Ministry of Education of the Republic of Belarus
Educational institution
Belarusian State University of Informatics and Radio electronics

UDC 004.056

Deng Yuanyuan

SQL Injection Attack and Prevention Technology

Abstract
for a Master's Degree
Specialty 7-06-0611-02 Information Security

_

Supervisor:
Vrublevsky L.A., PhD, Professor

_

Minsk 2025

INTRODUCTION

SQL Injection Attack (SQLIA) is one of the most common and harmful security vulnerabilities in Web applications. With the rapid development of the Internet, Web applications are more and more widely used in various fields, and the database, as a core component of Web applications, stores a large amount of sensitive information (e.g., user data, transaction records, etc.). SQL Injection Attacks inject malicious SQL code into the database by taking advantage of the application's improper handling of user input to bypass authentication, steal data, tamper with data, and even completely control the database server. , tamper with data or even take full control of the database server.

This paper helps developers and security researchers better understand the hazards of SQL injection attacks by examining their principles and types. Compare the performance and application scenarios of commonly used SQL injection detection tools to provide developers with selection references. For parameterized queries, the following work is done to analyze SQL injection statements from the semantic point of view and to study the detection methods of SQL injection attack statements based on semantics:

The malware detection method based on text and security features is experimentally validated. Firstly, the input data are cleaned and normalized by text preprocessing techniques, including tokenization, deactivation and stemming extraction. Then, the inadequacy of TF-IDF for the classification of some specific SQL injection attack statements is examined, and the TF-WIDF method is improved to extract text features and combine them with security features (e.g. malicious code features) for feature fusion to form a comprehensive feature vector. Subsequently, machine learning models (e.g., plain Bayes) are used for training to achieve malware identification and classification. Finally, the effectiveness of the approach is validated by evaluating the model performance (e.g., accuracy, recall, and F1-score). The overall approach aims to improve the accuracy and efficiency of malware detection and effectively prevent detecting SQL injection risks. It provides more reliable technical support for network security.

GENERAL DESCRIPTION OF WORK

The Research Aim and Objectives

The aim of the research is to analyze the existing types of SQL injections as well as prevention methods and to develop an improved SQL injection detection method based on the results of these analyses.

To achieve the aim, it is necessary to achieve the following objectives:

1) Study the mechanisms of SQL injection attacks, including syntax patterns, execution processes, and common attack types (e.g., union-based, blind, time-based). Categorize attack variants to understand threat diversity.

2) Analyze strengths and weaknesses of traditional approaches (static/dynamic analysis, regex matching) and machine learning models TF-IDF.

3) Design advanced feature extraction methods that combine term frequency, inverse document frequency, and semantic weighting. Integrate security-specific features (e.g., dangerous function detection, symbol anomaly metrics) to enhance payload discrimination.

4) Optimize machine learning models (e.g., modified logistic regression) to reduce false negatives/positives. Address class imbalance issues using resampling or cost-sensitive learning.

5) Develop hybrid defense frameworks combining preprocessing (input validation, parameterization), runtime monitoring, and anomaly detection.

6) Construct a labeled dataset with diverse attack samples (e.g., SQLmap-generated payloads) and normal traffic. Evaluate model performance using metrics like accuracy, F1-score, ROC-AUC, and testing against baseline tools (e.g., SQLmap vs. TF-WIDF-based system).

Work Connection with Priority Areas of Scientific Research

The topic of the thesis corresponds to paragraph 6 Ensuring the security of humans, society and the state (means of technical and cryptographic information protection, cryptology and cybersecurity) of the list of priority areas of scientific, scientific-technical and innovative activities for 2021–2025, approved by the Decree of the President of the Republic of Belarus dated 07.05. 2020 No. 156.

Thesis Results Approbation

The main results of the dissertation were reported and discussed at the 60th Scientific conference of Graduate Students, Master Students and Students of BSUIR (Minsk, April 22, 2024), Belarusian-Chinese Youth Innovation Forum “New Horizons – 2024” (Minsk, November 21, 2024).

CHAPTER 1 SQL INJECTION RESEARCH PREVENTION BACKGROUND

Figure 1.1 shows the change in ranking of the top ten most serious Web security vulnerabilities published by OWASP (Open Web Application Security Project) from 2013 to 2023. Injection vulnerabilities topped the list twice in a row between 2013 and 2017, and are ranked third in 2021, a slight decline in position but a strong indication of the threat. This third place is evidence that the threat is still strong enough to be overcome.

