Multi-Phase Algorithmic Framework to Prevent SQL Injection Attacks using Improved Machine learning and Deep learning to Enhance Database security in Real-time

Ahmed Abadulla Ashlam
Department of Computer Science
University of Reading
Reading, UK
a.ashlam@pgr.reading.ac.uk

Atta Badii
Department of Computer Science
University of Reading
Reading, UK
atta.badii@reading.ac.uk

Frederic Stahl
German Research Center for Artificial Intelligence GmbH (DFKI),
NiedersachsenOldenburg,26129 Germany
frederic_theodor.stahl@dfki.de

2022 15th International Conference on Security of Information and Networks (SIN) | 978-1-6654-5465-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/SIN56466.2022.9970504

***Abstract*—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.

To overcome this deficiency, a Multi-Phase algorithmic framework is proposed with improved parameterised machine learning and deep learning to enhance database security in real-time at the database layer. The proposed method has been tested within a university and also in one of the branches of a commercial bank. The results show that the proposed method is able to i) prevent SQLi; ii) classify the type of attack during the detection process, and therefore iii) secure the database.

Keywords— SQL Injection, Machine Learning, Deep Learning, Real-time, Database Layer, algorithmic

I. INTRODUCTION

Over the last decade the number of internet users, globally, has maintained a steadily increasing trajectory, for example, rising from 4.6 billion in 2020 to 4.9 billion in 2021 [1].

Online transactions supported by data-driven applications and web services have facilitated the significant increase in new users connecting to the internet. The latest Open Web Application Survey Project (OWASP) indicates SQLI as posing the highest risks relative to all other cyber-attacks in the context of the top ten OWASP vulnerabilities [2]. Innovation in digitization of services and their online delivery through service-oriented architectures has resulted in the expansion of attack surfaces and the exposure to new vulnerabilities posing greater threats if the resulting security risks were not to be fully addressed [3].

The increasing number of users connected to the internet provides massive opportunities for malicious actors to deploy SQLI attacks on a mass-scale for stealing data from databases and thus violating citizens' and business rights to privacy and confidentiality and harming the economy [4].

Compromised web authentication, attacks on a domain name server, remote code execution and SQLi have targeted critical vulnerabilities in the application layer [5, 6].

However, presently most of the proposed solutions could only detect a subset of the SQLi attacks and have not been able to deal with the evolving techniques deployed in such attacks.

Over the last decade, researchers have proposed many methods for detection of SQLi; these can b e categorised under three types of approaches as follows: i) traditional algorithms ii) classical Machine Learning (ML) models, iii) deep neural networks. However, traditional approaches have proved inadequate particularly due to the deployment of auto-injection attacks. As for classical ML approaches, these still suffer from shortcomings arising from sub-optimal feature extraction and overfitting [7]. As a result, most studies have concluded that deep learning which is capable of automated feature extraction is required to cope with the many variants of SQLI attacks.

This paper reports on the results of our research to build a new comprehensive method for addressing the SQLi threats, with improved parameterised machine learning and deep learning to enhance database security in real-time. By analysing a large corpus of SQLi data, relevant features are extracted, then several machine learning models, and neural networks are trained using a large amount of actual data.

To review the current methods deployed for prevention of SQLi, the shortcomings in the state-of-the-art and the responsive research challenges arising from the increasing trend in SQLi attacks, section III sets out the Methodology deployed for detection and prevention of SQLi, section IV presents the Results and analysis, section V concludes the study and provides suggestions for future work.

A. SQL injection

SQL is a programming language for database access. It can be exploited to introduce malicious code into a website and transfer it to a database to execute different commands [8, 9]. Once a malicious actor has performed a SQLi scenario to run an illegal command, an attack has already happened [10, 11]. The attacker can thus proceed to further exploit vulnerabilities and take over a victim's database. Many websites that collect user data transfer it to SQL databases, which are intended to

execute any SQL commands. Figure 1 shows the SQLi attack scenarios.

