Survey on SQL Injection Attack Detection

Article *in* Strad Research · April 2024

CITATIONS
0

READS
261

4 authors, including:

Dipika D. Raigar
Pune Institute Of Computer Technology
10 PUBLICATIONS **1** CITATION

SEE PROFILE

Survey on SQL Injection Attack Detection

Aditya N. Deshpande
BE Student, Computer Engineering
PICT, Pune

Dikshant G. Borse
BE Student, Computer Engineering
PICT, Pune

Atharv C. Kulkarni
BE Student, Computer Engineering
PICT, Pune

Prof. Dipika D. Raigar
Assistant Professor, Computer Engineering
PICT, Pune

***Abstract*—SQL injection attacks are a significant threat to web applications which cause severe harm to organizations through financial losses and data breaches. In this kind of attack, malicious actors try to access databases and potentially leak valuable information. These attacks involve injecting malicious queries into a database, leading to data breaches and the exposure of sensitive user information. Hence, it is crucial to develop strategies for preventing SQL injection attacks.**
A study was conducted to examine relevant literature, interpret their findings, and seek to understand the approaches they took to detect and prevent SQL Injection. The aim was to gain insights into the tactics and methods proposed in these papers for mitigating the risks associated with SQL injection attacks.

***Index Terms*—SQL Injection Attacks, Vulnerabilities, Malicious Queries, Machine Learning Techniques, Detection of attacks, Database server**

INTRODUCTION

Software development always aims to create secure and trustworthy software. After gaining adequate experience, programmers understand why software needs to be secure while handling sensitive and confidential data. While designing online applications, secure programming should be given priority over functional validation. However, it is difficult to rewrite web applications to address vulnerabilities like injection attacks and hence need protection.

SQL injection is a commonly used web-hacking technique. In SQL injection, hackers place malicious code in SQL statements, which are generated via web page input. It involves the manipulation of SQL code, either to execute harmful commands or gain unauthorized access to protected resources. Hence, SQL injection is a significant problem that needs to be addressed. SQL queries, which are actions taken on the database, can be tampered with by attackers to get unauthorized access to the database. The process begins with understanding the operations performed by the database, which is done by altering the query's random values and observing the server's response.

Web servers communicate with database servers whenever they need to store or retrieve user data. Attackers try to design SQL statements to execute them while the web server is fetching some content from the application server. SQL injection exploits weaknesses in an application's user input handling. There are various forms of SQL injection attacks. In classic SQL injection, attackers inject harmful code into input fields, which leads to unauthorized access, data alteration, or data deletion. Due to delayed feedback received by the server, blind SQL injection is more complex as it requires the attackers to design custom queries to extract data. Other types include time-based blind SQL injection, union-based SQL injection, error-based SQL injection, and boolean-based blind SQL injection.

There are some scenarios where direct data extraction is not possible. In such cases, SQL injection allows attackers to retrieve data through other means. Stored user input, when exploited, can be used for injection attacks, leading to second-order SQL injection and other security risks. Time-based blind second-order SQL injection is slower than second-order injection for retrieving data. Double SQL injection may lead to unexpected risks as it alters user input in multiple queries. Stored procedure injection, which targets applications using stored procedures, can also compromise database security.

SQL injection can be prevented by user authentication. Validating input from the user by pre-defining the length and type of input in the input field and authenticating the user. We can restrict access privileges for users and define how much data any outsider can access from the database. In short, users should be given limited access to the database.

The importance of detecting and preventing SQL injection attacks is well understood by every organization. While designing an application, developers should take into consideration the security and authentication of that application. We have understood this importance by studying and analyzing 20 papers that highlight the serious security issues and methods for SQL Injection detection. In this paper, we have presented our understanding and literature review of these 20 papers.

