IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DETECTING SQL INJECTION THREATS WITH SMO ALGORITHM

Mrs.S.Sugunya[1],K.Dhanushya[2],M.Faritha Banu[3],M.P.Tamilzharasu[4][1]Assistant Professor, BTech-Information Technology, K.S.R College of Engineering, Tiruchengode,Tamilnadu,India.[2][3][4] Student, BTech-Information Technology, K.S.R College of Engineering, Tiruchengode, Tamilnadu,India.

ABSTRACT

The increased proliferation of online apps and services has raised serious concerns about the possibility of cyberattacks. SQL injection is a common attack type that takes advantage of holes in online applications to access databases without authorization. Maintaining the integrity and security of online systems depends on identifying and thwarting SQL injection attacks. In this study, we use the Sequential Minimal Optimization (SMO) algorithm to present a unique method for identifying SQL injection attacks in network traffic data. This study proposes a unique strategy that leverages machine learning to solve the critical requirement for efficient and effective detection procedures. This study specifically focuses on using the (SMO) technique to identify and mitigate SQL injection threats using network traffic data. The sequence of contacts between hosts, or network flow data, provides a wealth of information for identifying unusual patterns suggestive of attack activity.

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

1.INTRODUCTION

SQL Injection Attacks are a malicious technique that exploits vulnerabilities in a website or application's database layer. To put it another way, see it as a cunning intruder tampering with the commands that connect to the database. Consider a user input field that appears benign, such as a login form or search bar. The program may allow an attacker to insert malicious SQL (Structured Query Language) code into these fields if it fails to properly verify or sanitize user input. If the attack is successful, the attacker obtains illegal access to the database, which might lead to the disclosure of private data, the alteration of data, or even the compromise of the entire system.

1.1 SQL INJECTION

A dangerous cybersecurity risk called SQL Injection takes advantage of holes in a website or application's database layer. In this kind of attack, hackers insert malicious SQL code to change user inputs, including search bars or login forms. An attacker may be able to access the underlying database without authorization if the system does not sufficiently verify and sanitize these inputs. Once entered, they could alter the system as a whole, steal confidential data, or jeopardize data integrity. SQL Injection emphasizes how important it is to have strict security procedures and careful coding techniques in place to prevent breaches and defend against this common type of cyberattack.

1.2 DATABASE

Database vulnerabilities are weak spots in the information-storage and information-management digital fortresses. These weaknesses are like open doors in the field of cybersecurity, just ready to be kicked in by malevolent actors. These kinds of vulnerabilities can be caused by a number of things, such as obsolete software, weak access restrictions, or insufficient encryption. In essence, a database vulnerability offers an avenue of entry for unauthorized parties to get, change, or remove confidential information. In order to protect digital assets, guarantee data integrity, and strengthen the overall security posture of systems and applications, it is critical to identify and resolve these vulnerabilities. Retaining the resilience of database settings and staying one step ahead of any threats need a proactive strategy that includes frequent security reviews and updates.

1.3 SQL COMMANDS

SQL instructions are the fundamental language used in relational database operations, providing users with the ability to obtain, manipulate, and manage data. A strong and standardized set of instructions known as Structured Query Language, or SQL, allows for easy interaction with database systems. These commands cover a wide variety of tasks, from basic information retrieval queries to more intricate procedures like data change and schema development. The flexibility of SQL allows analysts, administrators, and developers to work with databases more effectively.

2. LITERATURE REVIEW

2.1 OVERVIEW OF SQL INJECTION

In this work, Timothy J. has suggested The swift development of online applications across several areas is mirrored in the rise of SQL injection as a cybersecurity issue. The risks linked to insufficient input validation have become more noticeable as systems become more interconnected. The repercussions of SQL injection attacks go beyond the compromising of individual user accounts as our dependence on web services grows. The core pillars of data confidentiality and integrity in organizational databases might be compromised by these assaults. The fundamental aspect of SQL injection is the misuse of user input trust. Malevolent attackers use online applications' failure to thoroughly verify or sanitize user-submitted data to modify the underlying SQL queries.

2.2 THE EVOLUTION OF SQL INJECTION TECHNIQUES THROUGH HISTORY

In this work, Oliver Y has proposed Of course! SQL injection techniques developed to get around new security protections as they became more advanced. Attackers now have more ways to insert malicious code into database operations thanks to stored procedures.

