Volume 5, Issue 12, December 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Analysis of SQL Injection Attacks and Prevention Methods in Web Applications

Dr. Amit Chaturvedi[*]
MCA Deptt, Govt. Engineering College,
Ajmer, India

Aijaz Ahmad Rather
M.Phil Scholar
Ajmer, India

Abstract— Database hacking is a major problem for organizations financially as well professionally. The recent invention of SQL injection attacks became known to online world, which is a big problem for the companies, where data is crucial. SQL injection can provide an attacker with unauthorized access to sensitive data including, customer data, personally identifiable information (PII), trade secrets, intellectual property and other sensitive information. An attacker can use SQL injection to bypass authentication or even impersonate specific users. The latest encryption algorithms like Advanced Encryption Standard, Secure Hashing techniques, etc should be applied for validating the user with examining its pre-recorded nature and behaviour. So, the intensions of malicious users may be pre-measured. In this paper, we proposed a model that will help to understand the scenario of such attacks and the phase where more research is required.

Keywords— SQL Injection, attacks, vulnerability, SQLi, database, hashing, AES

I. INTRODUCTION OF SQL INJECTION

SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements (also commonly referred to as a malicious *payload*) that control a web application's database server (also commonly referred to as a *Relational Database Management System – RDBMS*). Since an SQL injection vulnerability could possibly affect any website or web application that makes use of an SQL-based database, the vulnerability is one of the oldest, most prevalent and most dangerous of web application vulnerabilities.

By leveraging SQL injection vulnerability, given the right circumstances, an attacker can use it to bypass a web application's authentication and authorization mechanism and retrieve the contents of an entire database. SQL injection can also be used to add, modify and delete records in a database, affecting data integrity.

To such an extent, SQL injection can provide an attacker with unauthorized access to sensitive data including, customer data, personally identifiable information (PII), trade secrets, intellectual property and other sensitive information.

Keeping the above in mind, when considering the following, it's easier to understand how lucrative a successful SQL injection attack can be for an attacker.

- An attacker can use SQL injection to bypass authentication or even impersonate specific users.
- One of SQL's primary functions is to select data based on a query and output the result of that query. A SQL injection vulnerability could allow the complete disclosure of data residing on a database server.
- Since web applications use SQL to alter data within a database, an attacker could use SQL injection to alter data stored in a database. Altering data affects data integrity and could cause repudiation issues, for instance, issues such as voiding transactions, altering balances and other records.
- SQL is used to delete records from a database. An attacker could use an SQL injection vulnerability to delete data from a database. Even if an appropriate backup strategy is employed, deletion of data could affect an application's availability until the database is restored.
- Some database servers are configured (intentional or otherwise) to allow arbitrary execution of operating system commands on the database server. Given the right conditions, an attacker could use SQL injection as the initial vector in an attack of an internal network that sits behind a firewall.

II. THE ANATOMY OF AN SQL INJECTION ATTACK

An SQL injection needs just two conditions to exist – a relational database that uses SQL, and a user controllable input which is directly used in an SQL query.

It shall be assumed that the attacker's goal is to exhilarate data from a database by exploiting an SQL injection vulnerability present in a web application.

Supplying a SQL statement with improper input, for example, providing a string when the SQL query is expecting an integer, or purposely inserting a syntax error in an SQL statement cause the database server to throw an error. Errors are very useful to developers during development, but if enabled on a live site, they can reveal a lot of information to an attacker. SQL errors tend to be descriptive to the point where it is possible for an attacker to obtain information about the structure of the database, and in some cases, even to enumerate an entire database just through extracting information from error messages – this technique is referred to as *error-based SQL injection*. To such an extent, database errors should be disabled on a live site, or logged to a file with restricted access instead.

Another common technique for exfiltrating data is to leverage the UNION SQL operator, allowing an attacker to combine the results of two or more SELECT statements into a single result. These forces the application to return data within the HTTP response – this technique is referred to as *union-based SQL injection*.

III. NEEDS OF RESEARCH WORK

Database thefts and attacks are common in the world of information technology. As a consequence, database extrusion prevention (DBEP) products have been rising lately. Database hacking is a major problem for organizations financially as well professionally. The recent invention of SQL injection attacks became known to online world, which is a big problem for the companies, where data is crucial. Because everywhere online world has to maintain not only small data but they have protect data warehouse by this recent threat, which is known as SQL injection attack.