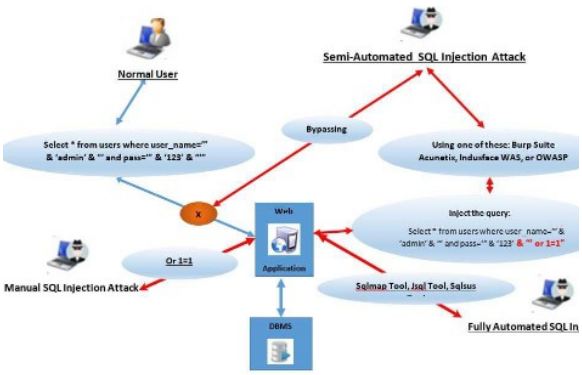

Figure 1. SQL injection attack scenarios

II. LITERATURE REVIEW

The rising trend of SQLI attacks and concerns regarding web security and risks to personal data have resulted in some machine learning techniques being deployed by research focused on various strategies to counter SQLi attacks. As a result, some steps can be taken to enhance the security for web services and databases at all the layers involved (the server, network and database and the application layer). However, the majority of organisations lack the security maturity stage to be able to respond to the required security steps beyond some basic measures that are inadequate for safeguarding their system given the continuously evolving and auto-injectable nature of SQLi attacks.

Web developers and security experts could adopt a secure coding approach as established by Bhawana Gautamet al. to protect their apps from such attacks during the coding process. A comparison of the proposed technique with existing preventative measures as well as a number of real-time PHP-based web apps have been carried out to arrive at enhanced accuracy and effectiveness of the selected approach [12].

Uwagbole, S et al. suggested a machine learning classification approach to identify and remedy SQLi vulnerabilities through a comparison between the retrieved database and known attacks published in the literature. The signatures list was next evaluated by the proposed classification approach to validate the token-based phase results. The support vector machine (SVM) approach was deployed by the categorisation system to block unauthorised user requests thus preventing them from accessing the back-end database of the targeted system. The suggested approach had the merit of being applicable to a large dataset, but its applicability to a wide range of SQLi attacks was limited [13].

A multilayer approach based on an intrusion detection system for multi-mode attacks was developed by Ajit Patil et al. Their analysis sets out a variety of attacks including SQLi and proposes solutions for them. They have put forward a Network Intrusion Detection and Prevention framework, which aims to detect and block coordinated attacks. The viability of the system is validated by systematic performance efficacy studies to convincingly verify that the recommended approach can indeed reduce the risk of successful attacks by internal and external adversaries [14].

For the purpose of identifying SQLi attacks, Joshi et al. adopted a hybrid solution integrating a Nave Bayes machine learning method with a role-based access control mechanism. They validated the model using SQLi attacks for annotations, summation, and paraphrasing [15].

Kamtuo and Soomlek deployed a decision tree-based SQLi protection system as well as training a machine learning model using 1,100 datasets of vulnerabilities [16]. Accordingly deep neural networks, also known as deep learning (DL), have been proposed in attempting to overcome the above-mentioned limitation of classical Machine learning approaches. For example, Selvaganapathy et al. studied the performance of deep confidence networks to extract URL elements, and distinguish between legitimate and non-legitimate URLs [17].

Sirenam et al. deployed a Convolutional Neural Network (CNN) demonstration site and launched fingerprint attacks to demonstrate that the accuracy in detecting CNN fingerprint assaults was above 98% [18].

The performance of the CNN classifier was further studied by S.S. Anandha Krishnan et al who concluded that compared to the other classifiers, CNN was capable of having the highest accuracy in identifying SQLi. They used a GitHub dataset to carry out their experiments resulting in the finding that, across a range of criteria, the CNN outperformed all other algorithms with an accuracy of 97% [19].