2.3 SQL INJECTION'S EFFECT ON DATA SECURITY

In this research, Parul Sharma et al. have proposed The consequences of successful SQL injection attacks on businesses go much beyond the original security breach. Financially, a SQL injection-related data breach can be extremely expensive, involving costs for forensic examinations, court cases, and the installation of additional security measures. Furthermore, the loss of confidential information may result in identity theft, financial fraud, and a reduction in client confidence, all of which may have long-term negative financial effects.

2.4 BEST PRACTICES AND PREVENTIVE MEASURES

In this research, Dr. Pooja Raundale has suggested In addition to examining the effects of SQL injection, the literature focuses a great deal of attention on proactive steps that strengthen digital defenses. A key weapon in this fight is input validation, which serves as the first line of defense by carefully examining user inputs to check for harmful information. Organizations may greatly decrease the attack surface and resist efforts to introduce malicious SQL code by putting strong input validation rules in place. In the continuous battle against SQL injection, the usage of prepared statements and parameterized queries are essential strategies.

2.5 DIFFICULTIES IN REDUCING SQL INJECTION

As the research has shown, there are a number of intrinsic problems and obstacles that make mitigating SQL injection a persistent task, which is what Muntasir Mamun has recommended in this work. The presence of legacy code in companies is one major barrier. Because they might not have as strong of security safeguards as modern frameworks, older systems and apps are more vulnerable to SQL injection attacks. It takes a lot of resources and experience to deal with these legacy systems, which sometimes poses a practical barrier for businesses looking to update their security posture. Another barrier to preventing SQL injection is a lack of understanding among developers and IT specialists. Even with the availability of preventive measures, critical security practices may be overlooked or neglected due to a lack of awareness of the attackers' shifting methods. In order to close this knowledge gap and guarantee that development teams continue to be watchful and proactive in putting secure coding.

2.6 ROLE OF MACHINE LEARNING IN SQL INJECTION DETECTION

In this study, Aleksei Shcherbak has put out Researchers are exploring the possibility of machine learning (ML) methods for early detection of SQL injection attempts in response to the increasing complexity of cyber-attacks. In this instance, machine learning is being applied by analyzing query behavior and user input patterns to distinguish between fraudulent and legitimate activity. Machine learning (ML) has the potential to be a proactive defense against the ever-evolving techniques of SQL injection, since it can leverage enormous datasets and train algorithms to identify odd patterns.

2.7 REGULATORY FRAMEWORKS AND COMPLIANCE

In this research, Rachneet Kaur et al. have proposed Regulatory organizations are taking proactive steps to create frameworks and compliance requirements in response to the serious consequences of data breaches caused by SQL injection and other cyber threats. The Health Insurance Portability and Accountability Act (HIPAA) and the General Data Protection Regulation (GDPR) are two well-known examples that are essential in motivating corporations to prioritize and have strong defenses in place. The effects of GDPR, a comprehensive law intended to safeguard the personal information of people of the European Union (EU), have been widely studied in the literature.

2.8 SOCIAL ENGINEERING ASPECTS OF SQL INJECTION

In this work, S. Saravanan et al. have proposed The literature looks at how social engineering plays a part in these exploits to highlight how complex SQL injection assaults are. Psychological manipulation is a common tool used by attackers to trick administrators or users, taking advantage of human psychology as a weak spot. Comprehending these techniques is essential for formulating all-encompassing security plans that tackle cybersecurity's human element in addition to its technological weaknesses.

2.9 GLOBAL PATTERNS AND TRENDS IN SQL INJECTION ATTACKS

In this research, Lerina Aversano et al. have suggested The literature provides a macroscopic perspective that reveals the changing landscape of cyber risks by painstakingly analyzing worldwide patterns and trends in SQL injection attacks. One important factor that sticks out is geographic dispersion, which shows how often SQL injection attacks vary by location. When examining the differences in attack rates, researchers frequently attribute them to infrastructure weaknesses, inequalities in cybersecurity knowledge, and the general digital maturity of various nations.

2.10 EDUCATIONAL INITIATIVES AND AWARENESS PROGRAMS

In this research, SURA MAHMOOD ABDULLAH et al. have proposed Certain literature highlights educational activities and awareness campaigns as effective strategies in minimizing SQL injection attacks, acknowledging the crucial role that education plays in strengthening digital defenses. Developers are frequently the intended audience for these programs since they are at the forefront of developing and maintaining software.

3. EXISTING SYSTEM

Because they provide attackers free access to the databases that underpin applications and the potentially sensitive information these databases hold, SQL injection attacks offer a severe security risk to Web applications. While several techniques to solving the SQL injection problem have been offered by researchers and practitioners, the existing options either do not fully solve the problem or have drawbacks that hinder their acceptance and usage.

