



VNiVERSiDAD
D SALAMANCA



CBRid4SQL: A CBR Intrusion Detector for SQL Injection Attacks

Authors: Cristian Pinzón
Álvaro Herrero
Juan F. De Paz
Emilio Corchado
Javier Bajo

HAIS'10



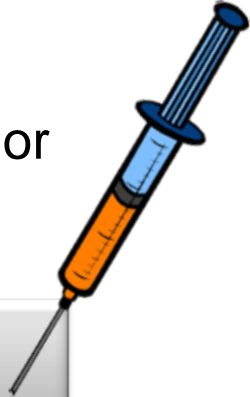
OUTLINE

- ❑ **SQL Injection Attack Problem**
- ❑ **CBR Paradigm**
- ❑ **Mechanism for the Classification of SQL Queries**
- ❑ **Results and Conclusions**

SQL Injection Attack Problem

What is a SQL Injection Attack?

- ❑ It is the most common type of attack on the database layer.
- ❑ SQL injection occurs when is inserted SQL keywords or special symbols into the original SQL string's request.



```
"Select field1, field2, field3 from table1  
Where field1 =" or 9876 = 9876 -'and field2="
```

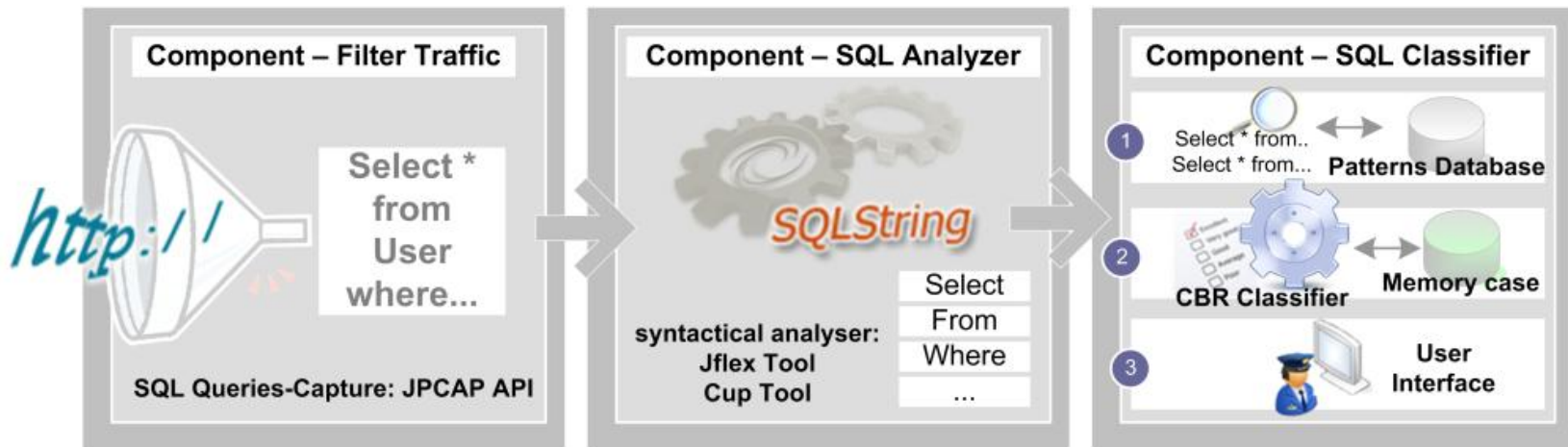
- ❑ An attacker can take advantage of a vulnerable application by providing inputs with malicious SQL commands. A SQL injection attack brings harm to the organizations.
- ❑ Many solutions had addressed this attack. This attack is a potential threat at Information Systems. It remains on the top of the published threat lists.

Case-Based Reasoning Technology

- ❑ CBR systems make use of past experiences to solve new problems.
- ❑ The fundamental concept when working with case-based reasoning is the concept of case.
- ❑ The way in which cases are managed is known as the case-based reasoning cycle. This CBR cycle consists of four sequential steps, namely: retrieval phase, reuse phase, revise phase and retain phase.