III. METHODOLOGY (DETECTION AND PREVENTION TECHNIQUE)

With the aim of developing a system that can detect and prevent various forms of SQLi attacks, the patterns of each

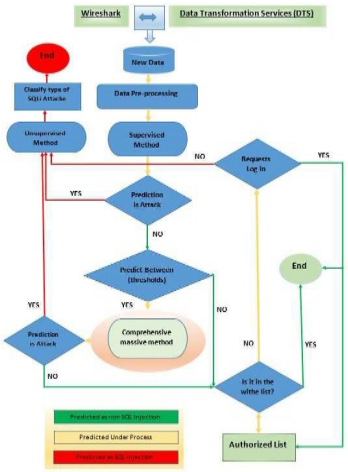

Figure 2. Workflow of the Multi-Phase algorithm

attack were studied, and solutions were designed responsive to these patterns. Figure 2 shows the workflow of the proposed method that has been implemented and tested in two application domains, namely i) a University IT Network, ii) a Commercial Bank.

A. Dataset

Creating a comprehensive dataset containing all conceivable payload and query types which can be used in SQLi attacks was one of the main goals of the project as this

would help train the model so as to enhance the security of the system against all kinds of queries used by attackers.

A comprehensive dataset containing a set of all possible SQLi attack payloads was generated by aggregating payloads for each type of SQLi such as generic, blinded, error-based and union-based SQLi. Queries for various databases such as MySQL, Oracle, MS-SQL, and Postgres were also included in the dataset. The data was sourced from various open-source libraries such as the National Vulnerability Database (NVD), GitHub, Kaggle and a python library named Lib-Injection. These versions have been added along with modified versions of these statements that will eventually help create a resilient system that detects malicious user-specified queries as well.

B. Pre-processing

Congested text documents which included payload queries were initially divided into separate sets of queries depending on the type of each query. The resulting sets were then ranked according to the injection type, with a tag of "1" for those which carried a harmful payload , or a tag of "0" if the payload was harmless. All payloads were then organised into a data frame comprising columns for the payload query and corresponding injection type as well as a column with values set to 1 or 0 to denote whether or not the payload had been classified as malicious or harmless. Thus, the resulting dataset was viewable, clean and usable for reading and understanding of the algorithm.

C. Feature engineering

Feature engineering includes the techniques that data scientists use to organise data in ways that make it more efficient to train data models and run inferences based on them. These technologies include the following: feature scaling or normalisation, data reduction, discretisation, and finally feature encoding.

Unstructured data should be converted into a structured representation as part of feature engineering. The Formats of Unstructured data can include text, audio, and video. For example, the development of natural language processing algorithms usually begins with data conversion algorithms such as Word2vec, N-Grams, CountVectorizer, the Hashing Vectoriser transformers, and several other methods to translate words into numerical vectors. The list of SQL query-encoding techniques used in the data processing pipeline for this research study.

B. Classification Model

Having considered the best performing machine learning models for classification, a number of the reportedly best performing classifiers were chosen and the best performing classifier was selected after the performance of each classifier had been enhanced and benchmarked against a set or metrics, namely F1-score ((2)* (Precision) * (Recall)) / (Precision + Recall).

E. Clustering Model

Clustering algorithms come in many different varieties. Several algorithms were used for clustering analysis of the pattern space distribution and to identify the data instances that constitute the core and the outlier type of SQLi attacks in the dataset. The unsupervised algorithm used here is K-Means, Agglomerative, Clustering, Density-Based Spatial Clustering of Noise, and Gaussian Mixture Model. The best performing algorithm was selected (K-Means) for integration in the Multi-Phase algorithmic framework to Prevent SQLi Attacks.

F. Multi-Phase Algorithm

Double-checking database security management is often a standard practice, especially given the high risk of SQLi, because the proposed algorithm has been shown to be effective in reducing false rates and improving results to enhance database security. It achieved satisfactory performance when it was tested with real-world data in the operational context of a university and a commercial bank.

IV. RESULTS *AND* DISCUSSION

This research study aimed at enhancing database security through the development of a multi-phase algorithmic framework capable of real-time prevention of SQLi Attacks. In this section the results of the performance evaluation of the proposed algorithmic framework are explored.

A. Model Performance Analysis

The performance of various models was evaluated through a benchmarking process including deep learning and machine learning algorithms to determine the relative efficacy alternative algorithms comprising part of the proposed approach.