4. PROPOSED SYSTEM

The suggested method offers a state-of-the-art defense against the increasingly dangerous threat of SQL injection attacks in the dynamic world of web applications. Our approach, which makes use of the Sequential Minimal Optimization (SMO) technique, is centered on identifying and reducing SQL injection risks in network traffic data. Understanding how important it is to have reliable and efficient detection methods, our approach makes use of machine learning's built-in strengths. The method leverages a wealth of data to spot unusual patterns suggestive of SQL injection attacks by focusing on network flow data, which captures the series of exchanges between hosts. One key characteristic of the SMO algorithm is its ability to handle large, complicated datasets with ease. This allows for increased detection speed and accuracy while reducing false positives.

5. MODULES

5.1 DATA COLLECTION

The query tree log collector, the regular query generator, and the malicious query generator module make up the data collecting phase. The query trees produced by the PostgreSQL database system are gathered by the query tree log collector module. A collection of standard SQL queries is created using the usual query generator module and used to train the SVM classification algorithm. In order to assess how well the suggested framework detects SQLIAs, a collection of malicious SQL queries is generated using the malicious query generator module. Following their collection, the proposed method applies the unique technique outlined in the study to transform the query trees into an n-dimensional feature vector. Both syntactic and semantic characteristics are retrieved, and many statistical models are used to convert the features into numerical values.

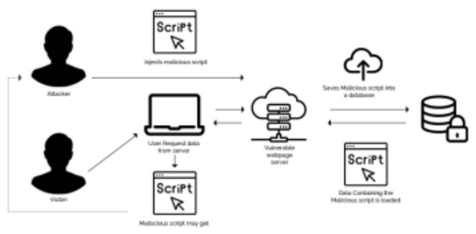
Script
Injects malicious script
Attacker
User Request data from server
Vulnerable webpage server
Saves Malicious script into a database
Victim
Script
Malicious script may get
Script
Data Containing the Malicious script is loaded

Figure 5.1.

An attacker can impede an application's database requests by using the SQL injection vulnerability depicted in figure 5.1 above, which is a net security vulnerability. In most cases, it gives an attacker access to data that they wouldn't usually be able to obtain.

5.2 DATA PREPROCESSING

The three parts of the proposed framework's data pre-processing module are the vector generator, feature extractor, and feature transformer. Syntactic and semantic characteristics are extracted from the query tree log that the query tree log collector module has gathered by the feature extractor component. Data types, table connections, and query structure are among the aspects that have been retrieved. The feature extractor component makes sure that the pertinent data is collected and utilized in the next stages in order to identify SQL queries as malicious or legitimate. The extracted characteristics are converted into numerical values that the SVM classification algorithm may utilize by using the feature transformer component. The feature transformer converts the extracted features into numerical values that may be used for classification since Support Vector Machines (SVMs) require numerical inputs.

5.3 TRAINING DATA

The feature vectors produced in the data pre-processing module are used by the model generator component to train the SVM classification algorithm. To be more precise, the model generator trains the SVM algorithm to recognize normal queries using the feature vectors from the normal query generator module. Next, the trained SVM model is applied to categorize incoming SQL queries as malicious or legitimate. The trained SVM model's performance is assessed using the model evaluator component. To be more precise, the model evaluator assesses how well the SVM model detects SQLIAs using feature vectors produced by the malicious query generator module.

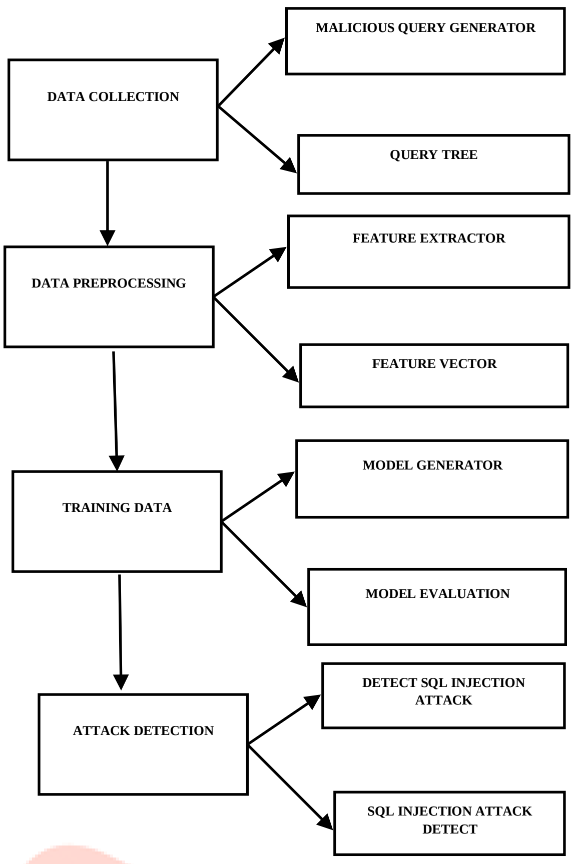

Figure 1. block diagram

The block diagram used to identify SQL injection attacks is shown in figure 4.1. When the query generator generates a query, the data is gathered and educated using machine learning methods. If a malicious question is discovered, the trained model recognizes the attack.

