khin Shar and Hee Beng Kuan Tan, "Defeating SQL Injection",IEEE Computer Society,Vol. 46 , Issue 3, March 2013.

[11] Yi Xie, S. Tang, Y. Xiang and J. Hu, "Resisting Web Proxy-Based HTTP Attacks by
Temporal and Spatial Locality Behavior", IEEE Transactions on parallel and distributed systems,July 2013.

[12] Cristian I. Pinzon, Juan F. De Paz, Alvaro Herrero , Emilio Corchado, Javier Bajo,Juan M. Corchado," idMAS-SQL: Intrusion Detection based on MAS to Detect and Block SQL injection through data mining", Information Sciences, May 2013.

[13] Chinmay C. Kulkarni, Dr S.A. Kulkarni , "Human Agent Knowledge Transfer Applied To Web Security", Fourth IEEE International Conference on Computing, Communications and Networking Technologies (ICCCNT),2013

[14] Inyong Lee, Soonki Jeong, Sangsoo Yeo, Jongsub Moon, "A novel method for SQL injection attack detection based on removing SQL query attribute values", Mathematical and Computer Modelling 55 (2012) 58-68.

[15] L.K. Shar H.B.K. Tan, "Auditing the XSS defence features implemented in web application programs ", IET Software, Vol. 6, Issue 4, 2012.

[16] Nuno Antunes and Marco Vieira, "Defending against Web Application Vulnerabilities", IEEE Computer Society, Vol. 45, Issue 2, Feb. 2012.

[17] Lwin Khin Shar and Hee Beng Kuan Tan, "Defending against Cross-Site Scripting Attacks", IEEE Computer Society, Vol.45, Issue 3,March 2012.

[18] A. Sravanthi, K. Jayasree Devi, K. Sudha Reddy, A. Indira, V. Satish Kumar, "Detecting Sql Injections From Web Applications",International Journal Of Engineering Science & Advanced Technology [IJESAT] ,Volume-2, Issue-3,May-Jun 2012 .

[19] Franc¸ois Gauthier, Ettore Merlo , "Fast Detection of Access Control Vulnerabilities in PHP Applications" ,19th IEEE Working Conference on Reverse Engineering, October, 2012.

[20] DrR.P.Mahapatra and MrsSubiKhan , "A Survey Of Sql Injection Countermeasures" , International Journal of Computer Science & Engineering Survey (IJCSES) Vol.3, No.3, June 2012.

[21] Kai-Xiang Zhang, Chia-Jun Lin , Shih-Jen Chen , Yanling Hwang , Hao-Lun Huang, Fu-Hau Hsu , "TransSQL: A Translation and Validation-based Solution for SQL-Injection Attacks", First International Conference on Robot, Vision and Signal Processing(RVSP) , November 2011 .

[22] Ahmed Salem , "Intercepting Filter Approach to Injection Flaws", Journal of Information Processing Systems, Vol.6, No.4, December 2010.

[23] Prof (Dr.) SushilaMadan ,Ms. Supriya Madan , "Security Standards Perspective to Fortify Web Database Applications From Code Injection Attacks" , IEEE International Conference on Intelligent Systems, Modelling and Simulation (ISMS) , January 2010 .

[23] S. Thomas, L. Williams, Using automated fix generation of secure SQL statements,IEEE Computer Society, 2007.

[24] T.C. Pietraszek, V. Berghe, Defending against injection attacks through context– sensitive string evaluation, LNCS, 2006.

[25] Z. Su, G. Wassermann, The essence of command injection attacks in web applications, ACM SIGPLAN-SIGACT on PrincIPles of Programming Languages, 2006.

[26] K. Wei, M. Muthuprasanna, S. Kothari, Preventing SQL injection attacks in stored procedures,Software Engineering Conference 2006.

[27] F. Valeur, D. Mutz, G. Vigna, A learning-based approach to the detection of SQL attacks,Conference on Detection of Intrusionsand Malware and Vulnerability Assessment, 2005.

[28] V.B. Livshits, M.S. Lam, Finding security errors in Java programs with static analysis,Usenix Security Symposium, 2005.

[29] W.G. Halfond, A. Orso, AMNESIA: analysis and monitoring for neutralizing SQL-injection attacks, IEEE/ACM International Conference on Automated Software Engineering, 2005.

[30] V. Haldar, D. Chandra, Franz, Dynamic Taint propagation for Java Computer Security Applications Conference, 2005.

[31] G. Buehrer, B.W. Weide, P.A.G. Sivilotti, Using parse tree validation to prevent SQL injection attacks, ACM, 2005.

[32] W.R. Cook, S. Rai, Safe query objects: statically typed objects as remotely executable queries Conference on Software Engineering, 2005.

[33] R. McClure, I. Krüger, SQLDOM: compile time checking of dynamic SQL statements,Conference on Software Engineering, 2005.

[34] Y. Huang, F. Yu, C. Hang, C.H. Tsai, D.T. Lee, S.Y. Kuo, "Securing web application code by static analysis and runtime protection", ACM conference, 2004.

[35] S. Boyd, A. Keromytis, SQLrand: preventing SQL injection attacks, in: Applied Cryptography and Network Security, LNCS,2004.

[36] G. Wassermann, Z. Su, An analysis framework for security in web applications, FSE Workshop SAVCBS, 2004.

[37] C. Gould, Z. Su, P. Devanbu, JDBC checker: a static analysis tool for SQL/JDBC applications, on Software Engineering, ICSE, 2004.