Securing Web Applications Against SQL Injection: A Survey on Machine Learning Approaches

[1]Bhanu Pratap Singh, [2]Prof. Manish Kumar Singhal
[1]M.tech Scholar, [2]Associate Professor & H.O.D
[1,2]Department of Information Technology (IT)
[1,2] NRI Institute Of Information Science And Technology, Bhopal (Mp), India,
[1] kbpsrmail@gmail.com [2]manishsinghal.nirt@gmail.com

---***---

Abstract -In this survey paper discuss the SQL injection (SQLi) is one of the most critical and prevalent security vulnerabilities affecting web applications. It occurs when attackers exploit user inputs to inject malicious SQL queries, compromising databases and leading to data breaches, unauthorized access, or even complete control over the system. This survey paper explores the Traditional defense mechanisms, such as input validation and parameterized queries, while effective, often fall short in preventing sophisticated attacks. In recent years, machine learning (ML) has emerged as a promising approach to enhance web application security by detecting and mitigating SQLi threats. This survey explores the state-of-the-art machine learning techniques for securing web applications against SQL injection attacks. We review a wide range of ML-based methods, including supervised, unsupervised, and hybrid approaches. The survey highlights key algorithms such as decision trees, support vector machines (SVM), artificial neural networks (ANN), and deep learning models, emphasizing their effectiveness in identifying SQLi patterns. Additionally, we examine real-time detection systems, anomaly-based solutions, and the challenges associated with implementing ML in real-world web security, such as data scarcity and false positives.

Key Words: SQL Injection, Cross Side Scripting, Denial of Service Attack, Naïve Bias, Gradient Boosting, etc.

1. INTRODUCTION

Web attacks such as SQL Injection, although they have been around for decades, continue to be a relevant and increasingly damaging cause of exposure of personal data as well as negative financial impact to business and governmental entities [2]. This is true, in particular, as old attacks are modified and evolved, and new attack vectors continue to appear. Industry and security firms devote a great deal of resources to mitigation of web attacks, and many current mitigation strategies have limitations that current research is continually striving to overcome.

Much traditional web attack mitigation is done by static analysis of incoming web traffic, also known as signature detection. This strategy involves the creation of a signature characteristic of the web attack and then when this signature is detected, the suspicious traffic can be blocked by a firewall or other security appliance. This method has the benefit of being quick and can be implemented in real time to protect network resources, but one drawback is that only known attacks can be detected.

Another strategy for web attack mitigation specific to SQL injection is to focus on the structure of incoming SQL queries, and if a malformed query is detected, this is considered to be an SQL injection attack. This method has good detection results, and can also detect new attacks that involve malformed queries, but a drawback is that it requires significant knowledge of the application and the structure of what are considered "normal" queries.

An SQL injection detection strategy that is a current topic of research involves the use of machine learning techniques. Popular techniques in this research are decision trees, rule-based learning techniques, support vector machines (SVM), and neural networks. A primary advantage of these techniques is that they are capable of detecting new attacks. A potential drawback with these techniques, however, is the possibility of increased processing time depending on the algorithm used.

Research into the SQL detection techniques mentioned and others rely on the availability of good data. Much current research uses web traffic captured coming in to the web application, or uses logs from the web application and/or web server [3]. The strategy that we are proposing uses traffic captured inbound to the web application in combination with traffic captured between the web application and the associated database server at a Datiphy appliance network node. We are using traffic captured at these two points to create our datasets, and we then create a third dataset by correlating events between the datasets derived from the two capture points.

