A System to Detect and Block SQL Injection with the help of Multi Agent System using Artificial Neural Network

Rohit Garde
Department of Information Technology, Sinhgad Academy of Engineering, Pune, India.

D R Anekar
Department of Information Technology, Sinhgad Academy of Engineering, Pune, India.

Niraj Kulkarni
Department of Information Technology, Sinhgad Academy of Engineering, Pune, India.

Mayur Ghadge
Department of InformationTechnology, Sinhgad Academy
of Engineering, Pune, India

ABSTRACT

This paper describes a Multi agent system which uses Artificial Neural Network algorithm, which helps to detect malicious SQL queries. As SQL injection queries are one of the most hazardous attacks for database security in today’s database system, this multi agent system is useful to catch SQL injection attacks. This system possesses a multi level architecture which uses multiple agents, where each level is assigned with some. The SQL injection attacks are one of the biggest security threats in databases. SQL injection is a technique used to take advantage of non-validated input vulnerabilities to pass SQL commands through a Web application for execution by a back-end database. This system checks each query rigorously and goes through idCBR cycle i.e case based reasoning is done which gives the output legal/illegal/suspicious.

Keywords

ANN(Artificial Neural Network), CBR, Detect and Block SQL injection, multi agents, database security, MAS(Multi-Agent System).

1. INTRODUCTION

In essence, SQL Injection attack takes place because the fields available for user input allow SQL statements to pass through and query the database directly[1]. It typically involves malicious modifications of the user SQL input either by adding additional clauses or by changing the structure of an existing clause[1]. SQL injection enables attackers to access, modify, or delete critical information in a database without proper authorization[1]. Thus SQL injection remains at the top on list of security threats for databases. The solution proposed to prevent and block this type of attack seems insufficient because they lack the learning capabilities. Also, the majority of these solutions are based on centralized mechanisms, with little capacity to work in distributed and dynamic environments. Although researchers and practitioners have proposed various methods to address the SQLinjection problem, current approaches either fail to address the full scope of the problem or have limitations that prevent their use and adoption[5].

With the help of previous research, this study presents Multi-Agent System for detecting and Blocking SQL injection, a solution based on a distributed architecture (multi agent system – MAS) which is capable of detecting and blocking SQL injection attacks[1]. The concept of multi-agent systems makes it possible to deal with SQL injection attacks. Every component in this system interacts and cooperates to achieve a global common goal: the detection and prevention of ongoing intrusions in a database[1]. This system presents a hierarchical organization structured by layers of agents, which distributes roles and tasks to detect and prevent SQL injection attacks. The agents at each level are assigned with specific tasks, due to their own abilities they execute at any physical location[1]. The agents are characterized by the unification of a CBR (Case-Based Reasoning) mechanism in a deliberative Belief - Desire – Intention (BDI) Agent [2]. The mechanism which is provided by the agents have a greater level of adaptation capability and also learning capabilities as CBR systems use past experiences to solve new problems. This technique is effective to block SQL injection attacks as this mechanism uses a strategy which is based on anomaly detection, which model's the normal/legal SQL queries. The main innovations of this study is the incorporation of a new classification strategy which is based on data mining in CBR agents, and of an agent with special capabilities for the visualization and subsequent analysis of data.. The use of CBR agents with advanced capabilities for analysing and predicting SQL attacks is one of the main features of the architecture[2]. Furthermore, the human experts provides a very useful tool which analyses those cases which are classified as suspicious by the CBR agent and which requires classification by expert.

2. ARCHITECTURE

The present study proposes the use of a multi-agent architecture.[1] It is based on a groundbreaking technique since there is no known architecture with these characteristics for detecting SQL injection attacks[1]. It utilises all the old techniques used to block to block the SQL injection, along with the multiple agents. The distributed resolution of problems balances the workload, facilitates recovery from error conditions, and also avoids centralized traffic[1]. The working analysis, classification and decision making and among others are distributed throughout all three layers in the proposed architecture, as depicted in Fig. 1. The agents that make up the architecture are assigned specific roles to perform their tasks. Moreover, the distribution greatly simplifies the capacity to recover from errors or failures because if an agent fails, it is immediately replaced without affecting the other agents at the same level or in other levels. Additionally, the proposed architecture is based on a hierarchical model that reduces the complexity of tasks such as monitoring and capturing user requests, classifying user requests, evaluating the final solution, etc.

