Classification Agent-Based Techniques for Detecting Intrusions in Databases

Cristian Pinzón[1], Yanira De Paz[2], and Rosa Cano[3]

[1] Universidad Tecnológica de Panamá, Av. Manuel Espinosa Batista, Panama
[2] Universidad Europea de Madrid, Tajo s/n 28670, Villaviciosa de Odón, Spain
[3] Instituto Tecnológico de Colima, Av. Tecnológico s/n, 28976, Mexico
cristian.pinzon@utp.ac.pa, yanirarosario.depaz@uem.es,
rdegca@gmail.com

Abstract. This paper presents an agent specially designed for the prevention and detection of SQL injection at the database layer of an application. The agent incorporates a Case-based reasoning mechanism whose main characteristic involves a mixture of neural networks that carry out the task of filtering attacks. The agent had been tested and the results obtained are presented in this study.

Keywords: SQL injection, multiagent systems, case-based reasoning, neural networks.

1 Introduction

Database security is a fundamental aspect of all current information systems. There are many ways of exploiting the security vulnerability in a relational database. SQL injection is one of more common types of attacks at the database layer of desktop and Web applications. SQL injection occurs when the intended effect of the SQL sentence is changed by inserting SQL keywords or special symbols [1]. The problem of SQL injection attacks has been traditionally addressed by using centralized architectures [2], [3]. Because this type of solution is incomplete, several types of intrusion detection system (IDS) solutions have been proposed [4]. Although IDSs are effective, there are a number of drawbacks such as a large number of false positives and negatives, limited learning capacity, and limited ability in adapting to changes in attack patterns.

This article presents a CBR-BDI [5] deliberative agent based on the BDI (*Belief*, *Desire*, *Intention*) [6] model specifically designed for the detection and prevention of SQL injection attacks in database layers. Our study applies a novel case-based reasoning (CBR) [7] [8] classification mechanism that incorporates a mixture of neural networks capable of making short term predictions [9].

This proposal is an innovative approach that addresses the problem of SQL injection attacks by means of a distributed artificial intelligence technique. Specifically, it combines the characteristics of multiagent systems such as autonomy, pro-activity, social relations, etc., [5] with CBR [7]. CBR Systems are adequate in dealing with

E. Corchado, A. Abraham, and W. Pedrycz (Eds.): HAIS 2008, LNAI 5271, pp. 46–53, 2008.

SQL injection attacks, insomuch as these systems find solutions to new problems by using previous experiences. This fact allows us to equip our classifier agents with a great capacity for adapting and learning, thus making them very adept in resolving problems in dynamic environments. The system developed within the scope of this work proposes a solution which combines a distributed approach and an advanced classification system, incorporating the best of both approaches.

The rest of the paper is structured as follows: section 2 presents the problem that has prompted most of this research work. Section 3 focuses on the structure of the classifier agent which facilitates the detection and prevention of malicious injection attacks, and section 4 explains in detail the classification model integrated within the classifier agent. Finally, section 5 describes how the classifier agent has been tested inside a multi-agent system and presents the results obtained.

2 SQL Injection Problem Description

A SQL injection attack affects the security of personal, social, financial and legal information for both individuals and organizations. A SQL injection attack takes place when a hacker changes the semantic or syntactic logic of a SQL text string by inserting SQL keywords or special symbols within the original SQL command that will be executed at the database layer of an application [1]. SQL injection attacks occur when user input variables are not strongly typed, thus making them vulnerable to attack. As a result, these attacks can produce unauthorized handling of data, retrieval of confidential information, and in the worst possible case, taking over control of the application server [2]. One of the biggest problems with SQL injection is the various forms of vulnerabilities that exist. Some of the better known strategies, such as tautologies, syntax errors or illegal queries, and *union* operators, are easy to detect. However other strategies can be extremely complex due to the high number of variables that they can generate, thus making their detection very difficult. Some examples of these strategies are inference mechanisms, data storage procedures, and alternative encoding.

Traditional security mechanisms such as firewalls or IDSs are not very efficient in detecting and preventing these types of attacks. Other approaches based on string analysis, along with dynamic and static analyses such as AMNESIA (Analysis and Monitoring for Neutralizing SQL Injection Attacks) [2], have the disadvantage of addressing just one part of the problem, and therefore deliver only a partial solution. Moreover, the approaches based on models for detecting SQL injection attacks are very sensitive. With only slight variations of accuracy, they generate a large number of false positive and negatives.

Some innovative proposals are incorporating artificial intelligence and hybrid~~s~~ systems. Web Application Vulnerability and Error Scanner (WAVES) [10] uses a black-box technique which includes a machine learning approach. Valeur [4] presents an IDS approach which uses a machine learning technique based on a dataset of legal transactions. These are used during the training phase prior to monitoring and classifying malicious accesses. Rietta [11] proposed an IDS at the application layer using an anomaly detection model which assumes certain behaviour of the traffic generated by the SQL queries; that is, elements within the query (sub-queries, literals, keyword