Application-level vulnerabilities, which are believed to account for 70% to 90% of overall flaws, are now the main focus of attackers and researchers. Online applications (websites and services) are especially at risk due to their universal exposure and their extensive use of the firewall-friendly HTTP protocol. Moreover, database security is too often overlooked in favour of web and application server security, resulting in backend databases being a major target for attackers, which are able to use them as easy entry points to organizations' networks.

Protecting online applications(e.g. websites) and web services against SQL Injection Attacks has thus become a major concern for organizations, which face threats that can go far beyond the expected reach of the public web or application server. While several effective prevention methods have been developed, ensuring full protection against SQL Injections remains an issue on a practical level. This paper will therefore discuss the difficulties that challenge the implementation of a comprehensive SQL Injection protection solution before giving a critical overview of some of the major research proposals and main types of commercial solutions.

In our paper, we will analyses and study the issues related to the SQL Injection attacks and vulnerability. We will propose a innovative approach for the protection from such attacks.

IV. RELATED WORK

The use of web application in present life is increasing day by day. These web application are like online payment in different way i.e. payment in online shopping, reservation, electricity bills etc. As today we are facing a lot problem to our database, as SQL attacker attack and steal our secret / important information. These important data consist of lot of data. Many existing techniques / schemes, such as, defensive coding, penetration testing, information-flow analysis, and filtering, can detect and prevent a subset of the vulnerabilities that lead to SQLIAs. In this Section, we list the most relevant techniques and discuss their limitations with relation to SQLIAs. As a many techniques or schemes come to prevention these web application data.

In 2005, W.G.J and A. Orso proposed "AMNESIA: Analysis and Monitoring for NEutralizing SQL injection Attacks" brings this concept to give a idea to these problem that a automatic technique for reporting detecting and preventing SQL Injection Attacks (SQLIAs). The technique is based on the intuition that the web-application code implicitly contains a "policy" that allows for distinguishing legitimate and malicious queries. They provide a tool called as AMNESIA, a prototype based on java based application, as these tool based seven web applications, in which two of them two techniques prepared by students teams used by others researchers, and five are real applications The AMNESIA tool stop all the 1470 attack that are performed these applications.[1] In 2007, S. Bandhakavi, P. Bisht, P. Madhusudan and V.N. Venkatakrishnan proposed "CANDID: Preventing SQL Injection Attacks using Dynamic Candidate Evaluations" to gives the tool called "CANDID" that retrofits in Java written (Web applications) to defend them against SQL injection attacks. This mechanism is theoretically well founded and is based on inferring intended queries / questioning by considering the symbolic query / question computed on a program run (web application run). [2]

In 2009, S. Ali, A. Rauf, and H. Javed, proposed "SQLIPA: An Authentication Mechanism against SQL Injection," they approach a new technique / schemes known as SQLIPA to secure the authentication process of the database. SQLIPA uses password, user name, and their hash values for authentication process. SQLIPA is tested on sample data of different records in User account table. SQLIPA takes very less time overhead of 1.3 ms for authentication process (approach).[3]

In 2011, G.A. Kumar and S. Baggam gives "Enhanced Model of SQL Injection Detecting and Prevention" in which they used two methods namely: 1> signature based 2> Auditing method that can protect web applications from the attackers / attack by using SQL injection. On applying these two method can totally protects the web applications without ant hacking database and also cannot generate / produce any wrong /false transactions as a correct / right one.[4]

In 2012, D. R. Rani, B.S. Kumar, L.T. R. Rao, V.T.S. Jagadish and M.Pradeep "Web Security by Preventing SQL Injection Using Encryption in Stored Procedures" express the idea that the server has to maintain / accurate encrypted parameters of each user's password and username. The benefit of this proposed schemes / system is that the sanitize the user input and therefore stopping the user from entering the special characters /number. This schemes / system is highly protected due to use encryption algorithm.[6]

In 2013, M. Parande and B. Hajare "Build a Web Application with Precautions to Prevent SQL Injection Attack" express the idea that a simply precautions can shield web application from SQL injection to making the applications security a priority at the time of design / manufacturing .To testing each of the point should be the safeguard the software against the SQL injection. [8]

In 2013, N. Mishra and S. Gond proposed " Defences To Protect Against SQL Injection Attacks" express a new technique / schemes to protect Web applications against SQL injection Attacks. SQL Injection Attacks are a class of attacks that many of these systems are highly vulnerable to, and there is no known foolproof defense against such attacks. They design these system by using PHP and MYSQL.[9]

In 2014, **Manas Kumar1, S. Senthil kumar2 and D. Sarvanan** "SQL INJECTION MONITORING SECURITY VULNERABILITIES IN WEB APPLICATIONS" express the idea that ***a static/ rest analysis approach based on a scalable and precise point-to-point analysis.*** Readily available tools would motivate more developers to combat SQL injection. [10].