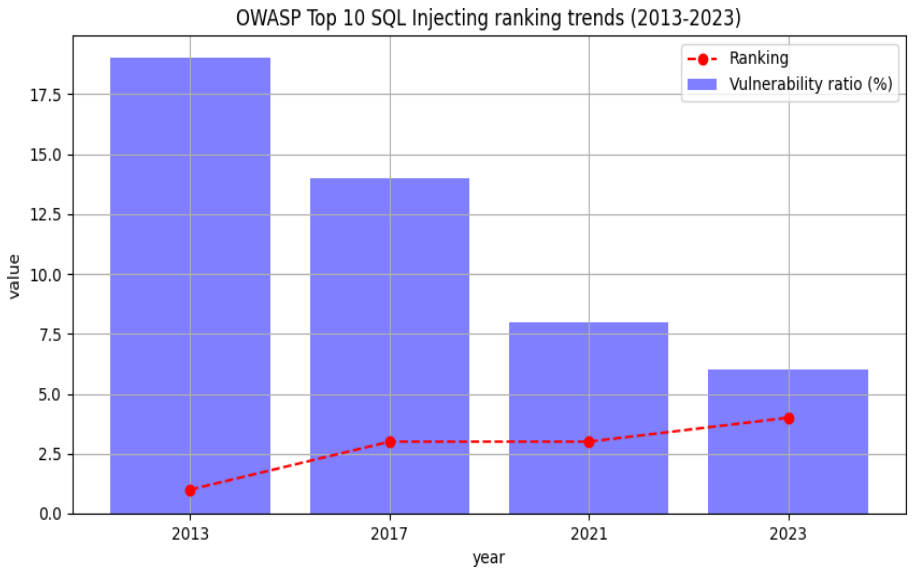

Figure 1 – Ranking of SQL Injection Vulnerabilities 2013-2023

SQL Injection Attack (SQLIA) is one of the most common and harmful security vulnerabilities in Web applications. Although existing defense techniques and detection tools have been able to effectively counter most SQL injection attacks, with the continuous evolution of attack techniques, future research directions could include automated defence systems, application of machine learning and deep learning, and developer security awareness training. The security of applications and databases can be significantly improved through in-depth research into the principles, types and defence techniques of SQL injection attacks, combined with effective detection tools and feature modelling methods.

CHAPTER 2 COMPARISON OF SQL INJECTION ATTACK TOOLS AND TEXT DETECTION METHODS

This chapter provides a detailed description of the principles, causes, and attacks of SQL injections, and compares the tools for detecting SQL injections, and finally provides a comprehensive analysis of text detection methods, and special classifications. SQLmap and JQSL have their own strengths and weaknesses and are suitable for different user needs and skill levels. SQLmap is suitable for security specialists who need powerful functionality, whereas jSOL provides a user-friendly graphical interface for beginners. In terms of text detection, the feature matching approach is simple and efficient, but vulnerable to sophisticated attacks, while the machine learning approach offers higher accuracy and adaptability but is relatively complex to implement. In the future, a hybrid detection strategy combining these two approaches may become a more effective SQL injection defense solution.

In summary, the detection and defense against SQL injection attacks requires comprehensive consideration of the functionality and ease of use of the tools as well as the effectiveness of the detection methods in order to improve the overall security of the system. By continuously optimizing the detection tools and methods, we can better cope with the increasingly complex network security threats.

CHAPTER 3 SQL INJECTION DETECTION METHOD BASED ON KEY FEATURE EXTRACTION

3.1 Data set generation

Through research and investigation, we found that there is no standard dataset for SQL injection attack, this paper uses public non-standard dataset as the main dataset, the main dataset is GitHub open source SQL injection attack dataset, and the supplemental dataset, the SQL injection attack samples collected by SQLmap+Burpsuite, are used to generate the dataset for feature modeling after data cleansing and preprocessing.

3.2 Data Cleaning and Feature Preprocessing

In SQL injection attack detection, the raw input data usually contains a large amount of noise (e.g. annotations, special symbols, deactivated words, etc.), which directly affects the efficiency and accuracy of subsequent feature modelling. This section proposes a preprocessing process based on regular expression cleaning and word splitting optimization as follows: Regular expression cleaning, Disabling word filter. By removing deactivated words, performing stemming extraction and word shape reduction, text normalization, and weighting adjustment, the quality of text data can be significantly improved, thus enhancing the performance of subsequent models. These steps lay a solid foundation for text feature extraction and analysis.