Multiagent System - Classification Mechanism

Design of the Mechanism of Classification



- Filter-Traffic agent
- SQL-Analyzer agent
- CBRid4SQL-Classifer and Visualizer agent

Classification Mechanism

CBRid4SQL agent

New Case – SQL Query:

```

“Select
field1, field2, field3 from table1
where
field1 =” or 9876 = 9876 -- ‘and field2=” ”
    
```

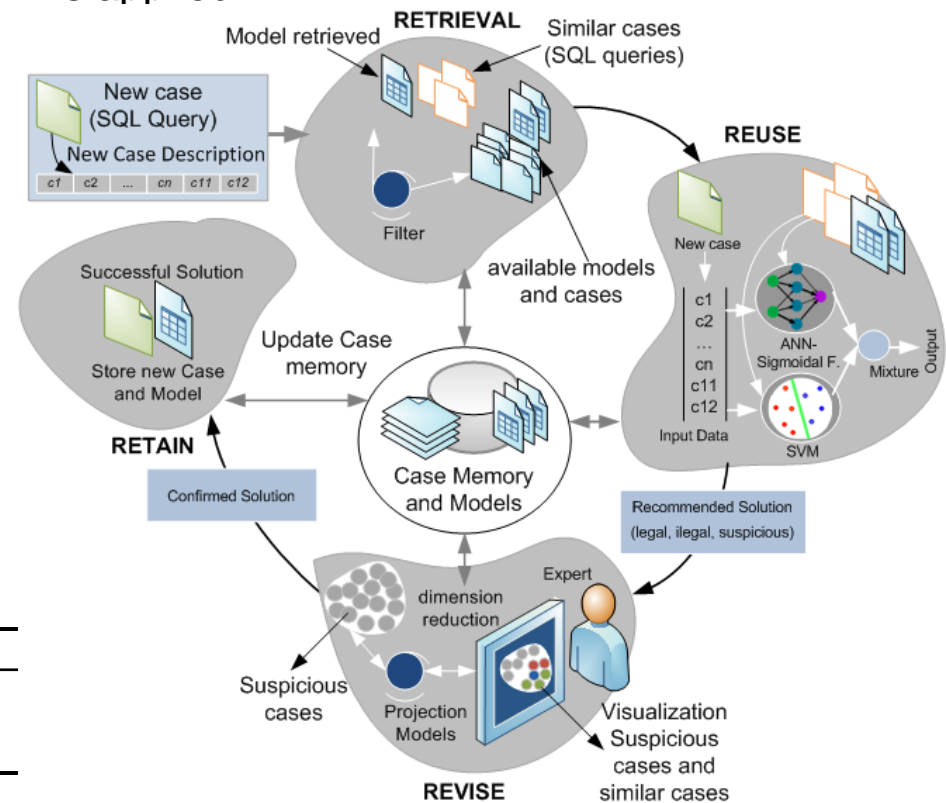
Case: Structure of the problem definition

- Affected_table^(c1)
- Affected_field^(c2)
- Command_type^(c3)
- Word_GroupBy^(c4)
- Word_Having^(c5)
- Word_OrderBy^(c6)
- Numer_And^(c7)
- Numer_Or^(c8)
- Number_literals^(c9)
- Length_SQL_String^(c10)
- Number_LOL^(c11)
- Cost_Time_CPU^(c12)
- Query_Category^(c13)

Fields - SQL String transformed through the string analysis

c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13
1	3	0	0	0	0	1	1	2	81	1	2,91	0

The first phase of the CBR cycle consists of recovering past experiences from memory of cases. A cosine similarity-based algorithm is applied.

