Mesopotamian journal of Cybersecurity
Vol.2023, **pp**. 30–34
DOI: https://doi.org/10.58496/MJCS/2023/006, ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/CyberSecurity

Review Article

SQL Injection Attack: Quick View

Vugar Abdullayev,[1,*], Dr. Alok Singh Chauhan [2],
[1] *Azerbaijan State Oil and Industry University, Azerbaijan*
2 Associate Professor, Department of Information Technology, ABES Engineering College, Ghaziabad, India

ARTICLE INFO

Article History

Received 10 Jan 2023
Accepted 11 Feb 2023

Keywords

Cybersecurity

SQL Injection

IDS

ABSTRACT

SQL injection is a type of security vulnerability that occurs in database-driven web applications where an attacker injects malicious code into the application to gain unauthorized access to sensitive information. This paper aims to provide a comprehensive and systematic review of the existing methods for preventing and detecting SQL injection attacks. The review covers a range of techniques, including input validation, parameterized queries, and intrusion detection systems, as well as the advantages and disadvantages of each method. The most common prevention techniques include input validation, parameterized queries, and stored procedures, while the most common detection techniques include intrusion detection systems (IDS), honeypots, and signature-based detection. The choice of method will depend on the specific requirements of the organization and the level of security required. Still, a combination of prevention and detection methods is likely to be the most effective way to secure web applications against SQL injection attacks. The paper concludes that SQL injection attacks continue to be a significant security threat to web applications, and it is essential for organizations to implement effective prevention and detection methods to secure their web applications against SQL injection attacks.

1. INTRODUCTION

Web applications that rely on databases are particularly vulnerable to SQL injection attacks. Sensitive information and data kept in the databases of these applications are at risk. An attacker compromises an application by inserting malicious code into it and using it to steal data. For nearly two decades, this kind of assault has been a major cause for security worry, leading to many data leaks and system vulnerabilities. Web applications vulnerable to SQL injection attacks are those that do not employ adequate input validation and security procedures when communicating with Structured Query Language (SQL) databases[1]. SQL (Structured Query Language) is a popular computer language for managing relational databases on the web. When there isn't enough care taken to check user input, an attacker can sneak malicious code into the program and have it run as part of a SQL query. SQL injection attacks can have devastating effects, from data theft to system compromise. Data saved in the database is vulnerable to manipulation, and the attacker can obtain access to sensitive information including login credentials and financial data. The attacker may be able to take full command of the compromised system. Since a successful SQL injection attack might have devastating effects [2], Organizations need to take measures to safeguard their online applications from this menace by putting in place efficient detection and prevention mechanisms. This research aims to do just that by doing a comprehensive literature assessment of the many approaches that have been developed to identify and prevent SQL injection attacks. Input validation, parametrized searches, and intrusion detection systems are only a few examples of the methods and their benefits and drawbacks will be discussed throughout the discussion[3].

Before an application processes user-provided data, the data must first undergo a procedure called input validation [4]. A possible indicator of a SQL injection attack is the existence of certain characters or keywords. By decoupling the parameters from the SQL statement itself, parameterized queries provide a safe means of executing SQL statements. [5]. By prohibiting an attacker from inserting harmful code into the query, SQL injection attacks are rendered impossible using this technique.

*Corresponding author. Email: Abdulvugar@mail.ru

SQL statements that have been pre-compiled and saved in the database are known as stored procedures [6]. These methods offer an extra defense against SQL injection assaults and might be utilized to carry out complicated database operations. Network traffic is monitored by intrusion detection systems, often known as IDS, in order to look for indications of malicious activity[7]. An IDS can be programmed to detect SQL injection attacks by monitoring the traffic on a network for particular patterns or features that are linked with SQL injection attacks. This can be done by looking for certain patterns. Honeypots are a type of decoy system that is intended to entice and then capture potential attackers. Honeypots are an effective tool for detecting SQL injection attacks because they watch the behavior of possible attackers and look for any signs of malicious activity. According to Chung et al. (2012), signature-based detection is a method for detecting and preventing SQL injection attacks. This method leverages a database of known attack signatures as its primary resource. Analyzing network information to look for patterns or characteristics that match known attack signatures is one way to use this technique to identify SQL injection attacks. SQL injection attacks, in conclusion, continue to represent a significant security issue to web applications. Input validation, parameterized queries, stored procedures, intrusion detection systems, honeypots, and signature-based detection are some of the various ways of detection and prevention. The method that is used will be decided based on the particular needs of the business as well as the necessary level of safety and protection. Nevertheless, it is likely that the methods of detection and prevention working together will be the most successful.

