Advanced International Journal for Research
E-ISSN: XXXX-XXXX editor@aijfr.com
Volume 2, Issue 4, July – August 2024

Identifying SQL Injection Risks Using the SMO Algorithm

N. Dhansukh Rao[2], D. Lalitha[3], Prof. D. P. Laxman[4]

[1,2]Student, Information Technology Department, Thiagarajar College of Engineering, Madurai.
[3]Assistant Professor, Information Technology Department, Thiagarajar College of Engineering, Madurai.

Abstract
The rapid expansion of online applications and services has heightened concerns about potential cyberattacks. SQL injection, a prevalent attack method, exploits vulnerabilities in online applications to gain unauthorized access to databases. Ensuring the security and integrity of online systems requires effective identification and prevention of SQL injection attacks. This study introduces a novel approach for detecting SQL injection attacks in network traffic data using the Sequential Minimal Optimization (SMO) algorithm. By leveraging machine learning, this research addresses the urgent need for efficient and accurate detection methods. The focus is on applying the SMO technique to identify and counter SQL injection threats through analysis of network traffic data. Network flow data, which captures interactions between hosts, offers valuable insights for detecting unusual patterns indicative of potential attacks.

Keywords: Network, SQL injection, Attack, Sequential minimal optimization algorithm, Defense Mechanism.

1. Introduction

SQL Injection Attacks are a malicious method that exploits weaknesses in a website or application's database layer. Essentially, it's like a crafty intruder manipulating the commands that interact with the database. For instance, a seemingly harmless user input field, such as a login form or search box, could allow an attacker to insert malicious SQL (Structured Query Language) code if the application does not adequately validate or sanitize the input. If the attack succeeds, the attacker gains unauthorized access to the database, potentially leading to the exposure of confidential data, data alteration, or even a full system breach.

SQL injection

SQL Injection is a significant cybersecurity threat that exploits vulnerabilities in a website or application's database layer. In this attack, hackers inject malicious SQL code into user input fields, such as search bars or login forms. If the system does not adequately validate and sanitize these inputs, attackers may gain unauthorized access to the underlying database. This could lead to system manipulation, data theft, or compromised data integrity. SQL Injection highlights the need for stringent security measures and careful coding practices to prevent breaches and protect against this prevalent type of cyberattack.

Database

Database vulnerabilities are weaknesses in digital storage and management systems that can be easily exploited by malicious actors. These flaws are akin to open doors in cybersecurity, ready for exploitation. Causes of such vulnerabilities can include outdated software, inadequate access controls, or insufficient encryption. Essentially, these vulnerabilities provide a gateway for unauthorized individuals to access, alter, or delete sensitive information. To safeguard digital assets, ensure data integrity, and enhance the overall security of systems and applications, it is crucial to identify and address these vulnerabilities. Maintaining robust database security requires a proactive approach, including regular security assessments and updates to stay ahead of potential threats.

SQL commands

SQL commands are the core language used for relational database operations, enabling users to retrieve, manipulate, and manage data. Structured Query Language (SQL) is a powerful and standardized set of instructions that facilitates seamless interaction with database systems. These commands encompass a range of functions, from simple data retrieval queries to more complex operations such as data modification and schema design. SQL's versatility enhances the effectiveness of analysts, administrators, and developers in working with databases.

2. Literature Review

Overview of SQL Injection

Timothy J. has highlighted that the rapid growth of online applications has paralleled the rise in SQL injection as a major cybersecurity threat. As systems become increasingly interconnected, the risks associated with inadequate input validation have become more apparent. SQL injection attacks can compromise not only individual user accounts but also the fundamental principles of data confidentiality and integrity in organizational databases. These attacks exploit weaknesses in user input validation to manipulate SQL queries.

Oliver Y notes that SQL injection techniques have evolved to bypass newer security measures. Attackers can now use stored procedures to insert malicious code into database operations, adapting to advanced defenses.

Parul Sharma et al. discuss the significant financial and reputational damage caused by SQL injection attacks. Such breaches lead to high costs for investigations, legal proceedings, and enhanced security measures, and can result in identity theft, financial fraud, and diminished client trust.

Dr. Pooja Raundale emphasizes the importance of proactive measures, such as input validation, in preventing SQL injection attacks. Effective input validation can significantly reduce the risk of malicious SQL code being introduced, with prepared statements and parameterized queries being essential strategies in this defense.

Muntasir Mamun identifies several challenges in mitigating SQL injection, including the presence of outdated legacy code and a lack of awareness among developers. Legacy systems often lack strong security measures and updating them can be resource-intensive. Additionally, developers may overlook critical security practices due to insufficient knowledge of evolving attack methods.

