International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 03 | March - 2022 ISSN: 2582-3930

SQL INJECTION DETECTION & PREVENTION

SAKSHI SAKPAL ,

MODEL COLLEGE AUTONOMOUS REGD. DOMBIVLI

Abstract

- In recent times hacking attacks is a very common and huge issue all over the world. Many large MNC's like Google, Apple, Microsoft etc. are investing millions of dollars to stop these types of Hacking and penetration attacks. Database driven web application are threaten by SQL Injection Attacks (SQLIAs) because this type of attack can compromise confidentiality and integrity of information in databases. Only a few techniques those take the advantage of these SQL vulnerabilities are known by the researchers. Hence, the solutions available so far solve only a few SQL related injections. This paper uncovers this problem to the research community by means of a chronological review of various SQL injection attacks. We analyse the best ways to protect ourselves from these attacks. We have also analysed and presented the various existing detection and prevention techniques against the SQL injection attacks.

Keyword: SQL Injection Attacks, detection, prevention, evaluation, technique

Introduction

An SQL injection attack typically means inoculation of a malicious SQL query as input by an attacker to the application. Then an effective SQL injection can lead to vulnerable attack and makes the attacker to read and modify the sensitive data in the database. Actually, an attacker intrudes to the web application database and consequently, access to data. For stopping this type of attack different approaches have been proposed by researchers but they are not enough because usually they have limitations. It also executes the administrative operations on the database. In some cases, attackers inject different commands to operating system to gain access. SQL attacks are type of the injection attacks where the malicious SQL commands are inserted into the input data to result the implementation of the predefined SQL commands. For preventing the SQLIAs defensive coding has offered as a solution but it is very difficult. Not only developers try to put some controls in their source code but also attackers continue to bring some new ways to bypass these controls. Hence it is difficult to keep developers up

to date, according the last and the best defensive coding practices. On the other hand, implementing of best practice of defensive coding is very difficult and need to special skills and also erring. These problems motivate the need for a solution to the SQL injection problem.

Prevention & Detection

In this section we present the most popular techniques, which are used to either detect or prevent classical types of SQLIA while they don't overcome the modern types of SQLIA as follows:

1. AMNESIA, that stands for Analysis and Monitoring for Neutralizing SQLIA. AMNESIA combines dynamic and static analysis for detecting and preventing web application vulnerabilities at the runtime. AMNESIA uses static analysis to generate different type of query statements. In the dynamic phase, AMNESIA interprets all queries before they are sent to the database and validates each query against the statically built models. to detect and prevent SQLIA at the runtime and it was developed based on SQL syntax-aware at the web application layer, and negative taint at the database layer.

2. SAFELI, which capable of discovering very delicate vulnerabilities by taking advantage of source code information. SAFELI framework aims at identifying the SQL Injection attacks during the compile-time. The drawback of this tool that the implementation does not complete, and dose not overcome the tautologies attack.

3. The authors developed a WASP (Web Application SQL Injection Protector) tool which is efficient and effective in stopping more than 12,000 attacks without generating any false positives in non-real-time environment. The limitation of this tool can be founded by implemented the approach for the binary applications and deployed web applications.

4. R-WASP (Real Time-Web Application SQL Injection Detector and Preventer) tool, which capable to stop all attacks effectively and detects SQLIAs in real-time environment. The limitation of this tool is required more practices in order to mitigate stored procedures attacks efficiently.

5. Real Time Web Application SQL Injection Protector (RT-WASP) tool to detect SQL injection attacks in stored procedures, besides detecting all classical types of SQLIA RTWASP technique to encompass SQLI and XSS attack in the web applications .

6. A technique that is based on the principle of dynamic query structure validation which is done through checking query's semantics. This major purpose of this technique focuses on this particular kind of attack, stored procedure attack, along with general prevention.

7. A prevention tool for java which is called SecuriFly. This tool is used to chase string instead of character for taint information and try to sanitize query strings that have been generated using tainted input. The limitation of this tool are listed as follows

- This approach cannot use to stop an injection result from inserting SQL command in numeric fields.
- the main limitation of this approach is the difficulty of identifying all the sources of user input.
- Besides, this technique stops all classical types of SQLIA partially because of limitations of the underlying approach .

