JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

OUTBREAK DETECTION AND PREVENTION TECHNIQUE OF SQL INJECTION ATTACKING USING MACHINE LEARNING

[1]Dr.Vineetha KR, [2]Jeena Thomas

[1]Associate professor, [2]MCA Scholar

[1] Department of MCA,

[1]Nehru College of Engineering and Research Centre, Pampady, India

Abstract : Online application assaults are becoming increasingly common and severe. The large amount of data accessible on the internet motivates hackers to initiate novel attacks. Extensive study on web application security has been done in this area. Structured Query Language Injection is the most hazardous online application exploit (SQLI). This attack poses a significant risk to online apps. Several studies have been carried out in order to reduce this assault, either by avoiding it at an early stage or spotting it when it happens. We give an overview of the SQL injection attack as well as a classification of the freshly suggested detection and prevention methods in this article. This paper discusses the methodology and analysis of using machine learning techniques for SQL injection attack detection and prevention. The report covers techniques such as feature selection, model training, and evaluation, and presents various evaluation metrics such as true positive rate, false positive rate, accuracy, precision, recall, F1 score, training, and testing time. The report also emphasizes the importance of using machine learning techniques in combination with other techniques to maximize the effectiveness of the overall strategy. Additionally, the report highlights the need to regularly update and test the machine learning techniques for SQL injection attack detection and prevention represents a promising approach to improving the security of web applications that use SQL databases. The development and use of effective machine learning techniques will become increasingly important for protecting against SQL injection attacks in the future.

IndexTerms **- SQL injection, Cyber security, Machine learning, Feature selection, Precision, Recall**

I. INTRODUCTION

The majority of the apps we use on a daily basis are web-based. Organizations choose to make their applications available via the Internet in order to receive more publicity. Being exposed to the Internet raises the security risks associated with different types of online transactions. All of the information submitted by users during these interactions on web apps or websites is saved in a database. Linked databases can be talked with using SQL, or Structured Query Language.

SQL injection is one of the most prevalent online application security flaws. It can be used to obtain illegal access to a database and alter data. In recent years, the use of machine learning techniques for SQL injection attack detection and prevention has been growing. Machine learning techniques enable the creation of models that can spot and avoid SQL injection attacks in real time by identifying patterns in data that indicate an attack. These models can also be used to detect and prevent suspicious queries before they even reach the database. Furthermore, machine learning techniques can be used to identify and react to malicious activities in an accurate and efficient manner.

Organizations can safeguard their databases from SQL injection threats by utilizing the power of machine learning. Using SQL to initiate attacks on databases and manipulate them to do what the user desires is a type of a web hacking method called SQL Injection Attack. SQL Injection Attacks are becoming a growing source of concern for computer fighters. SQL Injection and Remote Code Execution attacks alone accounted for more than four-fifths of all identified web-based assaults in the preceding year. SQL Injection assaults continue to be one of the most common types of cyber-attacks. Many methods have been created to combat such attacks, but cyber hackers continue to effectively circumvent the various security mechanisms in place to combat SQL Injection attacks.

Lately, the use of machine learning techniques to identify and avoid various cyber security risks is being discussed extensively. While the effectiveness of supervised and unsupervised learning techniques in detecting security threats cannot be disputed, the computing resources and time needed to execute such complicated algorithms remains a significant worry for the rapidly evolving cyber security

community. There has been a tremendous amount of study done on the use of different machine learning algorithms to identify SQL Injection attacks.

II. LITERATURE REVIEW

Kevin Ross made a research on "SQL Injection Detection Using Machine Learning Techniques and Multiple Data Sources". SQL Injection remains one of the most dangerous security vulnerabilities in terms of both personal information disclosure and monetary loss. Injection assaults are the most common vulnerability, according to the most current OWASP Top 10 survey, and the number of these attacks is growing. Traditional security strategies frequently employ static, signature-based IDS (Intrusion Detection System) principles that are mostly successful only against previously witnessed attacks and not against unknown, or zero-day, attacks. Much contemporary research includes the use of machine learning methods, which can identify new assaults but can be expensive in terms of speed based on the algorithm. Furthermore, most contemporary breach detection strategies gather traffic entering the web application from a network device or the web application host, whereas other strategies collect data from database server records. We are gathering traffic from two sources in this project: the web application host and a Datiphy appliance node positioned between the web application host and the related MySQL database server. We were able to demonstrate that accuracy obtained with the correlated dataset using algorithms such as rule-based and decision tree are nearly the same as those obtained with a neural network algorithm, but with significantly improved performance, in our analysis of these two datasets and another dataset that is correlated between the two.

