Prototipo de un sistema multi-agente para la detección de ataques por inyección SQL

Prototype of a multi-agent system for the detection of injection attacks SQL

Luis Felipe Wanumen Silva *
Gloria Andrea Cavanzo Nisso **
DJuan Carlos Guevara Bolaños ***

Fecha de recepción: marzo 17 de 2013
Fecha de aprobación: Abril 30 de 2013

Resumen

La seguridad en las bases de datos es uno de los elementos neurálgicos dentro de las organizaciones, ya que requieren diferentes estrategias de protección para salvaguardar puntos vulnerables, y de la coordinación y distribución de esfuerzos para cercar todas las posibles formas de ataques informáticos, en el caso del presente documento se hace referencia a los ataques por inyección SQL que son una de las más graves amenazas en la seguridad de las bases de datos, ya que permiten que cualquier persona pueda tener control sobre esta poniendo en peligro la confidencialidad y la integridad de la información.

En este documento se propone un sistema multiagente para la detección de ataques por inyección SQL desarrollado bajo las tecnologías Java, plataforma multiagente BESA y el manejo de bases de datos SQL Server 2008.

* Docente, Facultad Tecnológica de la Universidad Distrital Francisco José de Caldas. lwanumen@udistrital.edu.co

** Docente, Facultad Tecnológica de la Universidad Distrital Francisco José de Caldas. gacavanzon@udistrital.edu.co

*** Docente, Facultad Tecnológica de la Universidad Distrital Francisco José de Caldas. jcguevarab@udistrital.edu.co

El sistema propuesto está conformado por un grupo de agentes reactivos con características particulares que observan el entorno en el que se encuentran y extraen información del mismo.

Palabras clave

Sistemas multiagente, inyecciones SQL.

Abstract

Security in databases is one of the nerve within organizations because they require different protection strategies to safeguard vulnerable points, and the coordination and distribution of efforts to encircle all possible forms of cyber attacks in the if this document makes reference to SQL injection attacks are one of the most serious threats to the security of databases, allowing anyone to have control over is endangering the confidentiality and integrity of information.

This paper proposes a multi-agent system for detecting SQL injection attacks developed under Java technologies, BESA multiagent platform and database management SQL Server 2008.

The proposed system consists of a set of particular characteristics reactive agents that observe the environment in which they find and extract information from it.

Keywords

Multi-agent systems, SQL injections

[**Security**]: La seguridad se define en este artículo como la capacidad de un sistema de bases de datos de no permitir que usuarios no autorizados hagan operaciones no autorizadas de ejecución de instrucciones SQL que vulneren el acceso no permitido a los datos o que alteren en forma anormal la integridad de la información.

1. Introdución

Los ataques por inyección SQL, han sido uno de los ataques que han cobrado atención, debido a que no son fáciles de percibir fácilmente por los sistemas tradicionales de detección [1]. La prevención de ataques a servidores de bases de datos es apropiada hacerla con una aproximación de agentes [2], no solo porque permitirían una escalabilidad del sistema en entornos de bases de datos distribuidas [3], sino también a sistemas de bases de datos donde se haga necesario consultar los datos modificados en las bases de datos [4]. Este enfoque de agentes no sólo ha sido usado y probado en ambientes de bases de datos relacionales [5], sino en administración de bases de datos [6]. En todos los

casos se ha tenido un fuerte grado de ventaja el uso de agentes gracias a la capacidad de colaboración que ofrece ésta tecnología, la cual la hace susceptible de mejorar, incluso hacia sistemas automáticos de decisión [7]. Para el análisis de datos, se usa el diseño experimental [8], y se comprueba que efectivamente el SMA desarrollado es capaz de detectar intrusos de forma más acertada que con otras arquitecturas. Sin embargo, cuando el SMA se ejecuta durante mucho tiempo, se presentan degradaciones en el funcionamiento del SMA, que deben ser tenidas en trabajos futuros.