LITERATURE REVIEW

Sr. No.	Paper Title	Author	Year	Worked On	Limitation/ Future Scope
1.	SDSIOT: An SQL Injection Attack Detection and Stage Identification Method Based on Outbound Traffic	Houlong Fu, Chun Guo, Chaohui Jiang, Yuan Ping, and Xiaodan Lv	2023	SDSIOT is a method for detecting SQL Injection Attacks (SQLIAs) and identifying their stages based on outbound traffic from the Web server. It analyzes the distinct characteristics of outbound traffic generated during different stages of SQLIA.	The limitation of the SDSIOT project is its dependency on outbound traffic for SQLIA detection and stage identification. This approach may not be effective in scenarios where outbound traffic is limited or not available. While SDSIOT can identify the two main stages of SQLIA (FIP and LD), it does not provide a detailed identification of sub-stages within these stages. The project suggests exploring convolutional neural networks (CNN) to improve the accuracy of SQLIA detection and stage identification.
2.	Detection of SQL Injection Attacks using Machine Learning in Cloud Computing Platform	Jamilah M Alkhathami and Sabah M. Alzahrani	2022	The research focuses on detecting SQL injection attacks in cloud computing platforms using machine learning algorithms. The study compares the performance of four machine learning models: K-Nearest Neighbors (KNN), Multinomial Naive Bayes (MNB), Decision Tree (DT), and Support Vector Machine (SVM).	The study focused on a specific dataset and did not explore the performance of the models on different datasets or in real-world cloud computing environments. In terms of future scope, the performance of other machine learning algorithms or ensemble methods for SQL injection detection could be explored. It could be extended to evaluate the models' performance in different cloud computing platforms and with larger and more diverse datasets.
3.	Long short-term memory on abstract syntax tree for SQL injection detection	Z. Zhuo, T. Cai, X. Zhang, and F. Lv.	2021	The authors propose a novel detection approach based on long short-term memory (LSTM) and abstract syntax tree (AST) for SQL injection detection. The approach uses deep learning techniques to capture both context and syntax information from raw SQL queries, resulting in superior performance.	The limitations of the paper include the reliance on raw SQL queries, which may not capture all possible attack scenarios, and the focus on a specific database (MySQL) for the SQL grammar parser. In terms of future scope, the authors suggest that their approach can be extended to other databases by implementing additional SQL grammar parsers. They also mention the potential for collaboration with intrusion detection systems (IDS) and web application firewalls (WAF) to enhance the detection capabilities.
4.	Ensemble Machine Learning Approaches for Detection of SQL Injection Attack	Umar Farooq	2021	The paper proposes an ensemble machine learning approach for detecting SQL Injection Attacks, using algorithms like LGBM and AdaBoost. The results show high accuracy and precision for the proposed model.	The proposed model relies on labeled data for training the classifier. This means that it may not be able to detect future attacks that were not present in the labeled dataset. The paper does not discuss the performance of the model on a real-world dataset or in a production environment. Researchers could investigate the application of other machine learning algorithms or ensemble techniques to improve the detection accuracy of SQL injection attacks.
5.	An Improved SQL Injection Attack Detection Model Using Machine Learning Techniques	Yazeed Abdulmalik	2021	The paper proposes a model that extracts semantic features from SQL statements to enhance the efficiency of SQLIA detection. The model consists of three phases: dataset creation, static and dynamic analysis, and model construction.	The limitation of this paper is that it does not provide an evaluation or validation of the proposed detection model. The paper suggests that the proposed model can be further improved by testing different machine learning algorithms such as Random Forest, Artificial Neural Network, Support Vector Machine, and Logistic Regression. It also mentions the need for extracting semantic features from SQL statements to enhance the detection of SQL injection attacks.