V. MODEL FOR PREVENTING WEB APPLICATION FROM SQL INJECTIONS:

Following is a model for preventing web application from sql injections:

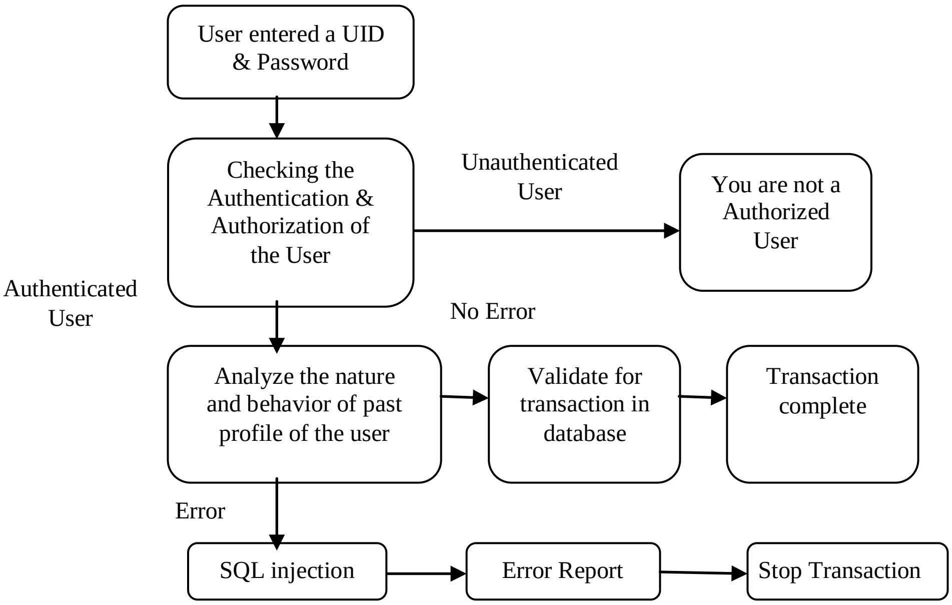
User entered a UID & Password
Checking the Authentication & Authorization of the User
Unauthenticated User
You are not a Authorized User
Authenticated User
No Error
Analyze the nature and behavior of past profile of the user
Validate for transaction in database
Transaction complete
Error
SQL injection
Error Report
Stop Transaction

Fig 1: Web application prevention model from SQL injection attack

This model explain that when a user entered with the user name and password, there should be a process for checking the authentication and set the authorization for that user, simultaneously there should a mechanism that keep track of the nature and behavior of this entered user. This will help to understand the regular authenticated user and a user with malicious intention. The analysis the nature & behaviour Process helps to judge the user and its malicious intention.

If it is found that the intended user trying to enter with malicious intentions or try to apply SQL injection attack and is not supposed to enter & access the database then it is reported as error and then the transaction get stopped forcibly. If there is no error & is a valid user then permitted to execute the transaction.

VI. CONCLUSION

After analyzing the various schemes or approaches for protecting the web application databases from SQL Injection attacks. We find that the more research is required in the authentication and authorization phase.

The latest encryption algorithms like Advanced Encryption Standard, Secure Hashing techniques, etc should be applied for validating the user with examining its pre-recorded nature and behaviour. So, the intensions of malicious users may be pre-measured.

As mentioned in figure 1, the phase checking the Authentication & Authorization of the User is an important phase for applying these solutions.

ACKNOWLEDGMENT

We feel grateful to the referees for their valuable suggestions that have helped immensely in preparing the revised manuscript.

REFERENCES

[1] Halfond, W. G. J. and A. Orso (2005). AMNESIA: analysis and monitoring for Neutralizing SQL-injection attacks. . *ASE'05.* Long Beach, California, USA.