Chen and Yeh (2019) proposed a machine learning approach to detect SQL injection attacks based on HTTP requests. Their approach involved using machine learning algorithms, such as Naive Bayes, Decision Tree, and Random Forest, to classify HTTP requests as either benign or malicious. The authors evaluated their approach using a dataset of HTTP requests, and found that their approach achieved high detection rates with low false positives. However, further research is needed to optimize the feature selection process and improve the accuracy of the approach. Overall, Chen and Yeh's approach shows promise in detecting SQL injection attacks using HTTP requests, and could be a valuable addition to existing web application security measures.

Alam et al. (2020) proposed a deep learning approach for SQL injection attack detection in web applications. Their approach involved using a convolutional neural network (CNN) to analyse the content of HTTP requests and detect potential SQL injection attacks. They evaluated their approach using a dataset of HTTP requests, and found that their approach achieved high accuracy rates with low false positives. Additionally, their approach was found to outperform traditional machine learning algorithms such as Support Vector Machines (SVM) and Random Forest. However, the authors noted that further research is needed to optimize the CNN architecture and improve the scalability of the approach. Overall, Alam et al.'s approach shows promise in detecting SQL injection attacks in web applications using deep learning techniques, and could be a valuable addition to existing web application security measures.

Hu et al. (2019) proposed a hybrid model for detecting SQL injection attacks in web applications using machine learning and rule-based methods**.** Their approach involved using a feature selection algorithm to extract important features from HTTP requests, and then combining machine learning algorithms such as Support Vector Machines (SVM) and decision trees with rule-based methods to detect SQL injection attacks.

Al-Qureshi and Ibrahim (2018) proposed a novel approach for detecting SQL injection attacks in web applications using machine learning algorithms. Their approach involved using a feature selection algorithm to select the most relevant features from HTTP requests, and then training various machine learning algorithms such as Naive Bayes, Decision Trees, and Support Vector Machines (SVM) on the selected features to detect SQL injection attacks.

III. METHODOLOGY

We focused on detecting SQL injection attacks using machine learning rather than traditional methods because artificial intelligence has a greater ability to obtain detection methods for these attacks than traditional approaches, which have limitations in this regard. The suggested model's main goal is to detect SQL Injection attacks, The technique is divided into four steps: The initial step involves gathering a dataset where we have collected a dataset containing SQL queries and SQL injection attack queries, The second stage involves data Pre-processing, in the third stage, feature extraction is performed for the data set, The model is trained in the fourth stage with four machine learning algorithms which are KNN,MNB, DT and SVM algorithms , In this phase, 80% of the dataset is used to train the model, and the fifth stage focuses on testing and evaluating the proposed model with 20% of dataset that we separated from the collected dataset. The methodology used in this study are:

1. Dataset

Gathering a useful dataset that contains both SQL injection attacks and normal SQL queries is the essential step in detecting a SQLIA. In our study, we used a SQL injection data collection and pre-processing methods to prepare the data for training. We then extracted certain features. After that, we used the machine learning algorithm to train the model with the training data and extract features. After that, some performance metrics are used. We decided to divide the dataset into two parts: training and testing. The training set contains 80% of the data and is used to train the models, while the remaining 20% is used to evaluate the trained models. We collected datasets from publicly available repositories. We used the Kaggle website to collect data on SQL injection attacks and benign traffic.

2. Data Preprocessing

The data should be pre-processed to ensure that it is in the correct format for machine learning models. This includes removing any irrelevant or redundant data, normalizing the data, and transforming it into a vector format.

3. Feature extraction

Feature extraction is the process of converting raw data into numerical characteristics that can be handled while retaining the information in the original data collection. It extracts the most important characteristics from the information for training the machine learning model. This involves choosing the most essential attributes, such as request headers, query strings, and user input.

4. Training Algorithms

For training, we'll utilize K-Nearest Neighbor algorithm, Multinomial Naive Bayes algorithm, Decision tree algorithm and Support Vector Machine algorithm, which are all machine learning algorithms.

4.1 K-Nearest Neighbors Algorithm

K-nearest neighbor is a supervised ML algorithm for classifying and predicting data, in classification cases, it assigns new data to a specific class based on its similarity to classified data in that class, which can be determined by computing the shortest distance between the updated data and the classified data. A common method for computing distance in KNN is to use the distance, which calculates a straight line between two points. To calculate the distance in KNN, use the following equation: The distance between points A(x1,y1) and B(x2,y2) is:

$$d(x,y) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

4.2 Multinomial Naïve Bayes Algorithm

MNB is a Naive Bayes variant developed to perform text document classification, MNB uses a multinomial distribution as a classification feature, with the number of words appearing or the word weight, MNB counts the number of times each word appears in the document.

