SQL Injection Prevention Using Machine Learning

The web is currently the most reliable and popular form of commercial and personal communication. On the web, users load millions of gigabytes of data every day through a variety of routes, and user input might be malevolent. As a result, security becomes a crucial component of web applications.

Due to their accessibility, they are vulnerable to several flaws that, if ignored, might be harmful. These gaps are used by the attackers to engage in a variety of illicit operations that allow them to get unauthorized access. One such attack that is simple to carry out but challenging to detect due to its various forms and channels is SQL Injection.

by Rahul V

HT.no: 1053-22-504-005

M.Sc.(Computer Science)

Introduction to SQL Injection

SQL Injection Attacks

The internet is a popular platform for many enterprises and everyday transactions around the world. Relational databases, which are accessed through statements written in a unique language known as Structured Query Language (SQL), constitute the foundation of every web-based application. Using unconstrained user input parameters, the attacker injects SQL characters or keywords into a SQL statement to change the original query's logic. This injection technique is used to attack websites.

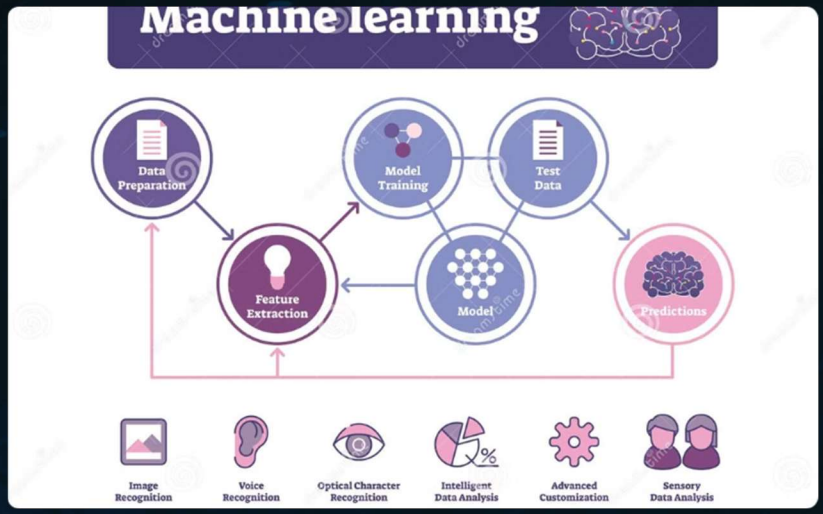
Vulnerability and Prevention

A query is created each time a user-generated request is made. The user input in the query could be harmful. It's crucial to teach our web apps that user input can come from dangerous sources and comes from outside sources. Therefore, before it is actually executed, we need to process it. The programmer is in charge of creating sophisticated code that thwarts any unauthorized entry. However, because of carelessness or ignorance, the user input is left unprocessed, giving the attacker a way in to the system.

Existing System and Its Limitations

Existing Solutions

This paper analyzes existing solutions to SQL injection problems, such as AMNESIA and SQLrand. The limitations of these solutions are discussed. A classifier for detecting SQL injection attacks has been devised.



Machine Learning Algorithms

The research utilizes machine learning algorithms, including Decision Tree, Conversions, and SVM. These algorithms are used to develop a classifier for detecting SQL injection attacks.

Proposed System Overview

Machine Learning Algorithms

The proposed system utilizes various machine learning algorithms. A classifier combines these algorithms, including the Naïve Bayes algorithm.

Testing and Evaluation

The proposed model is rigorously tested using test cases derived from three SQLIA attacks: comments, union, and tautology.