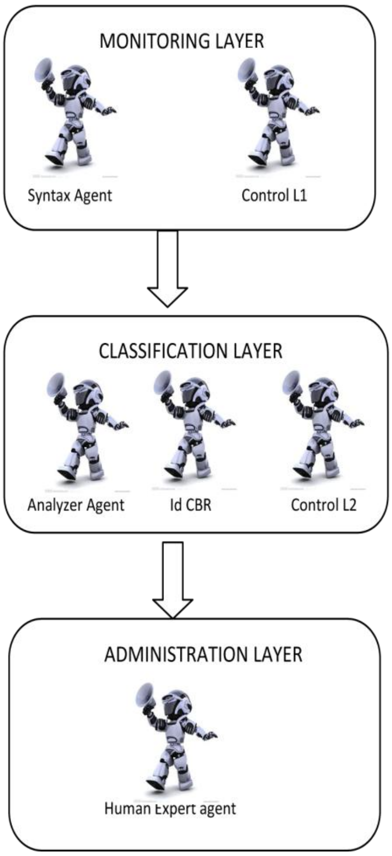


Fig. 1Architecture

Distributing the functionality at each level, while maintaining each level independently, allows new changes to be easily adapted. Each level of the architecture houses a collection of agents with well-defined roles that allow their tasks and responsibilities to be clearly specified. The architecture has been divided into three levels so that the specific tasks are assigned according to the degree of complexity. Fig. 1 depicts the system architecture with each level and the respective agents. System is presented as an evolution of the SC-MAS architecture that has proposed a strategy to identify and block SQL injection attacks through a distributed approach based on the capabilities of CBR agents. CBR agents are a particular type of CBR-BDI agents. This agent as well as with CBR-BDI agent with their capabilities to visualize is used to assist the expert in decision making regarding queries that are classified as suspicious. To do so, a visualization mechanism is proposed which combines clustering techniques and neural models, based on unsupervised learning, to reduce dimensionality.

The different types of agents located at the different levels of the architecture can be described as:

4.1. Syntax:

This agent is situated in topmost layer in the architecture that is monitoring layer and is responsible for syntax checking of the SQL query fired.

4.2. Control-L1:

Its function is to communicate with the lower layers of the architecture. It is present in the monitoring layer, and all communication from this layer is administered by the agent. This agent receives data from the Syntax agent and assigns the Analyzer agent the task of searching for patterns of attacks; and then it reports to the human expert in the administration layer the detection of any intrusion during the process of comparing attack signatures. Basically this agent controls and communicates with other levels.

4.3. Analyzer:

This type of agent is situated in the classification layer. Its work is to do matching patterns of known attacks; a database with previously built patterns allows this task.

4.4. CBR:

This type of agent is also situated in the classification layer and is a core component of the architecture as it carries out a classification of SQL strings through detection anomalies. It integrates a case based reasoning (CBR) mechanism. It generates a classification (legal, illegal or suspicious) to the query.

4.5. Control-L2:

This is the second type of agent for carrying out control and communication functions. Once the syntax checking is done in monitoring layer, this agent takes processed data from the monitoring layer and the assigns the work to CBR or Analyzer agent .All of the incoming and outgoing communication of the classification layer is administered by the Control-L2 agent. This agent is responsible for the evaluation and coordination of the overall architectural operation.

4.6. Human Expert:

This agent is located in the Administration layer; this agent facilitates the interaction between security personnel and the architecture. The human gets the query when above two layers does not process the query.

3. ARTIFICIAL NEURAL NETWORK (ANN)

An artificial neural network (ANN), also called a simulated neural network (SNN) or commonly just neural network (NN) is an interconnected group of artificial neurons that uses a mathematical or computational model for information processing based on a connectionist approach to computation. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases a neural network is an adaptive system changing its structure during a learning phase. Neural networks are used for modeling complex relationships between inputs and outputs or to find patterns in data.