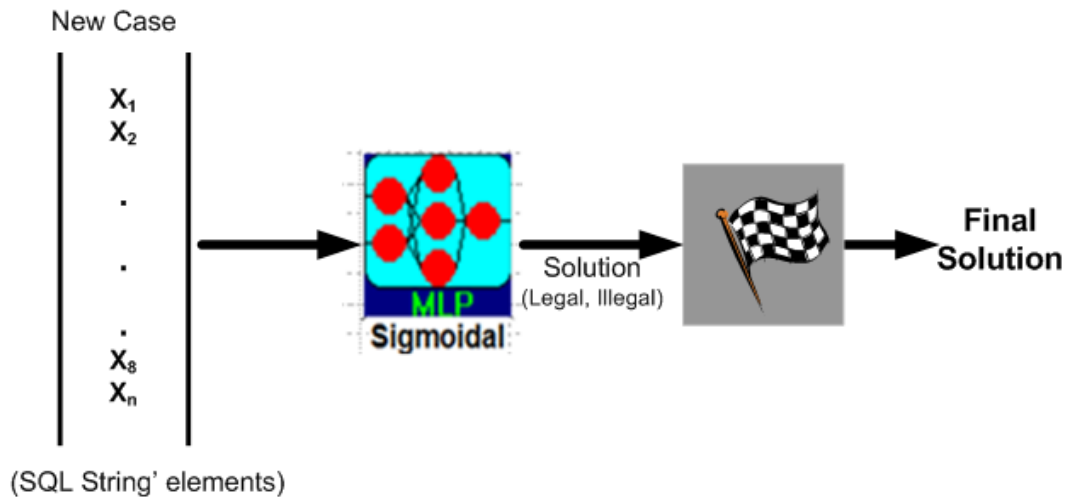


Classification Mechanism

Reuse - MLP

Classification by using a Neural Network

- Sigmoidal activation function



Sigmoidal activation function [0,2,0,8] (0,2 = Non-attack and 0,8 = attack)

In the reuse phase the network is trained by a back-propagation algorithm for the set of available training patterns.

The number of neurons in the output layer for the Multilayer Perceptron is one. It is responsible for deciding whether or not there is an attack. The sigmoidal activation function is given by:

$$f(x) = \frac{1}{1 + e^{-ax}}$$

Classification Mechanism

Reuse - SVM

Classification by using a Support Vector Machine

Polynomial Kernel function

$$class(x_k) = sign\left(\sum_{i=1}^m \lambda_i y_i \Phi(x_i) \Phi(x_k) + b\right)$$

Once the output values for the ANN and the SVM are obtained, the mixture is performed by way of a weighted average in function of the error rate of each one of the techniques.

Before carrying out the average, the values are normalized to the interval $[0,1]$, as SVM provides positive and negative values and those of greater magnitude, so that it could affect the final value in greater measure if it is not redimensioned.

In the reuse phase the SVM allows the separation of element classes which are linearly separable.

The selected kernel function in this problem was polynomial. The values used for the estimation are dominated by decision values and are related to the distance from the points to the hyperplane.

Classification Mechanism

Revise Phase

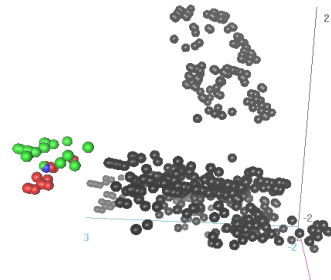
Selection of similar cases

Is carried out through the use of a neuronal GCS network, the different cases are distributed in meshes and the mesh in which the new case is found is selected.

Visualization of similar cases

The dimensionality of data is reduced by means of the CMLHL neuronal model which performs Exploratory Projection Pursuit by unsupervised learning.

Finally, the information is represented and the associated queries are recovered with the retrieved mesh



For cases detected as suspicious, with output values determined experimentally in the interval $[0.35, 0.6]$, a review by a human expert is performed

The review consists of recovering queries similar to the current one together with previous classifications. This combines a clustering technique for the selection of similar requests with a neuronal model for the reduction of dimensionality, which permits visualisation in 2D or 3D.