4.3 Decision Tree Algorithm

Decision trees are a form of prediction model that has been used to address classification issues, a decision tree generates models that allow object categorization by generating a set of decision rules, These rules are derived from the attributes of the training data, Each child node in the tree, including its branches, represents a set of characteristics that contribute to decision tree classification, As a result, classifying an object will begin by looking at the root node's value, then progressing down the tree branches that correspond to those values, This is repeated for each node until it reaches the leaf node, at which point it can no longer go any further, While developing the decision tree for the best model, the Information Gain (IG) are utilized to choose the root node and sub-node.

4.4 Support Vector Machine Algorithm

SVM is a collection of supervisory learning algorithms based on statistical learning theory that can be utilized for classification and regression applications. SVM is a global model of classification that generates non-overlapping portions and employs all attributes in general as a classification system. SVM is based on maximum margin and linear discriminant, comparable to a probabilistic technique, but without taking into account inter quality. An SVM is a linear classifier, It is common practice to employ machine learning (ML) techniques to categorize SQL injection attacks, and one of the most prominent ML algorithms for classification.

5. Model Training

After the data has been prepared, the machine learning model can be trained using the selected features. This involves creating a model that is capable of detecting abnormal traffic patterns and flagging it as a potential SQL injection attack.

6. Model Validation

The model must be tested on a separate dataset to ensure it is performing accurately. This allows the model to be tweaked and improved, if necessary.

7. Model Deployment

Once the model has been tested and validated, it can be deployed in the production environment and used to detect and prevent SQL injection attacks.

By using machine learning methods for SQL injection attack detection and prevention, organizations can ensure that their databases are protected from malicious activities. Additionally, these methods can help reduce the time and effort associated with manual detection and prevention of SQL injection attacks.

IV. SYSTEM ANALYSIS

System analysis for SQL injection attack involves analysing the web application and identifying vulnerabilities that could be exploited by attackers to launch SQL injection attacks. The investigation can be split into the following steps:

1.Identify the Attack Surface: The first step in system analysis is to identify the components of the web application that could be targeted by SQL injection attacks. This could include the login page, search functionality, and other forms that accept user input.

2. Understand the Attack Vector : Once the attack surface has been identified, it is important to understand how attackers could exploit vulnerabilities in the web application to launch SQL injection attacks. This could involve analysing the source code of the application, examining the database schema, and studying the network traffic.

3. Identify Vulnerabilities: Based on the attack vector, identify the vulnerabilities in the web application that could be exploited by attackers to launch SQL injection attacks. Common vulnerabilities include lack of input validation, inadequate access controls, and weak password policies.

4. Assess Risk: Once the vulnerabilities have been identified, assess the risk associated with each vulnerability based on the likelihood of an attack and the potential impact on the web application and its users.

5. Develop Mitigation Strategies: Based on the risk assessment, develop mitigation strategies to address the vulnerabilities and reduce the risk of SQL injection attacks. This could involve implementing input validation techniques, restricting database privileges, and deploying a web application firewall.

6. Test the Mitigation Strategies: Test the effectiveness of the mitigation strategies by conducting penetration testing and vulnerability scanning to identify any remaining vulnerabilities that could be exploited by attackers.

7. Monitor the System: Once the mitigation strategies have been implemented, monitor the web application for any signs of SQL injection attacks using tools such as intrusion detection systems and log file analysis.

It is important to note that system analysis for SQL injection attack should be an ongoing process and should be conducted regularly to ensure that the web application remains secure against evolving attack patterns.

V. IMPLEMENTATION

5.1 DETECTION TECHNIQUES:

SQL injection attacks can be discovered using a variety of methods, some of which are listed below.

Input Validation: Input validation is a technique for ensuring that user input is safe and can be processed without causing any issues. By validating input, it becomes much harder for an attacker to inject malicious code into the application.

Error Messages: Error messages can be an indicator of a potential SQL injection attack. When an application returns an error message that contains database-related information, it can reveal details about the structure of the database or the application, which an attacker can use to launch a more effective SQL injection attack.

Log Analysis: Log analysis can help detect SQL injection attacks by analyzing the logs generated by the application or the database. Logs can provide information about the type of queries executed, the source of the queries, and any errors that occurred during query execution.

Database Monitoring: Database monitoring can help detect SQL injection attacks by monitoring the database for unusual activity, such as an increase in the number of queries or a sudden spike in the amount of data retrieved. Database monitoring can also be used to detect any attempts to access sensitive data.

