Abstract

Web-based applications have become an integral part of our daily lives, but at the same time, security of sensitive information stored in the web databases has been a growing concern. SQL injection attacks have been a predominant and pervasive security threat on web applications over the last one and half decades. The vulnerability has been exploited by attackers resulting in massive data-breach incidents. Sophisticated and automated attack tools have made the situation alarming than ever before. Despite extensive research on mitigation of SQL injection attacks, the threat has persisted with nearly the same intensity. Over time, SQL injection attack vectors have evolved with new forms and features, which has made the problem even more challenging.

In this thesis, we investigate the technical details of the problem, systematically study the literature, and develop solutions for prevention and detection of SQL injection attacks using machine learning paradigms. Stepping aside from the traditional way of viewing SQL queries as executable expressions or parse-trees thereof, we focus on learning their structural construction by modeling them as reducible strings, documents, state sequences, and interaction networks of tokens. Beginning with prevention of SQL injection attacks at the application level, we proceed into developing solutions for server level protection which are specifically useful on shared hosting environments. Apart from accuracy and performance, we also emphasize on ease of implementation, platform independence, and adaptive learning in the studies.

The first study addresses prevention of SQL injection attacks at the application level, which is based on anomaly detection against a normal use query model. We introduce an important idea that, the identifiers and literals are irrelevant towards the structural composition of SQL queries. A query transformation scheme is designed to convert SQL queries into their structural form, which produces a minimal normal-query model. Fast and efficient matching at runtime is performed by using a hashing mechanism. The approach is found to be effective with negligible performance overhead.

Application level protection is generally inadequate on shared hosting environments. Therefore, we focus on the database firewall layer, and propose three solutions for detection of SQL injected queries. The first study employs document similarity measure typically used in Information Retrieval (IR) domain. The query transformation scheme is extended to normalize SQL queries into sentence-like forms, facilitating application of document similarity measures. A novel strategy to examine only the WHERE clause part of incoming queries is introduced, which substantially reduces the processing overhead. The system is trained with large number of injected queries collected from sample
web applications using several automated attack tools. At runtime, known injection vectors are identified by hash matching, while previously unseen attacks are detected by computing document similarity with clusters of attack vectors stored in its knowledge base. The system learns new attack vectors from its detection history as well as inputs from experts, such as the DBA.

Next, with further improvements to query normalization, we present another approach to detect SQL injected queries using a twin HMM ensemble. The two HMM ensembles consist of multiple HMMs, each of which is trained with a set of similar genuine and injected query samples. The outputs of the HMMs are combined in a decision module to determine whether a runtime query is malicious or not. When a conclusive decision cannot be arrived at, the query is marked as suspicious and logged for the DBA for further investigation. The system is found to be quite effective with low performance impact, with the drawback of high training time of HMMs.

The last study presented in this thesis draws inspiration from use of graph theoretic analysis of Social Networks and presents a novel approach to detect injection attacks by modeling SQL queries as graphs of tokens. The undirected and directed graphs are constructed using a sliding-window technique and the degree of interaction between the tokens is captured. The centrality of nodes are used to obtain the vector representation, which are then used to train an SVM classifier. We also propose alternative designs of the detection system comprising of single and multiple SVMs, and compare the results. The proposed systems offer high accuracy, low false alarms, and negligible performance impact. Furthermore, the study investigates feature subset selection using information gain method, and examines the applicability of other centrality indices which are popularly used in Social Network Analysis domain.

**Keywords** — SQL injection attack, SQLIA prevention, Query Transformation, Normalization of Queries, Document Similarity, Hidden Markov Model, Support Vector Machine, Graph of Tokens, Centrality Measures, Feature Selection.