The following are some of the contributions of the literature review on the prevention and detection of SQL injection attacks:

Compilation of key information: The paper provides a detailed overview of the current state of knowledge on SQL injection attack prevention and detection approaches. It also includes a compilation of important pieces of information. This contains a description of the various types of detection and prevention methods, their respective strengths and limitations, and the relative success of each.

Comparison of different techniques: Comparison of various methods The review analyzes and evaluates a wide variety of methods for both prevention and detection, offering insights into which methods are most effective in a variety of settings through comparison and contrast. This allows businesses to make well-informed decisions about the level of security to implement, according to the unique needs of their operations.

Emphasis on a comprehensive approach: In order to provide the best possible defense against SQL injection attacks, the review stresses the importance of taking a holistic approach to security that incorporates both prevention and detection measures. In particular, the assessment emphasizes the need for a holistic strategy toward security.

Insights into the evolving threat landscape: Understanding the dynamic threat landscape The analysis provides context for the dynamic threat landscape and highlights the need for businesses to adopt cutting-edge security policies to reduce their vulnerability to SQL injection attacks.

In sum, the literature review aids in the development of this sector by offering a practical resource that companies can utilize to implement efficient measures to guard against and identify SQL injection attacks. They want to stop SQL injection attacks from happening at their companies.

2. Prevention Techniques

Several methods exist for protecting against SQL injection attacks, including those that ensure the correct validation of user-provided data and the secure execution of SQL commands. Common safeguards, such input validation, parameterized queries, and stored procedures, will be discussed here.

Input Validation:
The first line of defense against SQL injection attacks is to ensure the input is valid. The user's input must be checked against predefined criteria to ensure that it can be processed by the software. The fundamental goal of input validation is to ensure that no harmful code is delivered to the database as part of a SQL query. Data type validation, range validation, and character set validation are just a few examples of input validation techniques. Checking to determine if the data that was supplied by the user is of the appropriate type—for example, a number or a string—is an example of data type validation. Range validation is the process of checking if the user's input falls within a specified range. To validate a character set, one must check that the user has only used characters that can be read by the software in question. Input

validation can also include a check for the presence of specific characters or keywords that can indicate an attempt at a SQL injection attack. An attempt to inject malicious code into a SQL query can be detected, for instance, by checking user-supplied data for the presence of a single quote character ('). Despite its importance, input validation is often overlooked while developing online applications, leaving them vulnerable to SQL injection attacks[8]. Client-side validation can be accomplished with JavaScript or another client-side scripting language, while server-side validation can be accomplished with a server-side scripting language like PHP or ASP.NET.

Parameterized Queries:
Parameterized queries provide a safe method of executing SQL statements by separating the arguments from the main SQL query. By removing the attacker's ability to insert harmful code into the query, this technique eliminates the possibility of a SQL injection attack. A SQL injection attack can't happen now that this is in place. Since the arguments are not embedded within the SQL statement, but rather presented to the query as standalone variables, an attacker cannot inject harmful code into the query. The best defense against SQL injection attacks is widely acknowledged to be the use of parameterized queries[9]. Most modern programming languages and database systems support them, and their implementation requires no technical knowledge on the part of the user.

Stored Procedures:
Stored procedures are possibly precompiled SQL statements that are kept in the database. These techniques can be used to perform advanced database operations and provide further protection against SQL injection attacks. Stored procedures make it harder for an attacker to inject malicious code into the SQL query by having it performed on the database server rather than the web server. The database server is safer than the web server, and for that reason it is preferred. Data accumulation and processing of massive amounts of data are two examples of difficult database operations that benefit greatly from the use of stored procedures[10]. Because they have already been compiled and optimized for speed, they are also more efficient than dynamic SQL statements, which must be constructed from scratch. In conclusion, input validation, parameterized queries, and stored procedures are the most typical safeguards against SQL injection attacks. Although input validation is a crucial step in avoiding SQL injection attacks, it is often overlooked in the process of creating web applications. Parameterized queries and stored procedures offer an extra line of defense against SQL injection attacks. Because of this, an attacker will have a harder time gaining access to the database and injecting malicious code into a SQL query. To best protect online applications from SQL injection attacks, however, it is generally best to employ a combination of preventative techniques. The company's needs and the desired level of safety will determine which preventative method is used.