Advantages and Literature Survey

Advantages

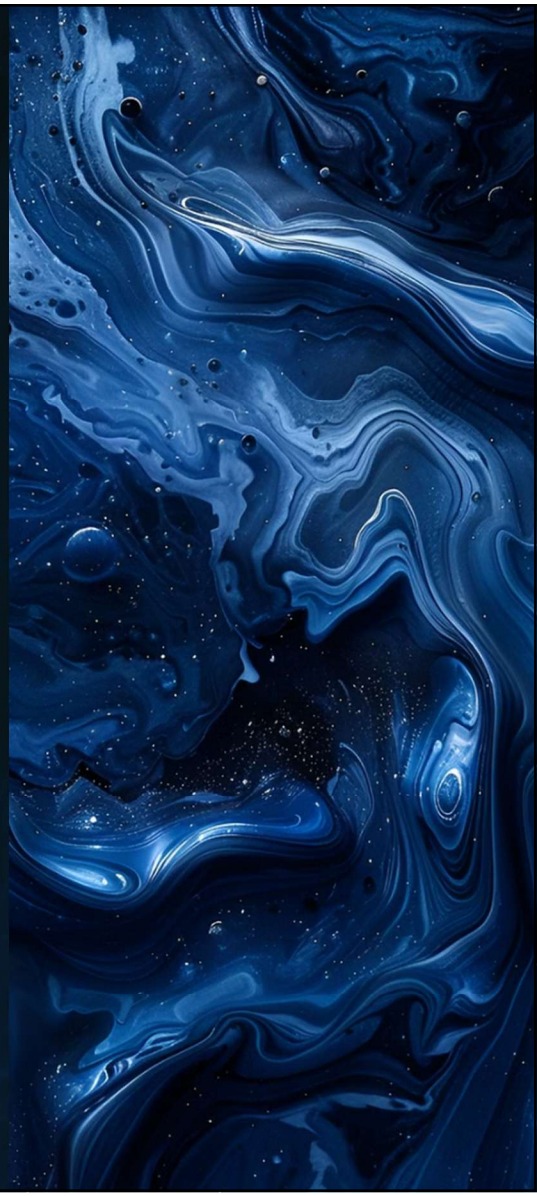
The proposed system offers several advantages over existing methods. It leverages machine learning algorithms to achieve higher accuracy in detecting SQL injection attacks. By utilizing a diverse training dataset, the system can learn to identify subtle patterns and variations in malicious queries.

Literature Survey

Extensive research has been conducted on SQL injection detection and prevention. Existing approaches can be broadly categorized into two types: static analysis and dynamic monitoring. Static analysis focuses on analyzing source code to identify potential vulnerabilities, while dynamic monitoring involves real-time analysis of SQL queries during execution.

Combined Approaches

Recent research highlights the benefits of combining static and dynamic approaches for more reliable SQL injection detection. Tools like AMNESIA utilize both methods to create a comprehensive system that can effectively identify and prevent malicious queries.



System Requirements

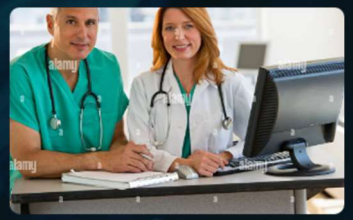
Hardware

The system requires Windows 8 and 10 operating systems, both 32 and 64 bit versions. The minimum RAM requirement is 4GB.

Software

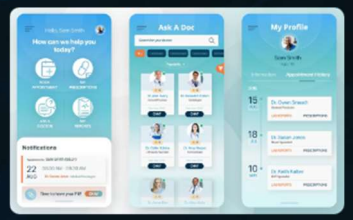
The system is built using Python language. Anaconda Navigator is the preferred software environment for running the system. Jupyter Notebook is used for coding and development.

Non-Functional Requirements



Scalability

The system should be designed to scale easily to accommodate an increasing number of new-borns and medical data. It should be able to handle growth in both the number of users and the volume of historical and real-time data.



Usability

The user interface should be intuitive and user-friendly, considering that healthcare professionals with varying technical expertise will use the system. Training requirements for healthcare professionals to use the system should be minimal.

Interoperability

The system should integrate seamlessly with existing electronic health record (EHR) systems and other healthcare information systems. It should adhere to relevant healthcare data exchange standards to ensure interoperability with various healthcare providers.

Security

The system should comply with healthcare data security standards (e.g., HIPAA) to protect patient confidentiality and privacy. Access to sensitive medical data should be restricted to authorized healthcare professionals.

Modules and Data Processing Steps

1

Data Collection

Data collection is a process in which information is gathered from many sources which is later used to develop the machine learning models. The data should be stored in a way that makes sense for the problem. In this step, the data set is converted into the understandable format which can be fed into machine learning models.

2

Data Pre-Processing

Organize your selected data by formatting, cleaning and sampling from it. Three common data pre-processing steps are: Formatting, Cleaning, and Sampling.

3

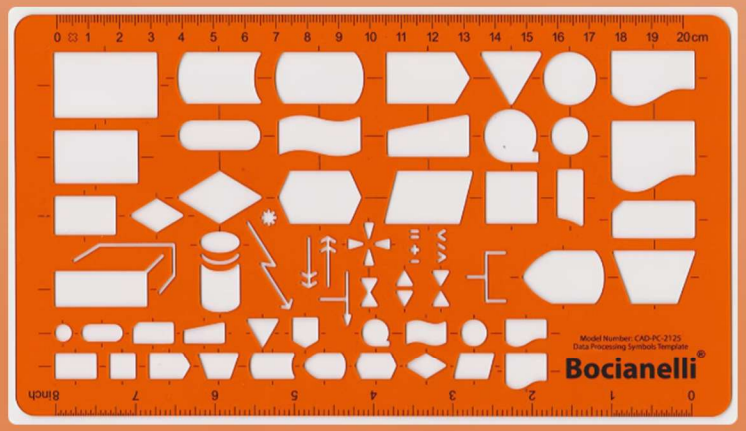
Feature Extraction

Feature extraction is an attribute reduction process. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes.