8. JDBC-Checker, which is technique used for statically checking the correctness type of dynamically generated SQL queries. JDBC-Checker is able to detect major causes of SQLIA vulnerabilities in code, which is checking on the improper typing of the query. The drawbacks of this technique are listed as follows:

- this technique could not catch more general types of SQLIA because in the most of these attacks, the attacker writes queries correctly.
- As mention the JDBC-Checker stop all classical types of SQLIA partially.

9. Dynamic Candidate Evaluations method for automatic prevention of SQLIA, which is called CANDID. This technique dynamically extracts the query structures from each SQL query location which are intended by the developer. Hence, CANDID solves the issue of manually modifying the application to create the prepared statements. The drawback of this technique is partially stop SQLIAs because of the constraints on the basic approach.

10. Swaddler analyses the internal state of a web application. It works based on both single and multiple variables and shows an impressive way against complex attacks to web applications. First the approach describes the normal values for the application's state variables in critical points of the application's components. Then, during the detection phase, it monitors the application's execution to identify abnormal states.

11. DIWeDa which is a prototype that acts at the session level rather than the SQL statement or transaction stage, to detect the intrusions in Web applications. The proposed framework is efficient and could identify SQL injections and business logic violations too. the drawback of this technique that it does not have the ability to detect all (classical and Modern) types of SQLIA except its ability to detect Inference attack.

12. Positive tainting not only focuses on positive tainting rather than negative tainting but also it is automated and does need developer intervention. Moreover this approach benefits from syntax-aware evaluation, which gives developers a mechanism to regulate the usage of string data based not only on its source, but also on its syntactical role in a query string.

13. SQL Prevent is consists of an HTTP request interceptor. The original data flow is modified when SQL Prevent is deployed into a web server. The HTTP requests are saved into the current thread-local storage. Then, SQL interceptor intercepts the SQL statements that are

made by web application and pass them to the SQLIA detector module. Consequently, HTTP request from thread local storage is fetched and examined to determine whether it contains an SQLIA. The malicious SQL statement would be prevented to be sent to database, if it is suspicious to SQLIA

14. Use of Automated approaches in order to prevent SQLI input manipulation flaws. This approach is based on defensive programming and code review, where in defensive programming an input filter is implemented to prevent user from entering malicious keywords or characters, this is achieved by using white lists or black lists. The major advantage of this technique is a low cost mechanism in detecting bugs, while the major drawbacks of this technique are listed as follows:

- It is required deep knowledge on SQLIAs.
- It is unable to detect the stored procedure attack, Alternate Encodings and (SQL Injection and XSS) attack.

15. SQLIPA (SQL Injection Protector for Authentication), that adopts hash value approach to further improve the user authentication mechanism. There is a hash value for username and other for password. Whenever user account is created, the hash values, that represent username and password, are created and calculated at runtime. The drawback of this technique is the SQLIPA has the ability to detect the only Tautologies attack and cannot detect other classical and modern types of SQLIA

Comparison of SQL Injection Detection/Prevention techniques Based on Deployment Requirement

Each technique with respect to the following criteria was evaluated: (1) Does the technique require developers to modify their code base? (2) What is the degree of automation of the detection aspect of the technique? (3) What is the degree of automation of the prevention aspect of the technique? (4) What infrastructure (not including the technique itself) is needed to successfully use the technique?

	Tautologies	Logically Incorrect Queries	Union Query	Piggy Query	Stored Procedures	Inference	Alternate Encoding	Modern types of SQLIAs
AMNESIA	✓	✓	✓	✓	×	✓	✓	×
SAFELI	×	✓	✓	✓	✓	✓	✓	×
WASP	✓	✓	✓	✓	✓	✓	✓	×
RWASP	✓	✓	✓	✓	×	✓	✓	×
RT-WASP	✓	✓	✓	✓	✓	✓	✓	×
SecuriFly	+	+	+	+	+	+	+	×
JDBC Checker	+	+	+	+	+	+	+	×
CANDID	+	+	+	+	+	+	+	×
Swaddler	+	+	+	+	+	+	+	×
DIWeDA	×	×	×	×	×	✓	×	×
Positive Tainting	✓	✓	✓	✓	✓	✓	✓	×
SQL Prevent	✓	✓	✓	✓	×	✓	×	×
Automated approaches	✓	×	×	×	×	×	×	×
SQLIPA	✓	×	✓	×	×	×	×	×