3.3 TF-IDF feature modelling

TF-IDF keyword extraction algorithm (Term Frequency-Inverse Document Frequency) is a common weighting technique widely used in the field of information retrieval and text keyword extraction. The idea of TF-TDF algorithm is that, if a word has a higher frequency in a certain document with high frequency and at the same time with low frequency in other documents, it is considered that the word can better express the characteristics of the document and is suitable for classification, and the word is given a higher weight. The word frequency weight, which represents the frequency of occurrence of a given word in a document, is expressed as TF, and the calculation formula is shown in Equation 3.1.

$$TF_{(t)} = \frac{df(t)}{N} \tag{3.1}$$

Where *df(t)* represents the frequency of word *t* in the document, and *N* represents the total number of words in the document.

$$IDF_{(t)} = \log log \left(\frac{N}{F(t)} + 0.01\right) \quad (3.2)$$

The result of the TF-IDF algorithm is the result of multiplying the TF and IDF values, which indicates the final weight of a word in the document, as shown in Equation 3.3.

$$\text{TF-IDF=TF*IDF} \quad (3.3)$$

Table 1 TF-IDF modeling yields the following results

TF-IDF Model	precision	recall	f1-score
0	0.92	0.99	0.96
1	0.99	0.86	0.92
accuracy			0.94
macro avg	0.95	0.92	0.94
weighted avg	0.95	0.94	0.94

3.4 Improved TF-WIDF modelling assessment

Fig. 2 shows the flowchart of the improved TF-WIDF detection process

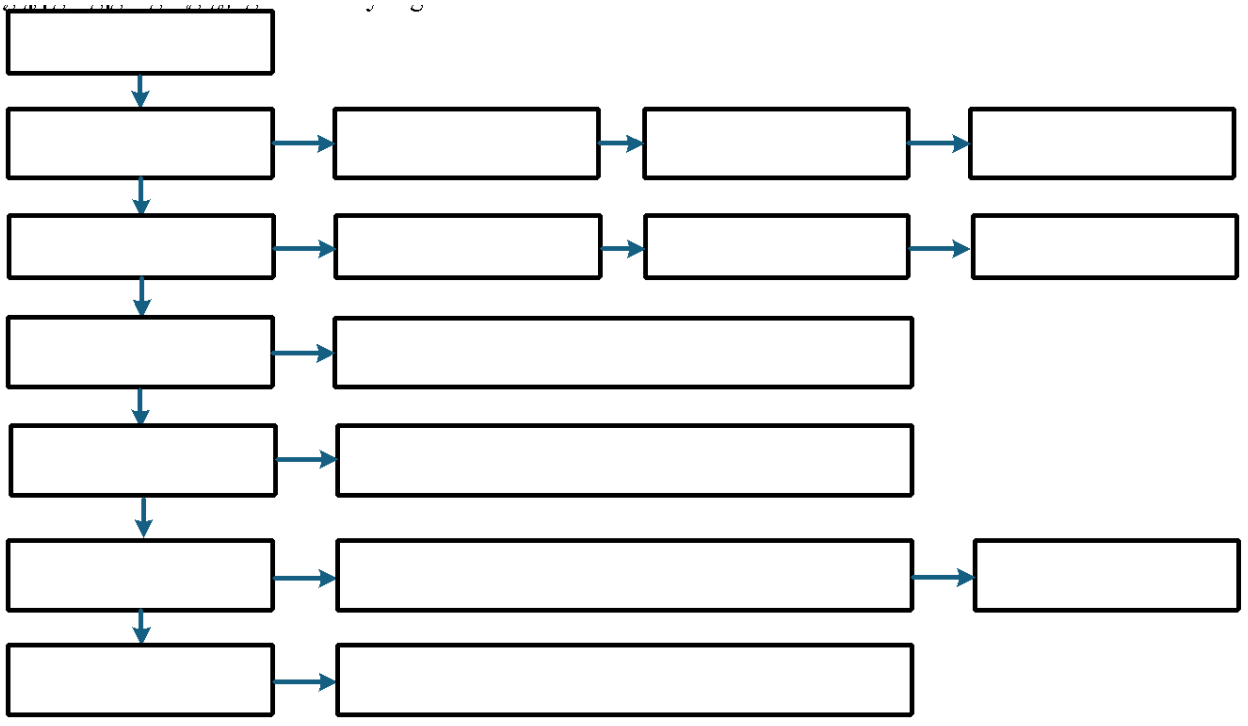

Figure 2 – Flowchart for TF-WIDF model detection

3.4.1 Improvement of the IDF algorithm to smooth the IDF

When calculating the IDF, a smoothing method (e.g., plus 1 smoothing) is used to avoid extreme penalties for low-frequency words. Formula

$$\text{IDF}(t)=\log log\left(\frac{N+1}{df(t)+1}\right) + 1 \tag{3.4}$$

where N is the total number of documents and $df(t)$ is number of documents containing the word t

Weighting of TFs Logarithmic TFs, TFs are calculated using a logarithmic approach to reduce the effect of high frequency words:

$$\text{TF}(t, d) =\log log\,(1 + f(t, d)) \tag{3.5}$$

Where $f(t, d)$ is the frequency of the occurrence of the word t in document d.