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. i.e., they are self-adaptive systems.
A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions. The output of a neural network relies on the cooperation of the individual neurons within the network to operate. Since it relies on its member neurons collectively to perform its function, a unique property of a neural network is that it can still perform its overall function even if some of the neurons are not functioning. In other words it is robust to tolerate error or failure.

The computing world has a lot to gain from neural networks. They are also very well suited for real time systems because of their fast response and computational times which are due to their parallel architecture. Neural networks also contribute to other areas of research such as neurology and psychology. They are regularly used to model parts of living organisms and to investigate the internal mechanisms of the brain.

The output of a neural network relies on the cooperation of the individual neurons within the network to operate.Since it relies on its member neurons collectively to perform its function, a unique property of a neural network is that it can still perform its overall function even if some of the neurons are not functioning. In other words it is robust to tolerate error or failure. It is composed of a large number of highly interconnected processing elements (neurons) working in to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons.

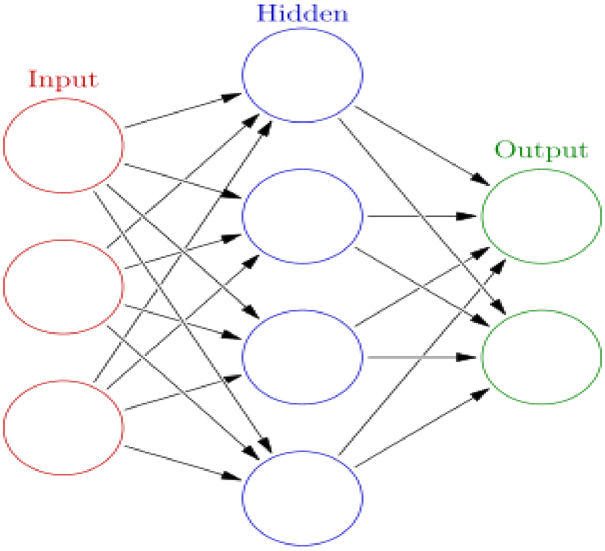


Fig 2. Artificial Neural Network

4. MULTILAYER PERCEPTRONS

A multilayer perceptron (MLP) is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate outputs. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron with a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training the network. MLP is a modification of the standard linear perceptron and can distinguish data that are not linearly separable.
The number of neurons in the hidden layer is 2n + 1, where n is the number of neurons in the input layer. Finally, there is one neuron in the output layer. The sigmoid activation function has been selected for the different layers.

Running the network consists of :

4.1 Forward pass:

The outputs are calculated and the error at the output units calculated.

4.2 Backward pass:

The output unit error is used to alter weights on the output units. Then the error at the hidden nodes is calculated (by back-propagating the error at the output units through the weights), and the weights on the hidden nodes altered using these values.
For each data pair to be learned a forward pass and backwards pass is performed. This is repeated over and over again until the error is at a low enough level.

5. WORKING OF ANN IN THIS SYSTEM

A query fired by the client is first checked by the syntactical agent to check the SQL Syntax.If the Syntactical agent’s result is that the query’s syntax is correct then the Query is passed to the Analyzer agent.

The Analyzer agent checks the passed Query whether it’s a Known attack or a legal query.

If the Analyzer agent is unable to detect the status of the query then it is declared as an Unknown Query and passes to the Case-Based-Reasoning Cycle.

The Case-Based-Reasoning cycle is as follows.

5.1 Retrieve:

Case retrieval is performed by using the Query Category attribute which retrieves queries from the case memory which were used for a similar query in accordance with attributes of the new case . Subsequently, the models for the MLP associated with the recovered cases are retrieved. The recovery of these memory models improves the system’s performance so that the time necessary for the creation of models will be considerably reduced, mainly in the case of the ANN training.

5.2 Reuse:

It begins with the information of the retrieved cases and the recovered models . The inputs of the MLP are: Affected _table, Affected_field, Command _type, Word_GroupBy, Word_Having, Word_OrderBy, Numer_And, Numer_Or, Number literals, and Length _SQL _String . The number of neurons in the hidden layer is 2 n + 1, where n is the number of neurons in the input layer. Finally, there is one neuron in the output layer.