Results

A series of tests were elaborated to verify the proposed model. These tests were executed on a memory of cases developed with synthetic data: 705 previously classified queries (437 legal, 268 attacks).

Method		Method		Method	
BayesNet	638	Naive Bayes	666	AdaBoostM1	665
Bagging	684	DecisionStump	598	J48	689
JRIP	692	LMT	693	Logistic	688
LogitBoost	680	MultiBoostAB	666	OneR	622
SMO	685	Stacking	437	CBRid4SQL	698

The number of queries detected as suspicious was limited to 7 being one of those :

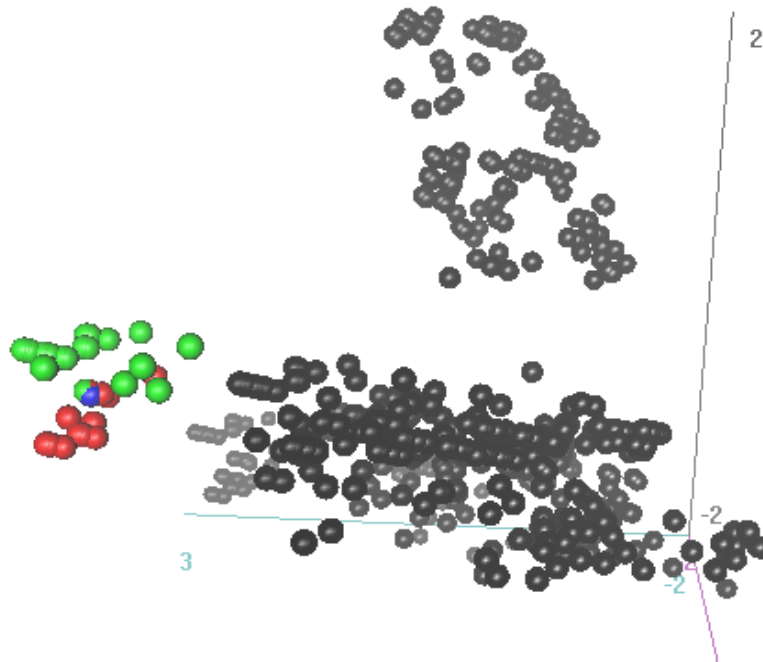
```
select pedido_cliente.id_pedido, linea, codigo, nombre, precio from pedido_lineas, pedido_cliente, producto where pedido_cliente.id_pedido = pedido_lineas.id_pedido and producto.codigo = pedido_lineas.codigo and pedido_cliente.id_pedido = 1 OR 1 = 1 order by fecha desc
```

This query represents an attack on the database since the presence of OR 1=1 implies the retrieval of a number of records not associated with requests from the client. The value obtained by the ANN for this query was 0.28. However SVM deemed that the output value was 0.66. The mixture gave an output value of 0.47, which is in the range of suspicious queries.

Results

During the manual review similar queries are recovered and dimensionality is reduced. The obtained results to be shown to the human expert can be seen. The most similar queries are coloured:

- Queries that correspond as legal are shown in green.
- Attacks are in red and current queries are in blue.
- Non-recovered queries are shown in black.



Conclusions

- The combination of different paradigms of AI allows the development of a HAIS with characteristics such as the capacity for learning and reasoning, flexibility and robustness which make the detection of SQL injection attacks possible.
- The proposed CBRid4SQL agent is capable of detecting these abnormal situations with low error rates compared with other existing techniques.
- It also provides a decision mechanism which eases the review of suspicious queries through the selection of similar queries and their visualization using neuronal models.



VNiVERSiDAD
D SALAMANCA



<http://bisite.usal.es>

CBRid4SQL: A CBR Intrusion Detector for SQL Injection Attacks

Authors: Cristian Pinzón
Álvaro Herrero
Juan F. De Paz
Emilio Corchado
Javier Bajo

HAIS'10