Square root TF further reduces the effect of high frequency words and makes the TF value smoother, thus improving the performance of the model.

$$\text{TF}(t, d) =\sqrt{f(t, d)} \tag{3.6}$$

3.4.2 security feature enhancements

The Security Feature Enhancement Module aims to complement the traditional word frequency features in SQL injection detection through syntax-level attack pattern detection and anomalous behavior statistics. The module contains four types of core features, which enhance the model's ability to discriminate malicious queries from four dimensions: dangerous function detection, parameterized risk analysis, symbolic anomaly metrics, and attack pattern recognition, respectively.

Tabel 2 TF-IDF modeling yields the following results

TF-WIDF Model	precision	recall	f1-score
0	0.98	0.99	0.99
1	0.98	0.97	0.98
accuracy			0.98
macro avg	0.98	0.98	0.98
weighted avg	0.98	0.98	0.98

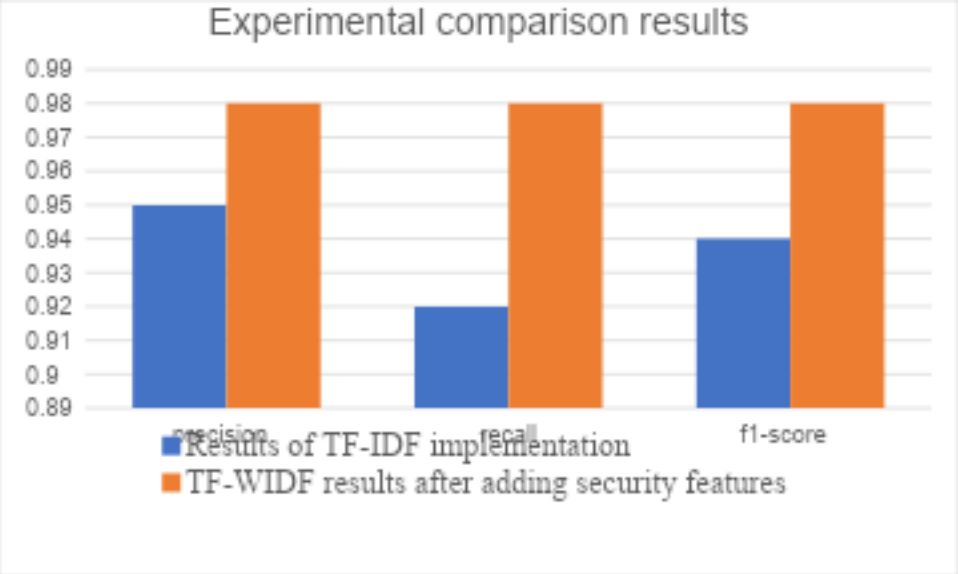

Figure 3 – Comparison of experimental results between TF-IDF and TF-WIDF models

3.5 Summary

Through these steps of data preprocessing, weighting adjustment and security feature enhancement, the effectiveness of the TF-WIDF model can be significantly improved. Removing noise, unifying the text format, optimizing the weighting calculation, and also enhancing the security features can better reflect the importance of the words in the text, thus improving the performance of the model in text analysis and information retrieval tasks.

CONCLUSION

This paper summarizes the principles, types, hazards and existing detection methods of machine learning through in-depth study of SQL injection attacks, than analyses the two most common tools for detecting SQL injection, SQLmap and jSQL, and proposes two inspection methods based on TF-IDF (Term Frequency-Inverse Document Frequency) and TF WIDF (Term Frequency-Weighted Inverse Document Frequency) based on TF-IDF (Term Frequency-Inverse Document Frequency) and TF-WIDF (Term Frequency-Weighted Inverse Document Frequency).

TF-IDF: Measures the importance of words in a document by calculating Term Frequency and Inverse Document Frequency.TF-IDF can effectively identify key features in SQL injection attack statements.

TF-WIDF: On the basis of TF-IDF, weighting adjustment and security feature enhancement are introduced to further optimize the feature extraction. By removing preprocessing steps such as deactivated words, stemming (Stemming) and lemmatization (Lemmatization), TF-WIDF is able to better reflect the semantic features of SQL injection attack statements.

The experiment verifies the effectiveness and accuracy of the TF-WIDF defense technique. It is hoped that the research results in this paper can provide practical technical guidance for developers to help them better prevent SQL injection attacks.

LIST OF OWN PUBLICATIONS

1–A. Deng Y. Y. SQLI attacks and prevention / Y.Y. Deng // The 60th Scientific conference of Graduate Students, Master Students and Students of BSUIR; April 22-26, 2024. –Minsk: BSUIR. – P. 11-14.

2–A. Deng Y. Y. A Strategies for enhancing database security against SQL injection / Y.Y. Deng // XI Belarusian-Chinese Youth Innovation Forum, November 21, 2024. –Minsk: BNTU. – T. 1.– P. 59-61.