CONCLUSION

It is obvious from above discussion and analysis that SQLIAs are one of the most threat to the applications, whether they are web, mobile or desktop applications, that are connected to the database. In this paper we have surveyed the most popular existing attack issues, which is SQLIA. Also we have presented a survey report on classical and modern types of SQLIA, their working methods, and detection and prevention techniques against classical and modern types of that attack. For evaluation, we compare the detection and prevention techniques in terms of their ability to detect the attack, or prevent the attack or partially stop the attack. Regarding the results, the efficiency of some techniques should be improved to overcome the SQLIA

REFERENCES

] W. G. J. Halfond and A. Orso, "Preventing SQL injection attacks using AMNESIA," presented at the Proceedings of the 28th international conference on Software engineering (ICSE), ACM, Shanghai, China, Pages: 795-798, May 20–28, 2006, 2006. X.Fu, X. Lu, B. Peltsverger, S. Chen, G. Southwestern, K. Qian, and S. Polytechnic, "A Static Analysis Framework For Detecting SQL Injection Vulnerabilities" 31st Annual International Computer Software and Applications Conference(COMPSAC 2007), IEEE, ISSN: 0730-3157, pages Number 1–8. , China ,2007. W. G. J. Halfond, A. Orso, and I. C. Society, "WASP: Protecting Web Applications Using Positive Tainting and SyntaxAware Evaluation", : IEEE Transactions on Software Engineering, volume. 34, Issue 1, pages. 65–81, 2008. M. H. A. S. P. Medhane, "R-WASP: Real Time-Web Application SQL Injection Detector and Preventer", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-5, pages. 327–330April 2013. Zainab S. Alwan et al, International Journal of Computer Science and Mobile Computing, Vol.6 Issue.8, August- 2017, pg. 5-17 © 2017, IJCSMC All Rights Reserved 17 S. Manmadhan , Manesh T. , "A METHOD OF DETECTING SQL INJECTION ATTACK TO SECURE WEB APPLICATIONS", International Journal of Distributed and Parallel Systems (IJDPS) ,Volume.3, November 2012. 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 Pages: 365-383, 2005. 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 (ICSE04) Formal Demos, ACM, ISBN: 0-7695-2163-0, pages 697– 698, 2004. P. Bisht, P. Madhusudan, and V. N. Venkatakrishnan, "CANDID: Dynamic Candidate Evaluations for Automatic Prevention of SQL Injection Attacks", ACM Transaction on information System Security, pages.1–39, 2010. Macro Cova, Davide Balzarotti." Swaddler: An Approach for the Anomaly-based Detection of State Violations in Web Applications", In Proceedings of the International Symposium on Recent Advances in Intrusion Detection (RAID), pages: 63–86, (2007) A. Roichman, E. Gudes, "DIWeDa - Detecting Intrusions in Web Databases , volume. 5094, pages. 313–329, Heidelberg (2008). William G. Halfond, Alessandro Orso, "Using Positive Tainting and Syntax Aware Evaluation to Counter SQL Injection Attacks", 14th ACM SIGSOFT international symposium on Foundations of software engineering, ACM. pages: 175 – 185, 2006. P.Grazie., PhD, "SQL Prevent thesis", University of British Columbia (UBC) Vancouver, Canada, 2008. Mei Junjin, "An Approach for SQL Injection Vulnerability Detection" Proceedings of the 2009 Sixth International Conference on Information Technology: New Generations, IEEE computer society, Las Vegas, Pages 1411-1414, April. 2009. S. Ali, SK. Shahzad and H. Javed, "SQLIPA: An Authentication Mechanism against SQL Injection", European Journal of Scientific Research, Volume.38, Number.4, pages: 604-611, 2009.