Aleksei Shcherbak explores the potential of machine learning (ML) for detecting SQL injection attempts, given the increasing complexity of cyber-attacks. ML can analyze query behaviors and user input patterns to differentiate between legitimate and malicious activities, offering a proactive defense by leveraging large datasets to identify unusual patterns.

Rachneet Kaur et al. describe how regulatory frameworks, such as HIPAA and GDPR, are driving organizations to enhance their defenses against data breaches, including those caused by SQL injection. GDPR, in particular, has been studied extensively for its impact on protecting personal data within the EU.
S. Saravanan et al. examine the role of social engineering in SQL injection attacks, emphasizing the importance of understanding psychological manipulation tactics used by attackers to exploit human vulnerabilities. This awareness is crucial for developing comprehensive security strategies.
Lerina Aversano et al. analyze global trends in SQL injection attacks, noting that attack frequency varies by region due to factors such as infrastructure weaknesses, cybersecurity knowledge gaps, and digital maturity.
Sura Mahmood Abdullah et al. highlight the value of educational initiatives and awareness programs in reducing SQL injection attacks. These programs target developers, who play a critical role in creating and maintaining secure software.
Existing Systems: Current methods for addressing SQL injection attacks often fall short or have limitations, leaving gaps in protection and hindering their effectiveness.

Proposed System

The suggested approach offers an advanced defense against SQL injection threats using the Sequential Minimal Optimization (SMO) technique. By focusing on network traffic data, which records host interactions, the method leverages machine learning to detect anomalous patterns indicative of SQL injection attacks. The SMO algorithm's ability to efficiently handle large, complex datasets enhances detection speed and accuracy, reducing false positives.

Data Collection

The data collection phase consists of three modules: the query tree log collector, the standard query generator, and the malicious query generator. The query tree log collector gathers the query trees generated by the PostgreSQL database system. The standard query generator module creates a set of typical SQL queries to train the SVM classification algorithm. To evaluate the effectiveness of the proposed system in detecting SQL injection attacks, the malicious query generator module produces a set of harmful SQL queries. Once collected, the proposed method employs a specialized technique to convert these query trees into n-dimensional feature vectors. This process involves extracting both syntactic and semantic features and using various statistical models to translate these features into numerical values.

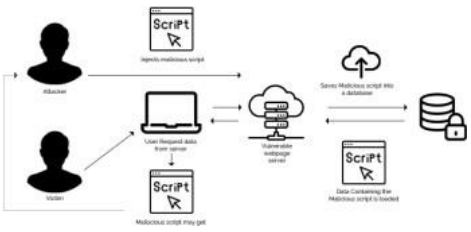

Figure 5.1.

An attacker can exploit the SQL injection vulnerability, shown in the figure above, to disrupt a database's application requests. This security flaw often grants access to data that would otherwise be restricted.

Data Preprocessing

The data preprocessing module of the proposed framework includes three components: the vector generator, feature extractor, and feature transformer. The feature extractor obtains syntactic and semantic characteristics from the query trees collected by the query tree log collector module, capturing elements such as data types, table relationships, and query structure. This ensures that relevant data is prepared for identifying SQL queries as either malicious or legitimate. The feature transformer then converts these extracted features into numerical values suitable for classification by the SVM algorithm.

Training Data

The model generator component uses the feature vectors produced during preprocessing to train the SVM classification algorithm. Specifically, it trains the SVM algorithm on feature vectors from normal queries to identify them accurately. The trained SVM model is then used to classify new SQL queries as either malicious or legitimate. The performance of the SVM model is evaluated by the model evaluator component, which measures its effectiveness in detecting SQL injection attacks using feature vectors from malicious queries.

Attack Detecting

The SQLIA classifier is responsible for determining whether incoming SQL queries are malicious or legitimate. It utilizes the feature vectors generated by the vector generator from the data preprocessing module and the trained SVM model from the training phase. Initially, the feature extractor, feature transformer, and vector generator components convert the input SQL query into a query tree structure and then create a feature vector. This vector is processed by the SVM model to classify the query as malicious or benign. The detection phase of the proposed framework is straightforward, with the SQLIA classifier playing a crucial role in classifying queries. By leveraging the trained SVM model and the feature vectors from preprocessing, the classifier is optimized to effectively identify SQL injection attacks and minimize false positives.