2. Metodología AOPOA (aproximación organizacional para programación orientada a agentes)

Esta metodología fue creada por el grupo de Investigación SIDRe de la Pontificia Universidad Javeriana de Bogotá y existe como una metodología global para desarrollo orientado a agentes, además cuenta con un "framework" de desarrollo de agentes llamado BESA [9] que se convierte en la herramienta de la fase de implementación. El enfoque de AOPOA [10] es similar al propuesto por GAIA, donde los SMA son vistos como organizaciones compuestas por niveles jerárquicos determinados por los esquemas de dependencia y comunicación entre las entidades que componen dichos niveles. Estas entidades son Agentes con arquitecturas internas definidas, cada Agente es un ente social el cual tiene asociadas un conjunto de habilidades, recursos y tareas que le permiten cumplir sus objetivos específicos, los cuales contribuyen al cumplimiento de los objetivos generales del sistema. Estas habilidades son el conjunto de facultades que tiene asociado un rol para cumplir sus objetivos, los recursos son entidades que permiten a los roles ejecutar sus tareas, y las tareas son actividades acordes con las habilidades del rol brindando consistencia en la arquitectura interna de él mismo. Para este fin los agentes exhiben comportamientos cooperativos que deben ser establecidos en algún punto dentro de la metodología [11].

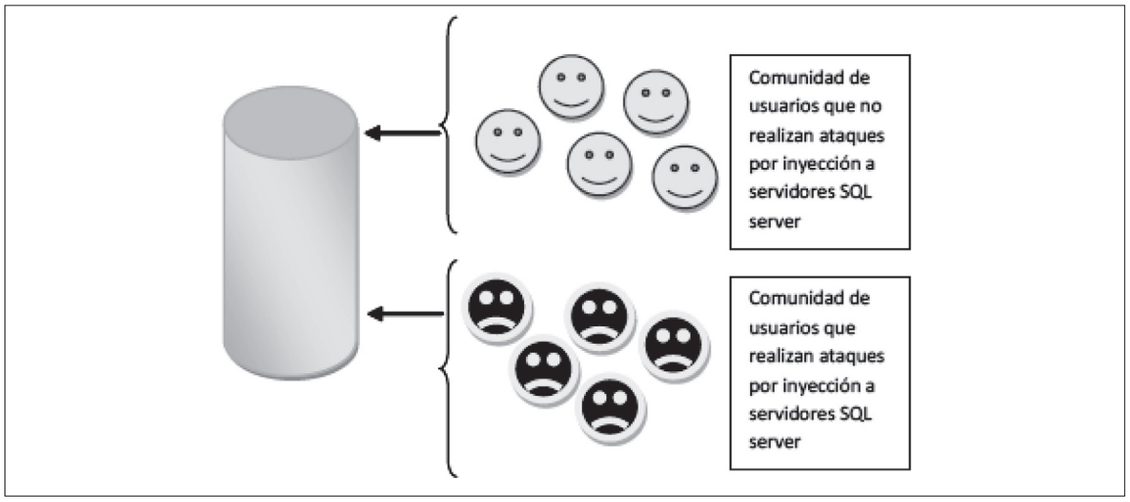
La construcción de estos sistemas es iterativa y desde el punto de vista de AOPOA [12] se centra en dos procesos principales: i) la descomposición organizacional y ii) el establecimiento de los vínculos cooperativos (dependencia o interacción entre varios roles, según AOPOA puede ser Colaboración Simple, Coordinada, Independencia, Obstrucción o Competencia). A continuación se describe brevemente el proceso metodológico definido en AOPOA.

- Fase de análisis: esta es la primera fase contemplada por AOPOA y para iniciarla se exigen dos prerrequisitos: la definición previa de los requerimientos del Sistema y el estudio de factibilidad del SMA. En esta parte de análisis se define la estructura organizacional que se tuvo en cuenta a la hora de implementar el sistema multiagente, mostrando la actividad y las metas de cada organización.

- Fase de Diseño: dentro de esta fase, AOPOA contempla tres pilares básicos que son: La definición y detalle de los Vínculos Cooperativos entre los roles, La definición completa de la Estrategia de Cooperación que se seguirá en el SMA y el Diseño de los Meta-Agentes de los cuales se "instanciaran" los Agentes finales del SMA.

- Fase de despliegue: esta es la fase final de la metodología AOPOA, dentro de esta se busca mapear de forma real la arquitectura del SMA que se ha definido en las etapas previamente descritas. Para este fin dentro de esta fase se contemplan actividades como la definición de los sitios de despliegue, definidos como contenedores o entornos lógicos en los cuales se situaran las diversas instancias que se creen de cada meta-agente definido en el diseño, respecto a estos sitios se define la asignación de recursos y procesamiento asociados para finalmente definir y crear las instancias de Agentes que poblaran estos contenedores.