Securing web applications is a critical concern in today's digital age, where cyber threats are constantly evolving. One of the most common and dangerous forms of attack is SQL Injection (SQLi), which targets the database layer of web applications by exploiting vulnerabilities in SQL queries. Traditional security measures, such as input validation and parameterized queries, have been effective but are often limited in their ability to adapt to new attack patterns. In this context, machine learning models offer a promising solution for enhancing security against SQL injection. By analyzing patterns in web traffic and identifying anomalies that may indicate malicious activity, machine learning algorithms can detect and prevent SQLi attacks with greater accuracy and efficiency. This introduction explores how incorporating machine learning into web application security frameworks can mitigate SQL injection risks, offering a proactive and intelligent defense mechanism for safeguarding sensitive data

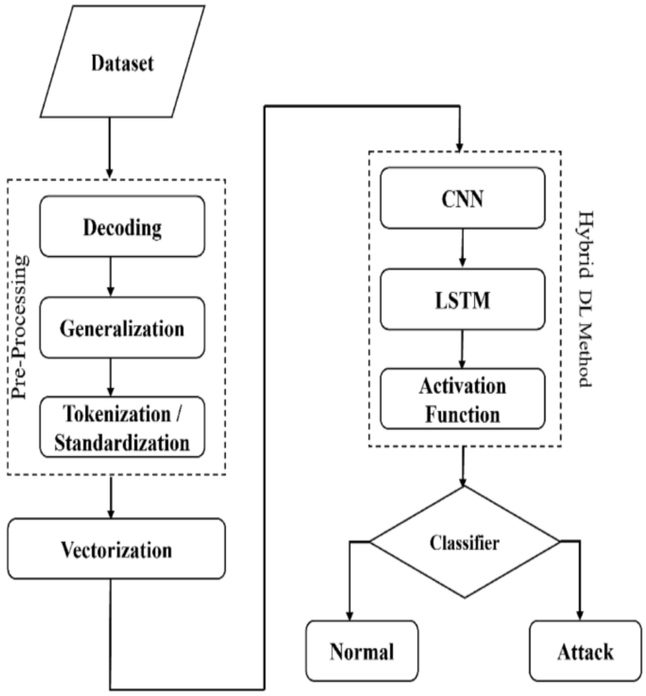

Fig 1 Securing Web Applications

2. LITERATURE SURVEY

S. Venkatramulu, et.al (2024), Author are presented The security in online applications cannot be guaranteed. Due to their accessibility, they are vulnerable to several flaws, and if these flaws are not fixed, they could have negative effects. One attack type that is simple to execute but difficult to detect is SQL Injection. This could lead to theft, the disclosure of private information, or the loss of property. This research effort has produced a novel method for detecting SQLi attacks utilizing word encoding techniques and machine learning algorithms. Our dataset includes legitimate, fraudulent SQL queries and plain text. We suggested a reliable methodology for differentiating plain text and regular queries from SQL injection attack queries. After evaluating the results, XGBoost algorithm with a unigram count vectorizer encoding of 70:30 split data ratio, gave us the best model with an F1-score of 0.992 and accuracy of 0.994. It has greater runtime as compared with other machine learning algorithms used. As numerous simple classifiers are used, ensemble learning techniques are said to produce results with higher accuracy [01].

Michael S. Souza ,et.al,(2024), Author are These services employ relational databases to store the collected data, thereby making them vulnerable to potential threats, including SQL Injection (SQLi) attacks. Hence, there is a demand for security solutions that improve detection efficiency and satisfy the response time and scalability requirements of this detection process. Based on this existing demand, this article proposes an SQLi detection solution that combines Regular Expressions (RegEx) and Machine Learning (ML), called Two Layer approach of SQLi Detection (2LDSQLi). The RegEx acts as a first layer of filtering for protection against SQLi inputs, improving the response time of 2LD-SQLi through RegEx filtering. From this filtering, it is analyzed by an ML model to detect SQLi, increasing the accuracy. Experiments, using a real dataset, suggest that 2LD-SQLi is suitable for detecting SQLi while meeting the efficiency and scalability issues [02].

