SQLi Prevention and Response in the Age of AI: A review of Recent SQLi Literature

Christoph Hagenauer, Shaun Poole, Daniel Scheer

Faculty Mentor Dr. Ronald Erdei

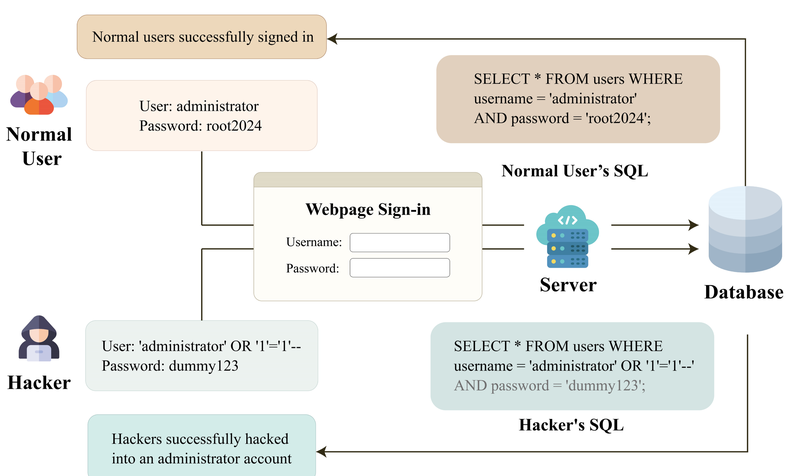
Department of Computer Science & Mathematics, University of South Carolina Beaufort, Bluffton, SC 29909

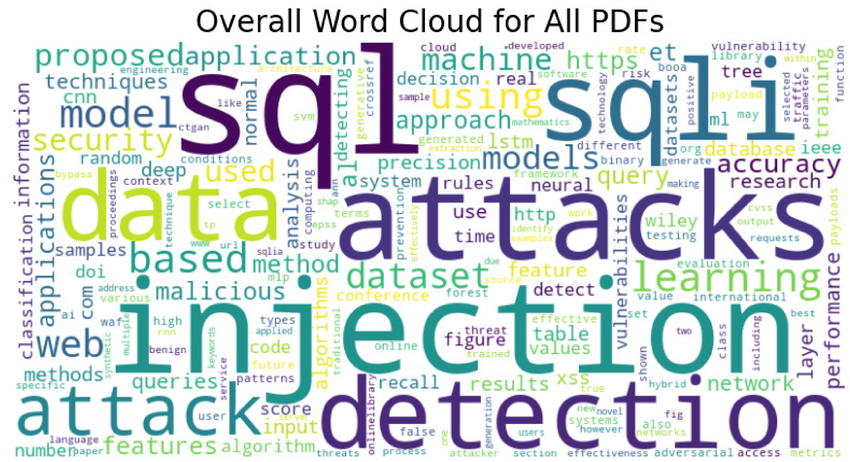
Abstract

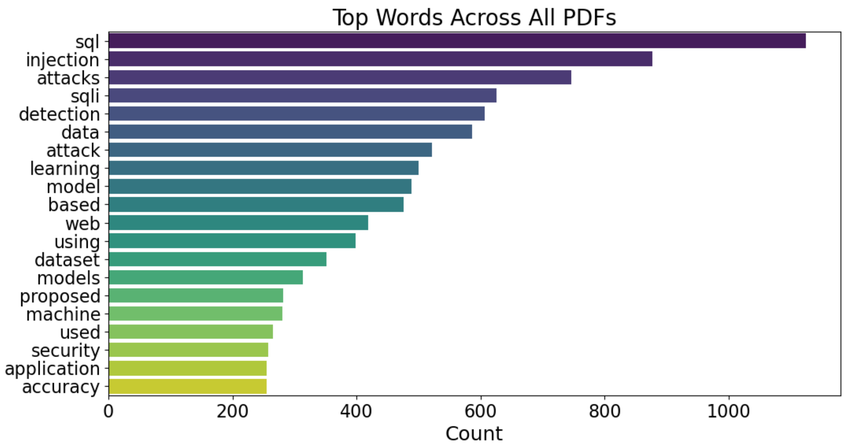
SQL Injection (SQLi) is a common cybersecurity vulnerability / attack vector. Our work provides an overview of recent developments in the field of SQLi prevention, detection, and response. Specifically, how SQLi prevention, detection, and response have evolved as Artificial Intelligence (AI) and Machine Learning Algorithms (ML) have been leveraged to combat SQLi attacks. A comprehensive overview of these topics as derived from analysis of 24 papers published in the past 4 years is provided (i.e., papers published since the advent of AL/ML), as is relevant historical background. Additionally, useful resources for novices in the field of cybersecurity are identified and shared.