Sr. No.	Paper Title	Author	Year	Worked On	Limitation/ Future Scope
6.	Effective Filter for Common Injection Attacks in Online Web Applications	Santiago Ibarrafiallos, Javier Bermejo Higuera, Monserrate Intriago-pazmino, Juan Ram ˜ on Bermejo Higuera, Juan Antonio Sicilia Montalvo, Javier Cubo	2021	Web application filter using regular expressions and sanitization for injection attack prevention, focusing on refining and expanding the proposed filter.	Refining and expanding the proposed filter by improving regular expressions. Expanding the sanitization process, and incorporating additional rules to detect and prevent injection attacks. It also involves integrating the filter with other security measures, optimizing its performance, and evaluating it on a larger scale.
7.	SQLIA Detection and Prevention using Dynamic Parse Tree with Query Tokenization	Manikandan	2020	This paper proposes a technique by using a dynamic parse tree with query tokenization. The technique involves tokenizing the original query and the injected query to detect the presence of malicious code.	It focuses on preventing SQL Injection attacks using dynamic parse tree and query tokenization, but it does not address other types of web application vulnerabilities. The proposed technique may have some overhead in terms of processing and storage due to the tokenization process. In the future, one can explore the integration of other security measures, such as input validation and parameterized queries, which can provide a more comprehensive defense against web application vulnerabilities.
8.	SQL Injection Detection Using Machine Learning	Sonali Mishra	2019	Algorithms called Gradient Boosting Classifier (GBC) and Naïve Bayes (NB) are used from ensemble ML approaches to classify and detect SQLIAs. It also states that the GBC achieved an accuracy of 97.4% in detecting SQLIs, which is higher than the accuracy achieved by the NB approach ie. 92.8%.	The document briefly mentions feature extraction, but it does not provide specific details about the features used. Further research can focus on developing more effective feature extraction techniques specifically tailored for SQL Injection detection. Future work could explore more advanced optimization techniques, such as grid search or Bayesian optimization, to find the optimal hyperparameters for the models. The paper also suggests that the machine learning approach for detecting SQL Injections could be used in combination with other detection mechanisms.
9.	Research on SQL Injection Attack and Prevention Technology Based on Web	Limei Ma, Yijun Gao, Dongmei Zhao, Chen Zhao	2019	This document discusses SQLIAs and prevention tech. based on web applications. It explains the principle of SQL injection, and different types of injection attacks, and provides examples of SQL injection. The document also offers recommendations methods.	The limitation of this paper is that it primarily focuses on the principle, types, and prevention of SQL injection attacks without delving into more advanced techniques or specific case studies. The paper does not provide an in-depth analysis of the effectiveness of the prevention techniques mentioned. The authors could consider evaluating the effectiveness of different prevention techniques through experiments or simulations.
10.	A Novel Approach to Foil SQL Injection Attack at Login Phase	Shivraj Sharma	2018	A novel approach for SQL Injection Attack detection during login phase using Secure Hash Algorithm (SHA) based OTP generation process and	Enhancing the designed application to detect all possible SQL Injection attack queries. Additionally, the paper suggests exploring the use of a machine learning-based approach for SQL Injection prevention on the server side. This could potentially improve the detection and prevention of SQL Injection Attacks in web applications.

Pillai & Verma (2023)[11] proposed a model which used the Authentication technique which detects the legitimacy of the webpage using hyperlink attributes. It provides accurate information whether the webpage is genuine or not. The system proposed here consists of neural networks based on the Maximal Munch Algorithm to further detect XSS attacks. The results are analyzed using Matlab tool to calculate the accuracy, precision, and other performance metrics, and results are displayed in the visual format for more insightful conclusion.

Zheng & Shen (2021)[12] propose a detection method using association rule mining to identify patterns of malicious queries and construct feature vectors for classification. Their results demonstrate that the method achieves high accuracy in detecting malicious queries and preventing noise-exploitation attacks. The study highlights the balance between quality and privacy in the context of big data.

Zhang et al. (2019)[13] proposes a traffic-based framework which can accurately detect and prevent SQLIA traffic from the internet. The proposed DIAVA framework consists of two parts, the front end which collects network traffic related to SQLIAs, and the back end which identifies SQLIAs. It used Regex for vulnerability detection as well as analysis of leaked data. It uses three levels of Regular expressions, first for traffic collection to filter out SQL Injection traffic, second at the back-end while behavior analysis identifies successful SQL Injection, and third to extract leaked sensitive data. It also employs GPU for fast processing of the model.

Xie et al. (2019)[14] proposed a method of SQL injection detection based on Elastic-Pooling CNN (EP-CNN) for web applications. The method aims to effectively detect SQL injection attacks by outputting a fixed two-dimensional matrix without truncating data. The model uses an elastic pooling layer to create a matrix with a different number of rows but fixed columns, retaining all the information of the original query strings. The experimental results show that the EP-CNN model has high accuracy and can successfully detect SQL injection attacks.

Li et al. (2019)[15] propose an Adaptive Deep Forest-based method for detecting SQL injection attacks. The method improves the structure of deep forests by concatenating the raw feature vector with the average of previous outputs, addressing the degradation of raw features. It also introduces the AdaBoost algorithm to update feature weights based on the classification error rate. Experimental results demonstrate that the proposed method achieves better detection accuracy, lower computational cost, higher flexibility, and robustness. Overall, the proposed method shows promising results in detecting SQL injection attacks in complex network environments.

Huang et al. (2017)[16] discuss the importance of web application security and the prevalence of cybercrime on websites. It focuses on two major security threats, SQL injection and cross-site scripting (XSS) attacks, and explores various countermeasures to mitigate these risks. The paper introduces VulScan, a web vulnerability scanner that uses combinative evasion techniques to bypass filters and web application firewalls (WAFs), thereby improving web application security. The scanner automatically generates test data and significantly expands test coverage, revealing more vulnerabilities.