Animesh Kumar, et.al, (2023), Author are study a From hosting websites to developing platforms and storing resources, cloud computing has tremendous use in the modern information technology industry. Although an emerging technique, it has many security challenges. In structured query language injection attacks, the attacker modifies some parts of the user query to still sensitive user information. This type of attack is challenging to detect and prevent. In this article, we have reviewed 65 research articles that address the issue of its prevention and detection in cloud and Traditional Networks, of which 11 research articles are related to general cloud attacks, and the rest of the 54 research articles are specifically on web security. Our result shows that Random Forest has an accuracy of 99.8% and a Precision rate of 99.9%, and the worst-performing model is Multi-Layer Perceptron (MLP) in the SQLIA Model. For recall value, Random Forest performs best while TensorFlow Linear Classifier performs worst. F1 score is best in Random Forest, while MLP is the most diminutive performer [03].

Babu R. Dawadi et.al, (2023), Using LSTM as our deep learning approach, the proposed model detected DDoS, XSS, and SQL injection attacks with considerably good accuracy. The first detection layer was a DDoS attack detection model with an accuracy of 97.57%, and the second layer was for XSS and SQL injection attack detection with an accuracy of 89.34%. We analyzed features and parameters for attack detection, which reduced false positives during traffic filtering in the WAF. As DDoS traffic comes at a higher rate than normal traffic, the system’s performance imporves when we check the traffic in a layered format, i.e., first checking for DDoS before testing for SQL injection and XSS. Moreover, we analyzed the performance perspective of the web application when an extra layer of filtering was added and found a slight impact on performance [04].

Manar Hasan Ali AL-Maliki, et. al (2022), Authors presented SQL injection attacks and the risks of these attacks on web pages and applications. The other objective is to know the latest studies on the solutions of SQL injection attacks and ways to address them to avoid exposure to this type of attack and provide a safe environment for use on the Internet. The database’s SQL queries are discussed to distinguish between malicious and normal. SQL injection is the most popular method hackers use to get sensitive data and information from users. Excessive privilege abuse, justified privilege abuse, privilege elevation, and platform vulnerabilities are all examples of database dangers, as well as methods to deal with them. Previous studies have identified ways to detect SQL injection attacks that give more accuracy and less time to detect maliciously; SQL queries are also covered [05].

Ahmed Abadulla Ashlamr, et.al, (2022), Author are study Structured Query Language (SQL) Injection constitutes a most challenging type of cyber-attack on the security of databases. SQLI attacks provide opportunities by malicious actors to exploit the data, particularly client personal data. To counter these attacks security measures need to be deployed at all layers, namely application layer, network layer, and database layer; otherwise, the database remains vulnerable to attacks at all levels. Research studies have demonstrated that lack of input validation, incorrect use of dynamic SQL, and inconsistent error handling have continued to expose databased to SQLI attacks

The security measures commonly deployed presently, being mostly focused on the network layer only, still leave the program code and the database at risk despite well-established approaches such as web server requests filtering, network firewalls and database access control [06].

Bronjon Gogoi, et.al, (2021), web applications for the delivery of services over the Internet. The risks to web applications have increased as their use has risen. SQL Injection Attack is a commonly exploited vulnerability used for stealing credentials, destroying and compromising data, and bypassing authentication and authorization controls of a web application. Traditional methods of detecting SQL injection attacks include software and hardware-based Web Application Firewalls, programmatic defense techniques like input filtering, input validation, using parameterized queries etc. and static and dynamic analysis are not sufficient for detection and prevention of SQLIA in web applications. In this paper, we present an approach to detecting SQLIA using NLP and Machine Learning [07].

Luca Demetrio et.al (2020) – Web Application Firewalls are widely used in production environments to mitigate security threats like SQL injections. Many industrial products rely on signature-based techniques, but machine learning approaches are becoming more and more popular. The main goal of an adversary is to craft semantically malicious payloads to bypass the syntactic analysis performed by a WAF. In this paper, we present WAF-A-MoLE, a tool that models the presence of an adversary. This tool leverages on a set of mutation operators that alter the syntax of a payload without affecting the original semantics. We evaluate the performance of the tool against existing WAFs, that we trained using our publicly available SQL query dataset. We show that WAF-A-MoLE bypasses the entire considered machine learning based WAFs [08].