SQLi Basics

Structured Query Language (SQL) is a standardized programming language used to interact with databases. These databases can have any kind of information regarding company sales records, clientele contact info, and in some cases, clients home addresses, banking information, and so on. Anywhere data can be inserted in web forms or websites, there is likely a database storing all the input information. If the database uses SQL to communicate, theres a chance the system can be manipulated into giving unauthorized users information about the database or even the information it contains. While this issue is very well known among industry professionals, it still ranks among the top web security issues.







Methodology

We began our research by formulating 3 questions:

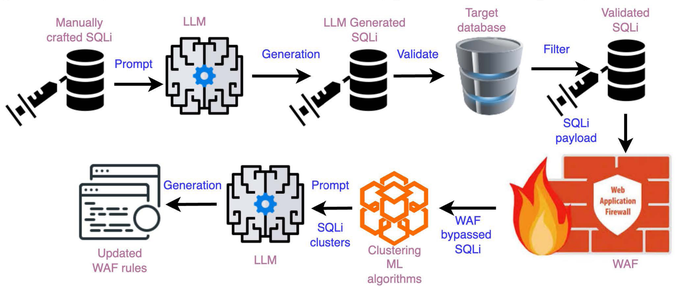
- RQ1: How has SQL injection attack detection evolved throughout the years?
- RQ2: Which AI/ML methods are effective for SQL-injection detection?
- RQ3: Which are the most popular methods used in terms of AI, ML, and SQL-injection?

We reviewed all selected papers to gain an understanding of what the progression of SQLi has been. In our review, we identified two papers published within the past 6 months and highlighted their novel approaches to improving the use of ML in SQLi detection. Both papers created frameworks that can be used to train a ML models to detect, and prevent SQLi attacks.

Highlighted ML Methods

GenSQLi

The framework presented in this paper utilizes popular Large Language Models (LLM), such as ChatGPT and Google Gemini, to write SQLi queries to target and manipulate a database. If the SQLi attack was successful, the framework passes the malicious query to an application firewall to test firewall rules. If the query passes through the firewall successfully, the framework flags the query for identification. Once categorized, the query is used to train another LLM which generates an appropriate update to the firewall rules. This can be implemented in to the firewall to protect against any attacks from a similar type of attack query.



Light-BERT

The framework presented in this paper focuses on improving current training methodology for LLMs with the goal of running an accurate lightweight model using low-end system hardware. A popular approach to training SQLi detection models involves using a pre-training model called the Bidirectional Encoder Representations from Transformers (BERT). BERT excels at generating valid SQLi queries which can be used to train accurate models. However, BERT requires a vast amount of system resources, making it nonviable in some production environments. The proposed framework, which we have dubbed Light-BERT, relies on extracting SQL keywords from incoming queries and accurately identifying their intent "on the fly". Once analyzed, the query is fed into and evaluated by a lighter weight Convolution Neural Network (CNN). The framework then double checks the results of the CNN and finally identifies if the query is safe or malicious. By evaluating the query as it is being sent to the system, this framework allows low end systems to run accurate ML models, improving their security.

Scan the QR code to view our references