Web Application Firewall: A web application firewall can detect and block SQL injection attacks in real-time. A web application firewall analyzes incoming traffic and blocks any traffic that matches predefined attack patterns. A WAF can also be used to block known SQL injection tools and techniques.

Machine Learning Techniques: Machine learning techniques can be used to detect SQL injection attacks in real-time. Machine learning algorithms can be trained on a dataset of normal and attack traffic to identify patterns that are indicative of SQL injection attacks.

Database Response Time: Database response time can be an indicator of a potential SQL injection attack. An attacker may execute a large number of queries in a short period of time, causing a noticeable slowdown in the database's response time.

5.2 PREVENTION TECHNIQUES:

SQL injection assaults can be avoided using a variety of methods, including:

Parameterized Queries: Parameterized queries are another effective technique for preventing SQL injection attacks. Parameterized queries use placeholders for user input, which are then replaced with sanitized input when the query is executed. This technique prevents attackers from injecting malicious code into the query.

Database Privileges: Database privileges should be restricted to prevent attackers from accessing sensitive data or executing malicious queries. This includes limiting access to specific tables or columns and using strong passwords for database accounts.

Web Application Firewall: A web application firewall can detect and block SQL injection attacks in real-time. A web application firewall analyzes incoming traffic and blocks any traffic that matches predefined attack patterns. A WAF can also be used to block known SQL injection tools and techniques.

Machine Learning Techniques: Machine learning techniques can be used to prevent SQL injection attacks in real-time. Machine learning algorithms can be trained on a dataset of normal and attack traffic to identify patterns that are indicative of SQL injection attacks. These patterns can then be used to identify and block potential SQL injection attacks before they can do any harm.

Regular Updates: Regularly updating the web application and its components, including the web server, database server, and third-party libraries, can ensure that any known vulnerabilities are patched.

Prepared Statements: Prepared statements are another technique that can be used to prevent SQL injection attacks. Prepared statements are similar to parameterized queries, but they are used for more complex queries.

Stored Procedures: Stored procedures are precompiled SQL statements that are stored in a database. By using stored procedures, the application can avoid sending SQL commands directly to the database, thereby reducing the risk of SQL injection attacks.

Limiting Error Messages: Error messages should be limited to avoid giving attackers information about the structure of the database or the application. If an attacker knows the structure of the database or the application, they can use this knowledge to launch a more effective SQL injection attack.

Regular Audits: Regular security audits can help identify and address any vulnerabilities that could be exploited by attackers to launch SQL injection attacks.

VI.RESULT ANALYSIS

The analysis of results obtained from using machine learning techniques for SQL injection attack detection and prevention can be done in various ways, some of which are:

True Positive Rate: The true positive rate is the percentage of actual SQL injection attacks that were correctly detected by the machine learning model. A higher true positive rate indicates that the model is effective at detecting SQL injection attack.

False Positive Rate: The false positive rate is the percentage of normal traffic that was incorrectly identified as a SQL injection attack by the machine learning model. A lower false positive rate indicates that the model is not generating too many false positives, which can be costly in terms of time and resources.

Accuracy: The accuracy of the machine learning model is the percentage of all traffic that was correctly classified as either normal traffic or SQL injection attacks. A higher accuracy indicates that the model is more effective at distinguishing between normal traffic and SQL injection attacks.

.