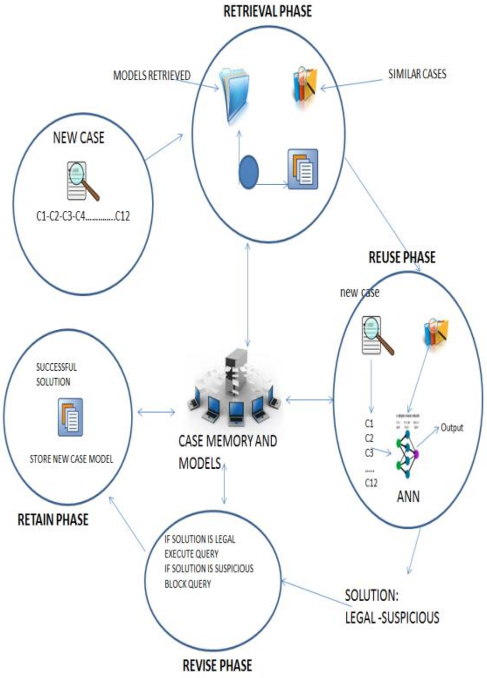


Fig 3. CBR cycle and classification mechanism of the idCBR agent

5.3 Revise:

The solution is given by the Artificial Neural Network as Legal and Suspicious and the status of the query is saved.

5.4 Retain:

The learning phase updates the information of the new classified case and reconstructs the classifiers offline to leave the system available for new classifications.

5.5 Actual Working of Case-Based-Reasoning cycle:

The Query passed by the client is is divided into 12 parts.They are as Follows:

Problem description fields	Type	Description
Affected_table	Integer	Number of tables affected by the query
Affected_field	Integer	Number of fields affected by the query
Command_type	Integer	Type of declared command in the query
Word_GroupBy	Boolean	Number of repetitions of Group By clause
Word_Having	Boolean	Number of repetitions of Having clause
Word_OrderBy	Integer	Number of repetitions of Order By clause
Number_And	Integer	Number of repetitions of the And Operator
Number_Or	Integer	Number of repetitions of the Or Operator
Number_literal	Integer	Number of Literal in the SQL string
Length_SQL_String	Integer	Length of the SQL String
Query_Category	Integer	Category of the query

These 12 fields are then stored into the database as a new row.

5.5.1 Explanation of Retrieve Phase:

The 12 fields of the query fired by the client are checked with the previous fields of various queries saved in the database. A condition has been set that atleast 8 fields of the current query should match with the rows in the database.

Such rows whose fields match, are then retrieved from the database and stored in a data structure along with the fields of the current query.This data structure is given as an input to the Reuse phase.

5.5.2 Explanation of Reuse Phase:

The output from retrieval phase is saved. Here the Algorithm “Artificial Neural Network ” is used.

Artificial Neural Network needs Input values, Hidden values.

The Input to the Neural network is the size of the data structure in which the the cases are retrieved from the retrieval phase.

The hidden value is calculated as (2*Input)+1.

And then the artificial neural network begins its work.

The output of Artificial neural network is in the range from -1 to 1.

5.5.3 Explanation of Revise Phase:

If the output generated is greater than 0 then then the query is declared as legal.

However if the output is less than 0 then the query is declared as Suspicious.

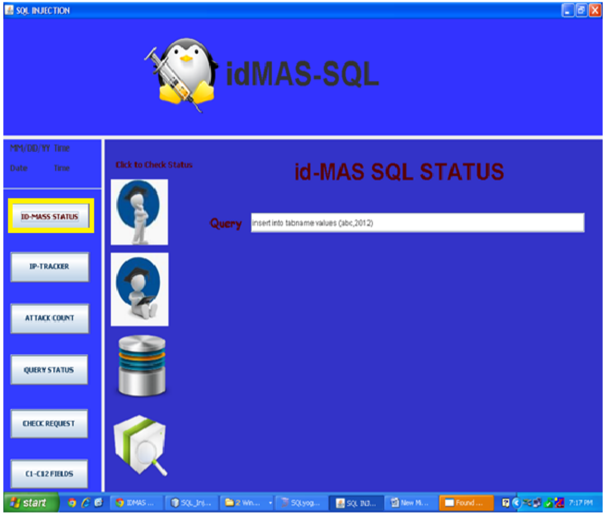
5.5.4 Explanation of Retain Phase:

The output generated by the neural network is thus saved in the database for training purpose.