Figure 3 presents all the classifiers and their benchmarking results. It can be seen that the Convolutional Neural Network (CNN) and Naïve Bayes resulted in the best performance with accuracy 97% and 95% respectively.

B. Double-check using A Multi-Phase Algorithm.

The results of a specific algorithm were used to determine whether an attempted log-in was an attack instance or not. This proceeds as follows:

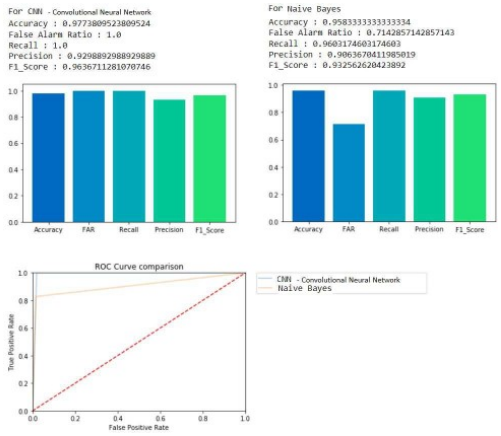

Figure 3. Comparison of all algorithms of machine learning

1) Once the selected module predicts that an SQLi attack is occurring, it blocks the attack and an alert is sent notifying that the attack is in process.

2) Once the selected module has predicted a SQLi attack, then the data will be further processed through one of the unsupervised machine learning techniques to determine the type of attack, as shown in (2).

3) If the classification determines that there is no attack, a Multi-Phase Algorithm will be used to confirm the classification and generate a final result. The Multi-Phase

Algorithm will verify and make a new a prediction, and if this were to determine that the data instance was an attack, a warning will be issued that an attack is in process; since the selected module had not detected the attack, the workflow means that the next step would be back to step-2.

4) Irrespective of whether the Multi-Phase Algorithm has detected an attack or not, the IP address and Mac address of the data instance will be submitted to the approved list to be checked again to see if the data instance represents a trusted sources list.

5) The database connection will be successful if both the IP address and the Mac address are in the trusted list.

6) If both the IP address and the Mac address are not in the trusted list, the SQL statement will be double-checked to see if it is intended for login or merely retrieving data.

7) The connection to the database can be allowed if the query was for login only. However, if the needed query is used to obtain the data from the database, the connection to the database will be disallowed, and the workflow next step will go to step-2

C. Field Testing the Multi-Phase Algorithm Framework with real data

The proposed approach has been tested in some Libyan state institutions with setup as illustrated in Figure 4, and the results are presented in table 1.1, below.

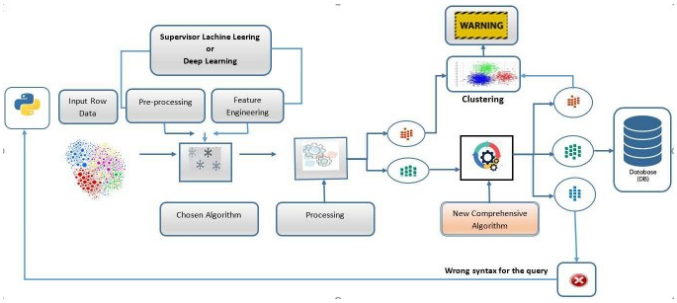

Figure 4. Testing the Multi-Phase Algorithm

Table 1. Results of the testing proposal approach

Statement	*Dataset Rows*	*Selected ML module*	*Multi-Phase Algorithm*
Normal SQLi	14147	13938	12921
Non- SQLi		209	223
Syntax Error		0	1003

V. *Conclusion* and Future Scope

A SQLi attack occurs as a result of flaws in query design which expose a vulnerability whereby an attacker can take advantage by inserting code into the query to alter the normal query, leading to unauthorised database access. This paper has proposed a Multi-Phase Algorithmic Framework which was designed, developed and evaluated in the lab through benchmarking as well as being tested with real-world data in the operational context of a university and a commercial bank to validate its performance. The results have shown improved parameterised machine learning and deep learning to enhance database security in real-time at the database layer. It can be concluded that the proposed algorithmic framework technique can efficiently detect and prevent a large subset of SQLi attacks in real-time.