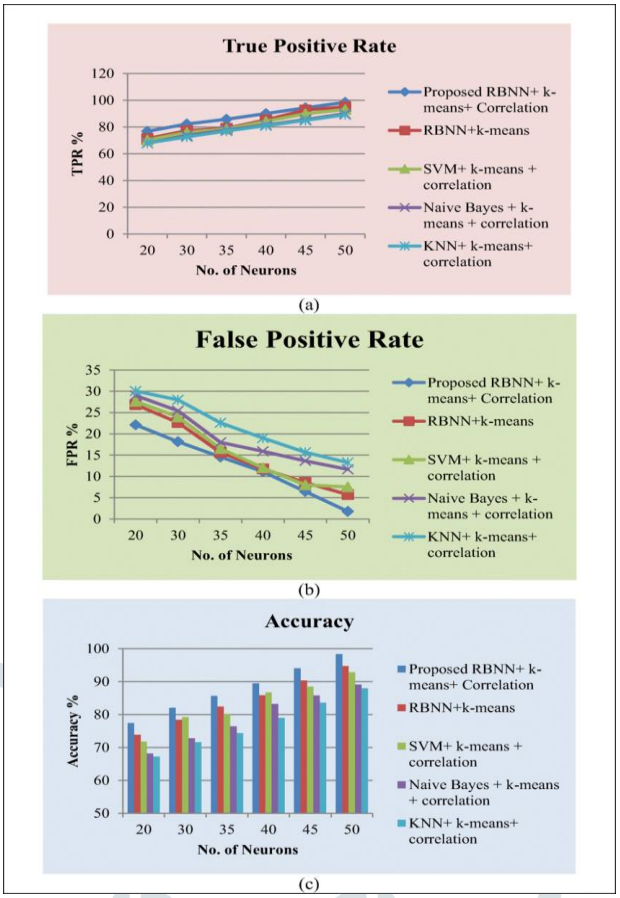

Figure 1:(a)True positive rate (b)False positive rate(c)Accuracy

Precision and Recall: Precision is the percentage of traffic identified as SQL injection attacks that were actually SQL injection attacks. Recall is the percentage of actual SQL injection attacks that were correctly identified by the model. A higher precision and recall indicate that the model is more effective at detecting SQL injection attacks.

F1 Score: The F1 score is the harmonic mean of precision and recall, and is a single metric that combines the performance of the model in terms of both precision and recall. A higher F1 score indicates that the model is more effective at detecting SQL injection attacks.

Training and Testing Time: The time taken to train the machine learning model and the time taken to test the model on new data can also be analyzed. A shorter training and testing time indicates that the model is efficient and scalable.

Robustness: The robustness of the machine learning model can be evaluated by testing it against different types of SQL injection attacks and different attack patterns. A more robust model can detect a wider range of SQL injection attacks.

The analysis of results obtained from using machine learning techniques for SQL injection attack detection and prevention can provide insights into the effectiveness, efficiency, and scalability of the model. This analysis can help improve the model's performance and make it more effective at detecting and preventing SQL injection attacks.

VII. CONCLUSION

Hence, SQL injection attacks remain a significant threat to web applications that use SQL databases. Traditional detection and prevention techniques have limitations and can be ineffective against evolving attack patterns. Machine learning techniques offer a promising solution to this problem. Machine learning techniques can be trained on a dataset of normal and attack traffic to identify patterns that are indicative of SQL injection attacks. These patterns can then be used to identify and block potential SQL injection attacks in real-time. Machine learning techniques can also adapt to new attack patterns and can be updated regularly to improve their performance. However, it's important to note that machine learning techniques should be used in combination with other techniques, such as input validation, parameterized queries, and database privileges, to maximize the effectiveness of the overall strategy. Additionally, machine learning techniques should be regularly updated and tested to ensure their effectiveness against evolving attack patterns. Overall, the use of machine learning techniques for SQL injection attack detection and prevention represents a promising approach to improving the security of web

applications that use SQL databases. As the threat landscape continues to evolve, the development and use of effective machine learning techniques will become increasingly important for protecting against SQL injection attacks.

VIII. REFERENCES

[1]. Ross, K. (2016). SQL injection detection using machine learning techniques and multiple data sources. Journal of Information Security, 7(4), 220-228

[2]. Chen, H., & Yeh, C. (2019). Detecting SQL injection attacks using machine learning techniques. International Journal of Innovative Computing, Information and Control, 15(1), 77-92.

[3]. Alam, M. M., Uddin, M. Z., & Bhuiyan, M. A. (2020). A deep learning approach for SQL injection attack detection in web applications. International Journal of Advanced Computer Science and Applications, 11(3), 115-122.

[4]. Hu, Y., Huang, Q., & Liu, Y. (2019). A hybrid model for SQL injection attacks detection. IEEE Access, 7, 55339-55346.

[5]. Al-Qershi, O. M., & Ibraheem, R. K. (2018). Feature selection using genetic algorithm for detecting SQL injection attacks. International Journal of Advanced Computer Science and Applications, 9(6), 238-246.

[6]. OWASP Top 10, 2013, "Top 10 2013-A1-Injection", https://www.owasp.org/index.php/Top_10_2013- A1-Injection

[7]. K. Ross, M. Moh, T-S. Moh, J. Yao, Poster: Multi-Source Data Analysis for SQL Injection Detection, 38th IEEE Symposium on Security and Privacy (IEEE S&P), San Jo-se, CA, 2017.

[8]. D. Kar, S. Panigrahi, "Prevention of SQL Injection attack using query transformation and hashing," in Advance Computing Conference (IACC), 2013 IEEE 3rd International, vol., no., pp.1317-1323, 22-23 Feb. 2013.

[9]. Alghawazi M, Alghazzawi D, Alarifi S. Detection of sql injection attack using machine learning techniques: a systematic literature review. J Cybersecur Privacy. 2022;2(4):764–77.

.