6. THE SYSTEM

The System can be deployed using Client-Server architecture, where client possesses the application of firing the query to check the status of it and server side possesses the application where he gets all the queries fired by all different clients and then server check the status of the queries.

6.1 Server Side Detection Of Query



Query: insert into tabname values (abc,2012)

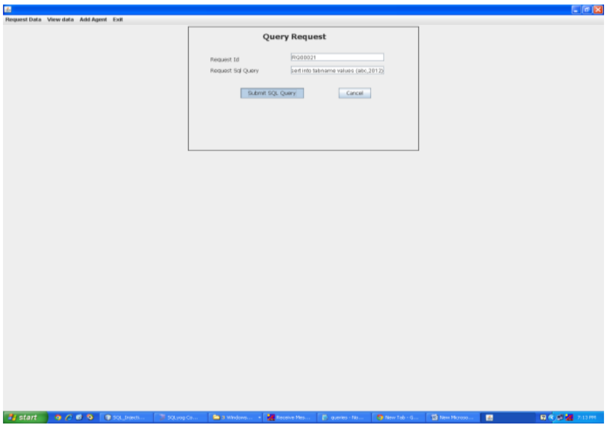
On this page the query status of the query fired is checked.

The working here is done according to the layers i.e firstly syntax is checked and analyzer agent checks the query pattern from the database.

Then this system with the help of ANN carries out case based reasoning of the query and gives the output

.

6.2 Client Side



Query Request

Request Id: RQ00021

Request Sql Query: sert into tabname values (abc,2012)

From this page the SQL query is fired. The query fired is checked by the system and output is showed to the client again.

7. ACKNOWLEDGEMENT

We take this opportunity to thank all the people involved in making this project a success. We want to thank the author whose study helped us in our study which Cristian I. Pinzon , Juan F. De Paz , Alvaro Herrero , Emilio Corchado , Javier Bajo , Juan M. Corchado. We specially thank our guide for guiding us. Our Head of Department has also been very helpful and we appreciate the support he provided us. Last but not the least we would like to convey our gratitude to all the teaching and non-teaching staff members of Information Technology, our friends and families for their valuable suggestions and support.

8. REFERENCES

[1] *NirajKulkarni ,D R.Anekar, MayurGhadge, RohitGarde* Multi-Agent System for detecting and Blocking SQL injection

[2] Cristian I. Pinzon, Juan F. De Paz, Alvaro Herrero, Emilio Corchado, Javier Bajo, Juan M. CorchadoidMAS-SQL: Intrusion Detection based on MAS to Detect and Block SQLinjection through data mining.

[3] Cristian Pinzon, Álvaro Herrero, Juan F. De Paz, Emilio Corchado, and Javier Bajo: A CBR Intrusion Detector for SQL Injection Attacks.

[4] CristianPinzón, Juan F. De Paz, Álvaro Herrero2, Emilio Corchado1, Javier:A Distributed Hierarchical Multi-agent Architecture for Detecting Injections in SQL Queries.

[5] IndraniBalasundaram,Dr. E. Ramaraj:An Approach to Detect and Prevent SQL Injection Attacks in Database Using Web Service.

[6] William G.J. Halfond, Jeremy Viegas, and Alessandro Orso:A Classification of SQL Injection Attacks and Countermeasures.

[7] Varian Luong Intrusion Detection And Prevention System: SQL Injection Attacks.

[8] Christian Bockermann, Martin Apel, and Michael Meier: Learning SQL for Database Intrusion Detection Using Context-Sensitive Modelling.

[9] SruthyManmadhan and Manesh: A METHOD OF DETECTING SQL INJECTION ATTACK TO SECURE WEB APPLICATIONS.

[10] ShaimaaEzzatSalama, Mohamed I. Marie, Laila M. El-Fangary&Yehia K. Helmy: Web Anomaly Misuse Intrusion Detection Framework for SQL Injection Detection.

[11] Lori Mac Vittie:SQL Injection Evasion Detection.