Som, Sinha & Kataria (2016)[17] proposes a working methodology for preventing SQL Injection Attacks (SQLIA) in web applications. The methodology involves two phases: front-end and back-end. In the front-end phase, new clients register by entering their login details, which are processed to generate a final hash code whereas in the backend phase, the input query is tokenized and encrypted using the AES encryption algorithm. The tokenized query is compared with the tokenized query stored in the server and if they match, the query is sent to the main database. Otherwise, it is rejected.

Alazab & Khresiat (2016)[18] focus on detecting and preventing SQL Injection attacks at runtime. Different layers in the system model architecture are discussed. The main objective of the author is to present a new model for the detection and prevention of SQL Injection attacks in web applications. The model discussed here is based on negative tainting and SQL syntax-aware methods and evaluation of this model is done through SQL penetration testing in web applications and database servers. The paper presents 4 algorithms to prevent SQLIAs, to create login tables in the database, to create Taints Table in the database, and to detect the SQLIA. The approach used in this paper gives several significant advantages over other mechanisms.

Singh (2016)[19] discusses modern SQL Injection attacks which are less known. Due to their complex nature, they require a considerable amount of time to understand. This survey paper discusses the prevention and detection techniques to prevent all types of attacks like deep blind SQLIA, compounded SQLIA, etc.

Antunes & Vieira (2011)[20] discuss the importance of defending against web application vulnerabilities and the need for a defense-in-depth approach. It highlights the common security flaws in web applications, such as injection attacks and cross-site scripting, and the risks they pose. The paper emphasizes the need for security measures throughout the software development life cycle, including input validation, hotspot protection, and output validation.

Relevant Datasets

In the paper [1] by Fu et al. (2023), the CSE-CIC-IDS dataset is used for the SQLI detection using outbound

traffic analysis. This dataset was released in 2017 and 2018 through the cooperation of the Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC) and includes PCAP traffic and CSV documents.

In the paper [11] by Pillai & Verma (2023), the KDD Dataset is used by which users and hackers are separated by detecting various attacks namely phishing, XSS, and SQL injection attacks. The KDD Cup dataset is a collection of data commonly used in machine learning and data mining research. The dataset contains a large amount of network data, specifically network intrusion detection data, which is used for building and testing intrusion detection systems.

Discussion

Zheng & Shen (2021)[12] suggest further enhancing the defensive measures against query attacks on anonymization mechanisms. This can involve exploring more advanced detection methods, such as machine learning algorithms, to improve the accuracy of identifying malicious queries. Additionally, the research can be extended to other anonymization mechanisms beyond Diffix, considering that many mechanisms are at risk of being attacked. Further investigation can also be conducted on optimizing the noise mechanism to minimize data distortion while ensuring data privacy.

Xie et al. (2019) suggest that the future scope of paper [14] lies in further studying the subdivisions of attack types and implementing a multi-classification model that is not limited to the identification of SQL injection attacks

Li et al. (2019)[15] suggest exploring different feature extraction methods, optimizing the hyper-parameters, and enhancing the model's ability to handle complex network environments with multiple types of attacks. Furthermore, exploring the integration of other advanced techniques, such as deep learning algorithms or ensemble methods, could potentially enhance the detection accuracy and robustness of the system.

Huang et al. (2017) highlight the need for secure implementation, defense mechanism deployment, and penetration testing as crucial components of web application security in [16]. Also aims to reduce the computational power and network bandwidth required by prioritizing test combinations with the best chances of discovering vulnerabilities. They also suggest exploring feedback- or statistics-based mechanisms to enhance the scanner's performance.

Antunes & Vieira (2011) suggest exploration of new methods to overcome the limitations of vulnerability detection tools and the adoption of disruptive approaches. Additionally, the paper [20] suggests the possibility of developing compilers that enforce secure coding practices and automatically fix existing security vulnerabilities.

Conclusion

This paper provides an in-depth analysis of SQL injection attacks on web applications and proposes various prevention strategies. It delves into the different types of SQL injection attacks, their vulnerabilities, and the potential damage they can inflict on targeted entities. The paper presents a variety of proposed solutions, including the use of machine learning algorithms, regular expressions, encryption, tokenization, and dynamic parsing to detect and thwart SQL injection attacks. It emphasizes the importance of implementing secure programming practices and the need for comprehensive developer education. Furthermore, the paper highlights the shortcomings of existing vulnerability detection tools and advocates for the investigation of novel approaches, such as machine learning algorithms and compilers, to enhance the security of web applications.

