CBRid4SQL: A CBR Intrusion Detector for SQL Injection Attacks

Cristian Pinzón[1,2], Álvaro Herrero[3], Juan F. De Paz[1], Emilio Corchado[1], and Javier Bajo[1]

[1] Departamento de Informática y Automática, Universidad de Salamanca, Plaza de la Merced s/n, 37008, Salamanca, Spain
{cristian_ivanp,fcofds,escorchado,jbajope}@usal.es
[2] Universidad Tecnológica de Panamá, A.P: 0819-07289, Panamá, Rep. De Panamá
[3] Department of Civil Engineering, University of Burgos, Spain
C/ Francisco de Vitoria s/n, 09006 Burgos, Spain
ahcosio@ubu.es

Abstract. One of the most serious security threats to recently deployed databases has been the SQL Injection attack. This paper presents an agent specialised in the detection of SQL injection attacks. The agent incorporates a Case-Based Reasoning engine which is equipped with a learning and adaptation capacity for the classification of malicious codes. The agent also incorporates advanced algorithms in the reasoning cycle stages. The reuse phase uses an innovative classification model based on a mixture of a neuronal network together with a Support Vector Machine in order to classify the received SQL queries in the most reliable way. Finally, a visualisation neural technique is incorporated, which notably eases the revision stage carried out by human experts in the case of suspicious queries. The Classifier Agent was tested in a real-traffic case study and its experimental results, which validate the performance of the proposed approach, are presented here.

Keywords: SQL Injection, Intrusion Detection, CBR, SVM, Neural Networks.

1 Introduction

Over recent years, one of the most serious security threats around databases has been the SQL Injection attack [1]. In spite of it being a well-known type of attack, the SQL injection remains at the top of the published threat list. The solutions proposed so far [2], [3], [4], [5], [6], [7], [8] seem insufficient to prevent and block this type of attack because these solutions lack the learning and adaptation capabilities for dealing with attacks and their possible variations in the future. In addition, the vast majority of solutions are based on centralized mechanisms with little capacity to work in distributed and dynamic environments.

This study presents the intelligent agent CBRid4SQL (a CBR Intrusion Detector), capable of detecting attacks based on SQL code injection. CBRid4SQL is an agent specially designed following the strategy of an Intrusion Detection System (IDS) and is defined as a Hybrid Artificial Intelligence System (HAIS). This agent is

E.S. Corchado Rodriguez et al. (Eds.): HAIS 2010, Part II, LNAI 6077, pp. 510–519, 2010.
© Springer-Verlag Berlin Heidelberg 2010

the principal component of a distributed hierarchical multi-agent system aimed at detecting attacks in dynamic and distributed environments.

The CBRid4SQL agent is a CBR agent [9] that is characterized by the integration of a CBR (Case-Based Reasoning) mechanism. This mechanism provides the agents with a greater level of adaptation and learning capability, since CBR systems make use of past experiences to solve new problems [9]. This is very effective for blocking SQL injection attacks as the mechanism uses a strategy based on anomaly detection [10].

Additional to the incorporated CBR motor in the CBRid4SQL agent's internal structure, an integrated mixture through an Artificial Neural Network (ANN) and a Support Vector Machine (SVM) are used as a mechanism of classification. Through the use of this mixture, it is possible to exploit the advantages of both strategies in order to classify the SQL queries in a more reliable way.

Finally, to assist the expert in the making of decisions regarding those queries classified as suspicious, a visualization mechanism is proposed which combines clustering techniques and neural models to reduce the dimensionality based on unsupervised learning. The rest of the paper is structured as follows: section 2 presents the problem that has prompted most of this research work. Section 3 explains the internal structure of the CBRid4SQL agent used as a classifier agent. Finally, the conclusions and experimental results of this work are presented in section 4.

2 SQL Injection Attacks

An SQL injection attack takes place when a hacker changes the semantic or syntactic logic of an SQL text string by inserting SQL keywords or special symbols within the original SQL command which is executed at the database layer of an application [1]. Different attack techniques exist which include the use of SQL Tautologies, Logic errors / Illegal queries, union query and piggy-backed queries. Other more advanced techniques use injection based on interference and alternative codification [1]. The cause of the SQL injection attacks is relatively simple: an inadequate input validation on the user interface. As a result of this attack, a hacker can be responsible for unauthorized data handling, retrieval of confidential information, and in the worst possible case, taking over control of the application server [1].

Different strategies have been presented as a solution to the problem of SQL injection attacks [1], with special attention given to strategies based on IDSs [2], [3], [4], [5], [6], [7], [8]. One approach based on anomaly detection was proposed by [2], applying a clustering strategy to group similar queries and isolate queries which are considered malicious. The main disadvantage of this approach is in its high computational overhead which would affect a real-time detection. Kemalis and Tzouramanis propose SQL-IDS (SQL Injection Detection System) [3] that uses security specifications to capture the syntactic structure of the SQL queries generated by the applications. The main limitation of this approach is the computational cost while comparing the new query with the predefined structure at runtime.

In [4] two types of SQL injection attacks are raised: tautology attacks and those based on the UNION operator. Through the syntactic analysis of SQL query strings, the data of the HTTP requests are extracted to later be used in the training phase and