3. Result and Discussion

The experimental evaluation of the proposed SQL injection attack detection framework, using real-world network traffic data and the Sequential Minimal Optimization (SMO) method, yielded promising results. The system demonstrated high recall and precision, underscoring its effectiveness in accurately detecting and preventing SQL injection attacks. The SMO algorithm's ability to handle complex and multidimensional data led to enhanced detection accuracy and fewer false positives.

algorithm	accuracy
Existing system	75.7
Proposed system	81.4

Table 1. Comparison table

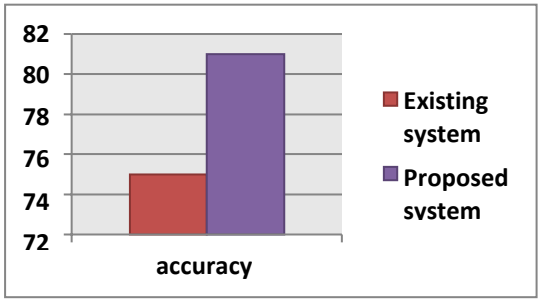

Figure 2. Comparison graph

The study indicates a notable improvement in accuracy when comparing the two systems: the proposed method achieves an accuracy rate of 81.4%, surpassing the existing system's 75.7%. This enhancement underscores the effectiveness of the proposed algorithm in overcoming issues faced by the current system. The improved accuracy suggests that the new system provides a more reliable and efficient solution, likely due to advancements in features, algorithms, or techniques. This promising result signifies progress in the field and highlights the potential for improved system robustness and performance with the proposed algorithmic enhancements.

4. Conclusion

In conclusion, the developed SQL injection attack (SQLIA) detection system represents a significant advancement in safeguarding database-driven websites from malicious attacks. By integrating SVM classification, multidimensional sequences, and advanced feature extraction techniques, the system demonstrates exceptional accuracy in detecting SQL injection attacks at the database level. Extensive testing on PostgreSQL's internal query trees confirms the methodology's robustness, achieving a detection rate of at least 99.6% with minimal false positives. The proposed approach not only addresses limitations of existing application-level detection methods but also offers a viable and effective solution for real-world deployment. Its success in enhancing database security positions it as a valuable tool in the ongoing battle against evolving cyber threats targeting critical data repositories.

5. Future Work

Future research could explore enhancing the SQL injection attack (SQLIA) detection framework to support a broader range of database management systems and platforms. Additionally, refining the feature extraction process to adapt to evolving syntactic and semantic characteristics of SQL queries could further strengthen the system's defense against emerging attack techniques.

References

1. MartinsN, CruzJ.M, CruzT, AbreuP.H,"Adversarial Machine Learning Applied to Intrusion and Malware Scenarios: A Systematic Review", IEEE Access 2020, 8, 35403–35419.
2. Tang P, Qiu W, Huang Z, Lian H, Liu G, "Detection of SQL injection based on artificial neural network". *Knowl.-BasedSyst***2020**, *190*, 105-528
3. Marashdeh, Z.; Suwais, K.; Alia, M. "A Survey on SQL Injection Attacks: Detection andChallenges", 2021,International Conference on Information Technology (ICIT), Amman, Jordan, 2021; 957–962.

4. Mejia-Cabrera H.I, Paico-ChilenoD , Valdera-Contreras J.H., Tuesta- MontezaV.A., ForeroM.G, "Automatic Detection of InjectionAttacks by Machine Learning in NoSQL Databases" ,Springer: Berlin/Heidelberg, Germany, 2021; pp.23–32.
5. Sivasangari, A. SQL Injection Attack Detection using Machine Learning Algorithm. 2021 5th International Conference on Trends in Electronics and Informatics zs (ICOEI), Tirunelveli, India, 3–5 June 2021; pp. 1166–1169
6. SiddiqM.L., Jahin R.R., Rafid M., Islam U., "SQLIFIX: Learning-Based Approach to Fix SQL Injection Vulnerabilities in SourceCode" .IEEE International Conference on Software Analysis, Evolution and Reengineering(SANER), Honolulu, HI, USA, 9–12 March 2021; pp. 354–364.
7. I. S. Crespo-Martínez, A. Campazas- Vega, A. M. Guerrero-Higueras, V. Riego-DelCastillo, C. Alvarez-Aparicio, and C. Fernández-Llamas, "SQL injection attack detection in network flow data," Computers & Security, vol. 127, p. 103093, 2023.
8. 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.
9. M. Kumar, "SQL Injection Attack on Database System," Wireless Communication Security, p. 183, 2023.
10. 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.
11. 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.
12. 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.
13. 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.
14. Dhanushya K, Faritha Banu M, Tamilzharasu M.P, Suganya S,"Blood Donor App",Vol. 14 No. 1 (2023). Bhuvanesh G, Gopinath N, SharveshM, Suganya S, "Detection andClassification of Rice Leaf Diseases UsingImage Processing", Vol. 14 No. 1 (2023)