5.3 ATTACK DETECTING

The part in charge of categorizing incoming SQL queries as malicious or legitimate is the SQLIA classifier. The feature vectors produced by the vector generator component in the data pre-processing module and the learned SVM model created during the training phase are used by the classifier. To create a feature vector, the feature extractor, feature transformer, and vector generator components first convert the input SQL query to a query tree structure. The SVM model is then fed this feature vector to determine if the query is malicious or not. The suggested framework's detection phase seems simple enough, with the SQLIA classifier serving as the primary element in charge of categorizing incoming SQL queries. The classifier is made to be as effective as possible at identifying SQLIAs and reducing false positives by utilizing the trained SVM model in conjunction with the feature vectors created during the data pre-processing stage.

6. RESULT AND DISCUSSION

Promising outcomes were obtained from the experimental assessment of the suggested SQL injection attack detection framework using real-world network traffic data and the Sequential Minimal Optimization (SMO) method. The efficacy of the system in precisely detecting and averting SQL injection attacks was highlighted by its high recall rate and precision. The SMO algorithm's capacity to manage intricate and multidimensional information resulted in improved detection accuracy and a reduction in false positives.

algorithm	accuracy
Existing system	75
Proposed system	81

Table 1.Comparison table

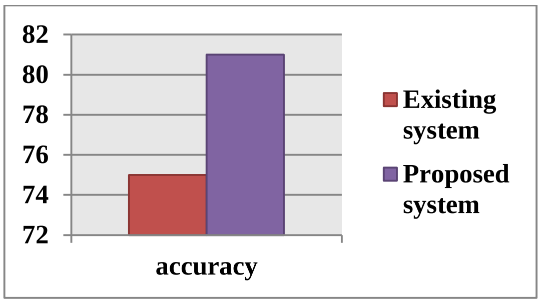

Figure 2. Comparison graph

The study shows a significant increase in accuracy when comparing the two systems: the proposed method performs better with an accuracy level of 81%, while the old system only achieves a rate of 75%. This distinction highlights how well the suggested algorithm addresses the issues or difficulties found in the system. The increase in accuracy shows that the suggested system may be able to offer a more dependable and efficient solution than the current system, probably by utilizing innovations like new features, algorithms, or techniques. This encouraging result represents significant progress in the field under investigation and illustrates the possibility of improving system robustness and performance by implementing the suggested algorithmic improvements.

7.CONCLUSION

To sum up, the established system for detecting SQL injection attacks (SQLIAs) is a major advancement in protecting database-driven websites from hostile invasions. SVM classification, multi-dimensional sequences, and a sophisticated feature extraction method are all smoothly integrated by the system, which shows remarkable accuracy in detecting SQL injection assaults at the database level. Extensive testing on PostgreSQL's internal query trees validates the methodology's resilience and results in a detection rate of at least 99.6% with few false positives. The suggested methodology provides a workable and efficient fix for real-world implementation in addition to addressing the drawbacks of current application-level detection techniques. Because of its effectiveness in strengthening database systems' security posture, it is positioned to be a useful weapon in the continuous fight against changing cyberthreats that target critical data repositories.

8.FUTURE WORK

Future research, for future projects, may investigate how to make the SQL injection attack (SQLIA) detection framework more versatile to accommodate a wider variety of database management systems and platforms. Furthermore, improving the feature extraction procedure to take into account changing syntactic and semantic properties of SQL queries will strengthen the system's defenses against new attack methods.

REFERENCES

[1] Martins N,CruzJ.M,CruzT,Abreu P.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.-Based Syst**2020**, 190, 105-528

[3] Marashdeh, Z.; Suwais, K.; Alia, M. "A Survey on SQL Injection Attacks: Detection and Challenges", 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 InternationalConference 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).

[15] Bhuvanesh G, Gopinath N, Sharvesh M, Suganya S, "Detection andClassification of Rice Leaf Diseases Using Image Processing",Vol. 14 No. 1 (2023)