The authors will continue to work on more advanced ways to prevent different types of vulnerabilities in web apps in the future.

References

[1] Statista,https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/, 2022.

[2] O W A S P, https://www.owasp.org /, 2022.

[3] Jamil. A and Zawiyah. M. Y, ―Information Security Governance Framework of Malaysia Public Sector‖, Asia-Pacific Journal of Information Technology and Multimedia, vol. 7 no. 2, 2018, pp.85 – 98

[4] Ma. L, Zhao. D, Gao. Y and Zhao. C, ―Research on SQLi Attack and Prevention Technology Based on Web", International Conference on Computer Network, Electronic and Automation (ICCNEA), Xi'an, China, 2019, pp. 176-179.

[5] Ahmed. M.A, Ali. F, Multiple-path testing for cross site scripting using genetic algorithms. J. Syst. Archit. 000, 1–13 (2015) https://doi.org/10.1016/j.sysarc.2015.11.001

[6] Jang. Y, Choi. J, Detecting SQLi attacks using query result size. Computer Security, 1–15 (2014) https://doi.org/10.1016/ j.cose.2014.04.007

[7] McWhirter. P. R., Kifayat. K, Shi. Q, and Askwith. B, "SQLi Attack classification through the feature extraction of SQL query strings using a Gap-Weighted String Subsequence Kernel," Journal of information security and applications, vol. 40, pp. 199–216, 2018.

[8] Ashlam, A. A., Badii, A., & Stahl, F. (2022). WebAppShield: An Approach Exploiting Machine Learning to Detect SQLi Attacks in an Application Layer in Run-Time. International Journal of Computer and Information Engineering, 16(8), 294-302..

[9] J. Clarke, SQLi Attacks and Defense, Second Edition, 2012. Syngress, pp.1–473.

[10] Pullagura. P. S. P. and Gokilavani, ―Defeating SQLi on Preventing Run Time Attacks‖, International Journal Of Science & Technology, 2(5), 2014, pp. 93–96.

[11] Ashlam, A. A., Badii, A., & Stahl, F. (2022, March). A Novel Approach Exploiting Machine Learning to Detect SQLi Attacks, in 2022 5th International Conference on Advanced Systems and Emergent Technologies (IC_ASET) (pp. 513-517). IEEE.

[12] Gautam, B., Tripathi, J., & Singh, S. (2018). A secure coding approach for prevention of SQLi attacks. International Journal of Applied Engineering Research, 13(11), 9874-9880.

[13] Uwagbole, S. O., Buchanan, W. J., & Fan, L. (2017, May). Applied machine learning predictive analytics to SQLi attack detection and prevention. In 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM) (pp. 1087-1090). IEEE.

[14] Patil. A, Laturkar, Athawale. A. S. V, Takale. R, & Tathawade, P. (2017, August). A multilevel system to mitigate DDOS, brute force and SQLi attack for cloud security. In 2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC) (pp. 1-7). IEEE.

[15] Joshi and Geetha. V, "SQLi Detection Using Machine learning," in Proceedings of the 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp. 1111–1115, IEEE, Kanyakumari, India, July 2014.

[16] Kamtuo. K and Soomlek. C, "Machine Learning for SQLi Prevention on Server-Side scripting," in Proceedings of the 2016 International Computer Science and Engineering Conference (ICSEC), pp. 1–6, IEEE, Chiang Mai, *ailand, December 2016.

[17] Chen. X. C, Research on Algorithm and Application of Deep Learning Based on Convolutional Neural Network, Zhejiang University of Commerce and Industry, Zhejiang, China, 2014, In Chinese.

[18] Sirinam. P, Imani. M, Juarez. M, and Wright. M, "Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep learning," in Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 1928–1943, Toronto Canada, October 2018.

[19] Anandha. K.S.S, Adhil.N. S, Priya. P. S, and Sreedeep. A.L" SQLi Detection Using Machine Learning", REVISTA GEINTEC-GESTAO INOVACAO E TECNOLOGIAS, vol: 11, pp:300-310,2021.