3. Detection Techniques

It is crucial for businesses to create and deploy measures to detect SQL injection attacks because of the high expenses they can incur. Log analysis, intrusion detection systems, and honeypots are some of the most frequent ways of detection that we will go through in the following section of this article.

Log Analysis:
Examining log files can help find security holes in a system, a practice known as log analysis. Log files are created by both the web server and the database server and are used to record data about the requests made to the online application. One technique to use log analysis to detect SQL injection attacks is to review the log files for indications of malicious code being run as part of a SQL query. Logs can be analyzed in two ways: by hand or with the use of automated tools. Manual log analysis comprises looking for signs of SQL injection attacks by analyzing log files line by line. Log files may now be analyzed in real time, all owing to automated systems that sound the alarm when they detect signs of an assault. The requests made to a web app can be fully accounted for with the help of log analysis. This greatly facilitates the detection and resolution of security vulnerabilities[11]. The main negatives of log analysis are that it takes time and might be challenging to carry out, especially when working with large log files.

Intrusion Detection Systems:
The term "intrusion detection system" (IDS) is commonly used to describe a group of programs designed to detect threats to a system's safety, such as SQL injection assaults. Network traffic can be analyzed in real time with the help of IDS, which can be deployed on either the web or database server. Multiple techniques, including signature detection, behavior detection, and anomaly detection, are used by IDS to identify potential security threats. Signature-based detection, which will be discussed in more detail below, compares the network traffic to a database of known security concerns. For behavior-based detection, it is necessary to observe not only the web application but also the database. The goal of this

monitoring is to detect any signs of an assault. The anomaly-based detection method involves monitoring the web application and the database and comparing it to a typical baseline in order to seek out symptoms of an attack. Intrusion detection systems (IDS) have the potential to be highly effective in detecting security threats in real time, allowing organizations to swiftly respond to any emerging risks[12, 13]. Because of the potential for IDS to generate several false positives, it can be challenging to distinguish between serious threats and false alarms. This is a significant disadvantage of IDS systems.

Honeypots:
A honeypot is a deception device used to attract and trap would-be intruders. SQL injection attacks can be uncovered with the use of a honeypot, which acts as a vulnerable web application in order to attract hackers' attention. This enables the honeypot to function as an attack target. If an attacker manages to introduce malicious code into the mock web application, the honeypot will detect the attack and provide evidence of it. Because honeypots allow the attacker to try to insert malicious code into the SQL query in a controlled environment, they are very effective at detecting SQL injection attempts. Because of this, the honeypot is able to successfully identify the intrusion attempt. Honeypots have the potential to be very effective in identifying real threats, and they are much less likely to cause false positives than other detection systems. This makes them a desirable choice[14]. The main disadvantage of honeypots is the considerable effort and upkeep involved in establishing and maintaining them. In conclusion, log analysis, intrusion detection systems, and honeypots are the most common methods for identifying SQL injection assaults. It is significantly easier to uncover security concerns and take corrective action when using log analysis since it produces a complete record of all requests sent to a web application. While intrusion detection systems can be useful in spotting actual security threats in real time, they can also generate a lot of false positives. While honeypots are tremendously helpful for detecting SQL injection attacks, they also require a substantial investment of time and resources to set up and maintain.

4. Discussion:

Businesses are at risk from SQL injection attacks because of the severe damage they may cause to databases and the sensitive data they contain. Therefore, it is crucial for companies to establish effective tactics that can detect and avoid such attacks. This literature study discusses a variety of preventative methods, including input validation, parameterized queries, and database security. One simple and successful method of preventing SQL injection attacks is input validation, which involves checking that the user's input is correct and follows the intended format. However, parameterized queries necessitate separating the user-supplied data from the underlying SQL code. Because of this, an attacker will have a harder time inserting malicious code into the SQL query. In conclusion, ensuring the safety of a database requires the installation of safeguards against unauthorized access. Firewalls, encryption, and permission systems are all examples of possible security measures. This literature review discusses several detection techniques, including log analysis, intrusion detection systems, and honeypots. The term "log analysis" refers to the process of inspecting the web server and database server logs for evidence of a SQL injection attack. Intrusion detection systems are software programs whose major purpose is the real-time analysis of network traffic. Network traffic is monitored by these systems in order to detect threats like SQL injection attacks. Finally, honeypots are an example of a decoy system created with the intention of luring and capturing intruders. Since honeypots mimic a web site that can be attacked, they can be used to detect SQL injection attacks. In conclusion, firms may best defend themselves from SQL injection threats by combining preventative and detection methods. Methods for detecting SQL injection attacks include log analysis, intrusion detection systems, and honeypots. Input validation and parameterized queries are basic and effective ways for preventing SQL injection attacks. The threat landscape is dynamic, so it's important for businesses to keep up with the latest innovations in information security technologies and best practices.