3. ATTACK GENERATION

Gathering data for SQL injection research is generally done in two primary ways: capturing actual web traffic coming into an organization or honeypot, or the generation of realistic simulated traffic. Both approaches have their advantages and disadvantages. Real web traffic is of course the most realistic, but it can be difficult to determine which packets belong to an attack. It can also be difficult to obtain this type of traffic, as organizations are typically reluctant to share web traffic due to privacy and security concerns. Another issue is that a simple research honeypot may capture mostly automated scans generated by common attack tools that would be more easily captured in a controlled lab setting. Simulated attacks have the advantage that they are controlled so that normal and malicious traffic is easily distinguished for labeling purposes, and a good assortment of attacks can be included based on the latest techniques. 12 There are many tools in use by researchers in an effort to generate realistic attack traffic to test proposed detection and mitigation strategies. As mentioned, Lee et. al. [8] use the attack simulation tool Paros [29]. The most commonly used attack tool as determined by searches on Google Scholar is SQLMap [28]. Pinzón et al. [14] among others are using SQLMap to generate malicious traffic. Kar and Panigrahi [6] discuss and have tested several examples, including SQLMap and manually coded attacks. Many researchers such as Moh et al. [3], as well as our previous research [4] and the current project, are using manually coded SQL injection attacks to generate attack traffic

4. CONCLUSIONS

In this survey paper discuss on securing web applications against SQL injection attacks is critical due to the widespread use of databases in modern applications. Machine learning approaches have emerged as effective tools for detecting and mitigating SQL injection threats. This survey highlights the potential of various machine learning models, such as supervised, unsupervised, and hybrid techniques, to enhance traditional security mechanisms. Despite their advantages, challenges such as dataset quality, false positives, and adaptability to evolving threats persist. Future research should focus on refining these models to improve accuracy, scalability, and real-time detection, ensuring robust defenses for web applications.