References

[1] Houlong Fu, Chun Guo, Chaohui Jiang, Yuan Ping, and Xiaodan Lv, "SDSIOT: An SQL Injection Attack Detection and Stage Identification Method Based on Outbound Traffic", Electronics 2023, 12, 2472, 2023.

[2] Jamilah M Alkhathami and Sabah M. Alzahrani, "Detection of SQL Injection Attacks using Machine Learning in Cloud Computing Platform", Journal of Theoretical and Applied Information Technology, Vol.100. No 15, ISSN: 1992-8645,15th August 2022.

[3] Z. Zhuo, T. Cai, X. Zhang, and F. Lv, "Long short-term memory on abstract syntax tree for SQL injection detection", IET Soft. 2021;15:188–197., 2021.

[4] Umar Farooq, "Ensemble Machine Learning Approaches for Detection of SQL Injection Attack", IC2ST ISSN 1848-5588, 2021.

[5] Yazeed Abdulmalik, "An Improved SQL Injection Attack Detection Model Using Machine Learning Techniques", International Journal of Innovative Computing 11(1) 53-57, 2021.

[6] Santiago Ibarra-fiallos, Javier Bermejo Higuera, Monserrate Intriago-pazmiño, Juan Ramón Bermejo Higuera, Juan Antonio Sicilia Montalvo, Javier Cubo, "Effective Filter for Common Injection Attacks in Online Web Applications", IEEE Access PP(99):1-1, January 2021.

[7] Manikandan, "SQLIA Detection and Prevention using Dynamic Parse Tree with Query Tokenization", e-ISSN:2395-0056, Volume 7, Issue 5, May 2020.

[8] Sonali Mishra, "SQL Injection Detection Using Machine Learning", San Jose State University ScholarWorks, May 2019.

[9] Limei Ma, Dongmei Zhao, Yijun Gao, Chen Zhao, "Research on SQL Injection Attack and Prevention Technology Based on Web", IEEE 2019 International Conference on Computer Network, Electronic and Automation (ICCNEA), September 2019.

[10] Shivraj Sharma, "A Novel Approach to Foil SQL Injection Attack at Login Phase", IJIRT, Volume 5 Issue 2, ISSN: 2349-6002, July 2018.

[11] Seema Pillai, Dr. Vijayant Verma, "A Novel Web Attack Detection Mechanism Using Maximal-Munch with Torrent Deep Network", IEEE Transactions on Cloud Computing, July 2023.

[12] Jianguo Zheng And Xinyu Shen, "Pattern Mining and Detection of Malicious SQL Queries on Anonymization Mechanism", IEEE Access (Volume 9), January 2021.

[13] Haifeng Gu, Jianning Zhang, Tian Liu, Ming Hu, Junlong Zhou, Tongquan Wei, Mingsong Chen, "DIAVA: A Traffic-Based Framework for Detection of SQL Injection Attacks and Vulnerability Analysis of Leaked Data", IEEE Transactions on Reliability (Volume 69, Issue 1, March 2020), July 2019.

[14] Xin Xie, Chunhui Ren, Yusheng Fu, Jie Xu, Jinhong Guo, "SQL Injection Detection for Web Applications Based on Elastic-Pooling CNN", IEEE Access (Volume 7), October 2019.

[15] Qi Li, Weishi Li, Junfeng Wang, Mingyu Cheng, "A SQL Injection Detection Method Based on Adaptive Deep Forest", IEEE Access (Volume 7), October 2019.

[16] Hsiu-Chuan Huang, Zhi-Kai Zhang, Hao-Wen Cheng, Shiuhpyng Winston Shieh, "Web Application Security: Threats, Countermeasures and Pitfalls", IEEE Journals & Magazines, Computer, Volume 50, Issue 6, June 2017.

[17] Subhranil Som, Sapna Sinha, Ritu Kataria, "Study on SQL Injection Attacks: Mode, Detection and Prevention", IJEAST Vol. 1, Issue 8, ISSN No. 2455-2143, July 2016.

[18] Ammar Alazab, Ansam Khresiat, “New Strategy for Mitigating of SQL Injection Attack”, International Journal of Computer Applications (0975 – 8887), Volume 154 – No.11, November 2016.
[19] Jai Puneet Singh, “Analysis of SQL Injection Detection Techniques”, Theoretical and Applied Informatics, Vol. 28 (2016), no. 1&2, pp. 37–55, DOI: 10.20904/281-2037, May 2016.
[20] Nuno Antunes and Marco Vieira, “Defending against Web Application Vulnerabilities”, Journals & Magazines, Computer, Volume 45, Issue 2, August 2011.