5. Conclusion:

SQL injection attacks pose a significant security risk because of the damage they can do to enterprises and the privacy of their data. Businesses need to employ both preventative and investigative measures to counter these threats. This literature review discusses several detection approaches and preventative measures, including log analysis, intrusion detection systems, honeypots, input validation, parameterized searches, and database security. An organization's best defense against SQL injection attacks is a well-rounded security strategy that incorporates many of these techniques. To keep up with the ever-changing nature of security threats, businesses need to implement cutting-edge security technologies and procedures. In this method, businesses can lessen the risk of SQL injection attacks on their private data.

Funding
No financial contributions or endorsements from institutions or sponsors are mentioned in the author's paper.

Conflicts Of Interest

The paper explicitly states that there are no conflicts of interest to disclose.

Acknowledgment

The author expresses gratitude to the institution for the opportunities provided to present and share preliminary findings of this research.

References

[1] W. G. Halfond, J. Viegas, and A. Orso, "A classification of SQL-injection attacks and countermeasures," in *Proceedings of the IEEE international symposium on secure software engineering*, 2006, vol. 1, pp. 13-15: IEEE.

[2] I. S. Crespo-Martínez, A. Campazas-Vega, Á. M. Guerrero-Higueras, V. Riego-DelCastillo, C. Álvarez-Aparicio, and C. Fernández-Llamas, "SQL injection attack detection in network flow data," *Computers & Security,* vol. 127, p. 103093, 2023.

[3] Y.-C. WANG, G.-L. ZHANG, and Y.-L. ZHANG, "Analysis of SQL Injection Based on Petri Net in Wireless Network," *Journal of Information Science & Engineering,* vol. 39, no. 1, 2023.

[4] M. S. Kim, "A Study on the Attack Index Packet Filtering Algorithm Based on Web Vulnerability," in *Big Data, Cloud Computing, and Data Science Engineering*: Springer, 2023, pp. 145-152.

[5] S. K. Shandilya, C. Ganguli, I. Izonin, and A. K. Nagar, "Cyber attack evaluation dataset for deep packet inspection and analysis," *Data in Brief,* vol. 46, p. 108771, 2023.

[6] V. Gorbatiuk and S. Gorbatiuk, "Method of detection of http attacks on a smart home using the algebraic matching method," *PROBLEMS IN PROGRAMMING,* no. 3-4, pp. 396-402, 2023.

[7] M. R. Erlambang, I. W. Hamzah, and F. Dewanta, "Machine Learning Approach for Intrusion Detection System to Mitigate Distributed Denial of Service Attack Based on Convolutional Neural Network Algorithm," *eProceedings of Engineering,* vol. 9, no. 6, 2023.

[8] M. Kumar, "SQL Injection Attack on Database System," *Wireless Communication Security,* p. 183, 2023.

[9] M. Baklizi, I. Atoum, M. A.-S. Hasan, N. Abdullah, O. A. Al-Wesabi, and A. A. Otoom, "Prevention of Website SQL Injection Using a New Query Comparison and Encryption Algorithm," *International Journal of Intelligent Systems and Applications in Engineering,* vol. 11, no. 1, pp. 228-238, 2023.

[10] N. Yadav and N. M. Shekokar, "SQL Injection Attacks on Indian Websites: A Case Study," in *Cyber Security Threats and Challenges Facing Human Life*: Chapman and Hall/CRC, 2023, pp. 153-170.

[11] A. Hadabi, E. Elsamani, A. Abdallah, and R. Elhabob, "An Efficient Model to Detect and Prevent SQL Injection Attack," *Journal of Karary University for Engineering and Science,* 2022.

[12] S. Manhas, "An Interpretive Saga of SQL Injection Attacks," in *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2022, Volume 1*: Springer, 2022, pp. 3-12.

[13] M. Alajanbi, M. A. Ismail, R. A. Hasan, and J. Sulaiman, "Intrusion Detection: A Review," *Mesopotamian Journal of CyberSecurity,* vol. 2021, pp. 1-4, 2021.

[14] D. Chou and M. Jiang, "A survey on data-driven network intrusion detection," *ACM Computing Surveys (CSUR),* vol. 54, no. 9, pp. 1-36, 2021.