REFERENCES

1. 1. S. Venkatramulu1 , Md. Sharfuddin Waseem2,* , Arshiya Taneem3 , Sri Yashaswini Thoutam4 , Snigdha Apuri5 and Nachiketh. " Research on SQL Injection Attacks using Word Embedding Techniques and Machine Learning Vol.02(01), Mar 2024, pp.55-66.
2. Michael S. Souza, Silvio E. S. B. Ribeiro, Vanessa C. Lima, Francisco J. Cardoso and Rafael L. Gomes. " Combining Regular Expressions and Machine Learning for SQL Injection Detection in Urban Computing." VOL. 15 NO. 1 (2024)
3. Animesh Kumar, Sandip Dutta, Prashant Pranav. "Analysis of SQL injection attacks in the cloud and in WEB applications" 18 January 2024, **https://doi.org/10.1002/spy2.370**
4. Babu R. Dawadi, Bibek Adhikari and Devesh Kumar Srivastava. " Deep Learning Technique-Enabled Web Application Firewall for the Detection of Web Attacks." Volume 23, Issue 4, 12 February 2023.
5. Manar Hasan Ali AL-Malikia,∗ , Mahdi Nsaif Jasim. " SQL injection attacks: Detection, to enhance the security of the website from client-side attacks." Appl. 13 (2022) 1, 3773-3782 ISSN: 2008-6822.
6. Ahmed Abadulla Ashlam, Atta Badii, Frederic Stahl. " Multi-Phase Algorithmic Framework to Prevent SQL Injection Attacks using Improved Machine learning and Deep learning to Enhance Database security in Real-time." November 2022, DOI: http://dx.doi.org/10.1109/SIN56466.2022.9970504.
7. Bronjon Gogoi; Tasiruddin Ahmed; Arabinda Dutta "Defending against SQL Injection Attacks in Web Applications using Machine Learning and Natural Language Processing**"** 01 February 2022, 10.1109/INDICON52576.2021.9691740.
8. Luca Demetrio, Andrea Valenza, Gabriele Costa, and Giovanni Lagorio. " WAF-A-MoLE: Evading Web Application Firewalls through Adversarial Machine Learning." 7 Jan 2020.
9. Mark A Aizerman. 1964. Theoretical foundations of the potential function method in pattern recognition learning. Automation and remote control 25 (1964), 821–837.
10. Hyrum S Anderson, Anant Kharkar, Bobby Filar, and Phil Roth. 2017. Evading machine learning malware detection. Black Hat (2017).
11. Dennis Appelt, Cu D Nguyen, and Lionel Briand. 2015. Behind an application firewall, are we safe from sql injection attacks?. In 2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST). IEEE, 1–10.
12. Dennis Appelt, Cu D Nguyen, Annibale Panichella, and Lionel C Briand. 2018. A machine-learning-driven evolutionary approach for testing web application firewalls. IEEE Transactions on Reliability 67, 3 (2018), 733–757.
13. Sruthi Bandhakavi, Prithvi Bisht, P Madhusudan, and VN Venkatakrishnan. 2007. CANDID: preventing sql injection attacks using dynamic candidate evaluations. In Proceedings of the 14th ACM conference on Computer and communications security. ACM, 12–24.
14. Marco Barreno, Blaine Nelson, Russell Sears, Anthony D Joseph, and J Doug Tygar. 2006. Can machine learning be secure?. In Proceedings of the 2006 ACM Symposium on Information, computer and communications security. ACM, 16–25.
15. Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. 2013. Evasion attacks against machine learning at test time. In Joint European conference on machine learning and knowledge discovery in databases. Springer, 387–402.
16. Battista Biggio and Fabio Roli. 2018. Wild patterns: Ten years after the rise of adversarial machine learning. Pattern Recognition 84 (2018), 317–331.
17. Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. 2010. The balanced accuracy and its posterior distribution. In 2010 20th International Conference on Pattern Recognition. IEEE, 3121–3124.
18. Nicholas Carlini and David Wagner. 2017. Adversarial examples are not easily detected: Bypassing ten detection methods. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. ACM, 3–14.
19. Mariano Ceccato, Cu D Nguyen, Dennis Appelt, and Lionel C Briand. 2016. SOFIA: an automated security oracle for black-box testing of SQL-injection vulnerabilities. In Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering. ACM, 167–177.
20. Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014).
21. Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. Machine learning 20, 3 (1995), 273–297.
22. Luca Demetrio, Battista Biggio, Giovanni Lagorio, Fabio Roli, and Alessandro Armando. 2019. Explaining Vulnerabilities of Deep Learning to Adversarial Malware Binaries. arXiv preprint arXiv:1901.03583 (2019).
23. Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli. 2018. On the Intriguing Connections of Regularization, Input Gradients and Transferability of Evasion and Poisoning Attacks. arXiv preprint arXiv:1809.02861 (2018).
24. Jeremy D'Hoinne, Adam Hils, and Claudio Neiva. 2017. Magic Quadrant for Web Application Firewalls. Technical Report. Gartner, Inc.
25. Parul Garg. [n.d.]. Fuzzing – Mutation vs. Generation. https://resources. infosecinstitute.com/fuzzing-mutation-vs-generation/. [Online; accessed 29- June-2019].
26. Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In International Conference on Learning Representations. http://arxiv.org/abs/1412.6572
27. Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. 2017. On the (statistical) detection of adversarial examples. arXiv preprint arXiv:1702.06280 (2017).
28. William GJ Halfond and Alessandro Orso. 2005. AMNESIA: analysis and monitoring for NEutralizing SQL-injection attacks. In Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering. ACM, 174–183.
29. Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. 2008. Kernel methods in machine learning. The annals of statistics (2008), 1171–1220.
30. Ling Huang, Anthony D Joseph, Blaine Nelson, Benjamin IP Rubinstein, and JD Tygar. 2011. Adversarial machine learning. In Proceedings of the 4th ACM workshop on Security and artificial intelligence. ACM, 43–58.