4

Evaluation Model

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future.



Results and UML Diagrams

Finally, you will get results as accuracy metrics. The Unified Modeling Language (UML) is used to specify, visualize, modify, construct, and document the artifacts of an object-oriented software intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as actors, business processes, logical components, activities, programming language statements, database schemas, and reusable software components.

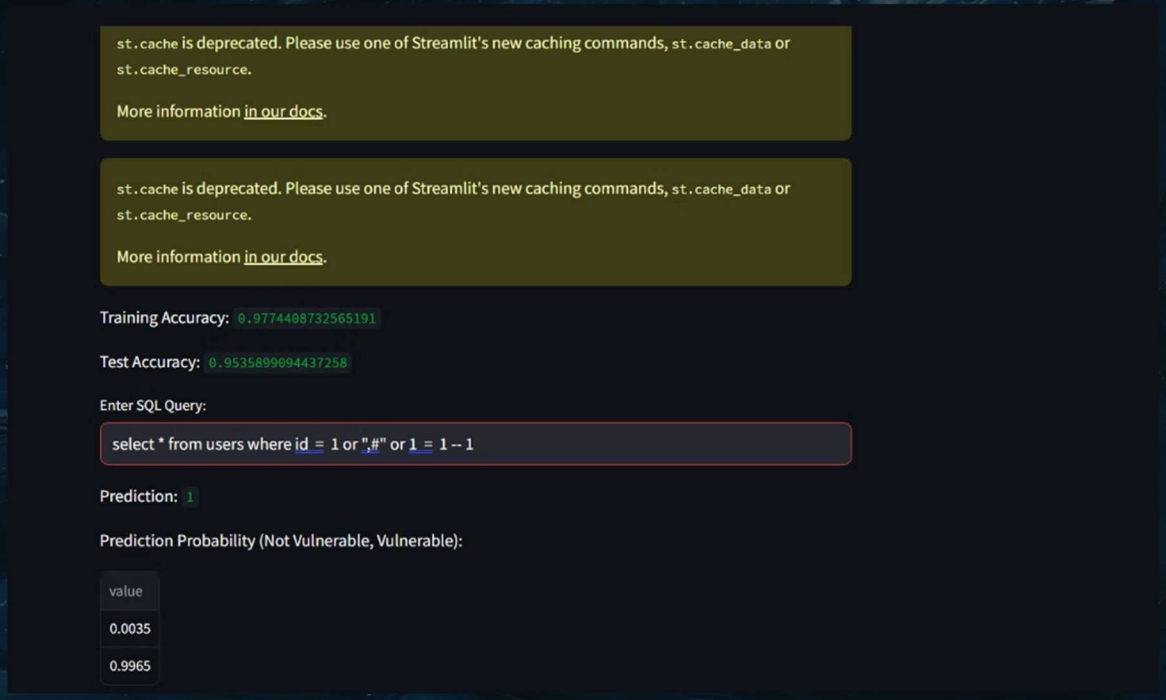
UML combines best techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies. UML has synthesized the notations of the Booch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language.

Sequence Diagram	Use Case Diagram	Activity Diagram	Class Diagram
Represents the objects participating the interaction horizontally and time vertically.	A kind of behavioral classifier that represents a declaration of an offered behavior.	Graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency.	The main building block of object-oriented modeling.

Output Screen

The output screen is the main interface for users. It shows the results of data processing and analysis.

Key features include customizable visualizations, filtering, sorting, and downloading or sharing data. The screen also provides helpful information to interpret the results, so users can make informed decisions.



Conclusion and Future Work

Conclusion

In this paper, we have proposed a method for detection of SQL injection attack based on Naïve Bayes Machine Learning Algorithm combined with Role Based Access Control. This approach leverages the power of machine learning to identify malicious SQL queries and enhance security.

Future Work

Future work could involve exploring other machine learning algorithms, such as Support Vector Machines or Random Forests, to further improve the accuracy and efficiency of SQL injection detection. Additionally, incorporating real-time analysis and dynamic updates to the model could enhance its adaptability to evolving attack patterns.

THANK YOU !

We're grateful for the time and feedback you've provided on our proposal. We look forward to continuing our collaboration and bringing this vision to life. Please don't hesitate to reach out if you have any questions.

Contact Me On:

Email Id: vasanthapuramrahul@gmail.com

Phone No: 9700349693