[2] Sruthi Bandhakavi and Prithvi Bisht Preventing SQL Injection Attacks using Dynamic Candidate Evaluations, Alexandria, Virginia, USA, 2007

[3] Shaukat Ali, Azhar Rauf, and Huma Javed, 2009. "SQLIPA: An Authentication Mechanism Against SQL Injection," European Journal of Scientific Research, ISSN 1450-216X Vol.38 No.4, pp 604-611.

[4] G.Anil Kumar, Srinivas Baggam, Enhanced Model of SQL Injection Detecting and Prevention, *International Journal of Science and Advanced Technology (ISSN 2221-8386)* Volume 1 No 9 November 2011

[5] Indrani Balasundaram An Authentication Mechanism to prevent SQL Injection Attacks, *International Journal of Computer Applications Volume 19– No.1, April 2011.*

[6] Deevi Radha Rani, B.Siva Kumar, L.Taraka Rama Rao, V.T.Sai Jagadish, M.Pradeep Web Security by Preventing SQL Injection Using Encryption in Stored Procedures International Journal of Computer Science and Information Technologies, Vol. 3 (2) , 2012,3689-3692.

[7] MayankNamdev , FehreenHasan, Gaurav Shrivastav "Review of SQL Injection Attack and Proposed Method for Detection and Prevention of SQLIA"Volume 2, Issue 7, July 2012

[8] Mugdha Parande, Bhushan Hajare Build a Web Application with Precautions to Prevent SQL Injection Attack **International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-3, Issue-5, October 2013**

[9] Neha Mishra1, Sunita Gond Defenses To Protect Against SQL Injection Attacks ***International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 10, October 2013***

[10] **Manas Kumar1, S. Senthil kumar2 and D. Sarvanan** SQL INJECTION MONITORING SECURITY VULNERABILITIES IN WEB APPLICATIONS *International Journal of Information Technology* **Volume 2, Issue 3, March 2014.**

[11] R.Ezumalai, G.Aghila "Combinatorial Approach for preventing SQL Injection Attacks" in International Advance Computing Conference (IACC) IEEE 2009

[12] Xiang Fu,Xin Lu,Boris Pelts verger, Shijun chen "A static Analysis framework of Detecting SQL Injection Vulnerabilities" IEEE Transaction of computer software and application conference 2007

[13] .Kontantinos kemalis and Theodoros Tzouramanis "Specification Based approach on SQL Injection Detection" ACM 2008.

[14] Shaukat Ali, Azhar Raut "SQLIPA: An Authentication Mechanism against SQL Injection" in European Journal of Scientific Research 2009 vol-38 pg 604-611.

[15] Stephen Thomas and Laurie Williams "Using Automated Fix generation to secure sql statements". International workshop on software engineering and secure system IEEE 06.

[16] T. Scholte and W. Robertson, Preventing Input Validation Vulnerabilities in Web Applications through Automated Type Analysis, IEEE 36th International Conference on Computer Software and Applications (COMPSAC), 16 July 2012, 233-243.

[17] M. Ghafari, H. Shoja and M. Y. Amirani, Detection and Prevention of Data Manipulation from Client Side In Web Applications, IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, 2012, 1132-1136.

[18] M. Alkhalaf, T. Bultan and Jose L. Gallegos, Verifying Client-Side Input Validation Functions Using StringAnalysis, IEEE 34th International Conference on Software Engineering (ICSE), June 2012, 947-957.International Journal of Advanced Technology in Engineering and Science www.ijates.com Volume No 03, Special Issue No. 01, March 2015 ISSN (online): 2348 – 7550

[19] R. B. Brinhosa, C. M. Westphall, C. B. Westphall, D. R. dos Santos and F. Grezele, A Validation Model of Data Input for Web Services, The Twelfth International Conference on Networks, ISBN: 978-1-61208-245-5,2013.

[20] W. Min and L. Kun, An Improved Eliminating SQL Injection Attacks Based Regular Expressions Matching, IEEE International Conference on Control Engineering and Communication Technology (ICCECT), 2012, 210-212..

[21] N. A. Lambert and K. S. Lin, Use of Query Tokenization to detect and prevent SQL Injection Attacks, 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), vol.-2, July 2010, 438-440.

[22] E. Merlo, D. Letarte, G. Antoniol, Insider and Ousider Threat-Sensitive SQL Injection Vulnerability Analysis in PHP, IEEE 13th Working Conference on Reverse Engineering, 2006.