Figura 1. Definición de comunidades



Fuente: elaboración propia.

3. Fase de análisis

Un aspecto inicial en la metodología AOPOA es la descomposición organizacional [13]

3.1 Comunidades

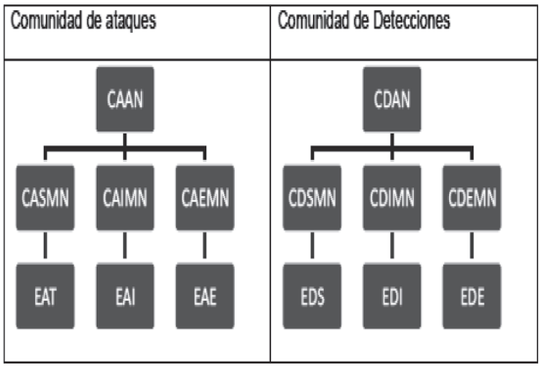
Al analizar el entorno organizacional tenemos una situación interesante con respecto a los usuarios que acceden a servidores de bases de datos. Se han identificado dos tipos de comunidades: Comunidades que realizan ataques y comunidades que no realizan ataques (figura 1).

3.2 Descomposición organizacional

En la siguiente figura se muestran los mnemotécnicos de cada uno de los roles, al interior de las dos comunidades, tanto de ataques como de detecciones. En la tabla de roles por agente se detallan las descripciones de las siglas de la figura 2.

Como no todos los ataques prosperan en la base de datos, es que se puede asegurar que existen muchas consultas que son tomadas como consultas de selección, inserción y eliminación son normales. Este es precisamente el aspecto interesante, que no se hace necesario crear una comunidad para la ejecución normal de consultas, ya que de hecho cuando se planean ataques, son más las consultas que no hacen daño, que las consultas que hacen daño.

Figura 2. Descomposición organizacional



Fuente: elaboración propia.

Las responsabilidades de cada uno de los agentes de la organización en relación con los roles que pueden tomar, se muestra en la tabla siguiente:

Tabla 1. Roles por agente

Roles que puede tomar el agente	Agente
CAAN coordinador de ataque de alto nivel	Chief High Atack: este agente es el jefe de la organización de los procesos de ataque al sistema y es el encargado de contratar monitorear y controlar a los agentes que se encuentran en el nivel medio de la organización
CDAN coordinador de detecciones de alto nivel	Chief High Detection: este agente es el jefe de la organización de los procesos de detección de los ataques que pueda sufrir el sistema y es el encargado de contratar, monitorear y controlar a los agentes que se encuentran en el nivel medio de la organización
CDSMN coordinador de detecciones de selección de mediano nivel CDIMN coordinador de detecciones de inserción de mediano nivel CDEMN coordinador de detecciones de eliminaciones de mediano nivel	Chief Middleware Detection: son los designados para contratar, monitorear y controlar a los agentes del nivel bajo de la organización encargados de detectar los posibles ataques que están ingresando al sistema
CASMN coordinador de ataques de elección de mediano nivel CAIMN coordinador de ataque de inserción de mediano nivel CAEMN coordinador de ataques de eliminación de mediano nivel	Chief Middleware Atack: son los encargados de contratar, monitorear y controlar a los agentes que ejecutarán los ataques pertenecientes al nivel bajo de la organización
EDS ejecutor de detecciones de selección EDI ejecutor de detecciones de inserción EDE ejecutor de detecciones de eliminación	Agent Detection: se encargan de detectar los posibles ataques que están ingresando en el sistema
EAT ejecutor de ataques de selección EAI ejecutor de ataques de inserción EAE ejecutor de ataques de eliminación	Agent Atack: se encargan de realizar las consultas SQL para atacar el sistema

Fuente: elaboración propia.

4. Fase de diseño

En el diseño basado en AOPOA, definir los recursos y las habilidades es pieza fundamental para implementar un SMA con esta aproximación [14].

4.1 Diagrama de habilidades

Tabla 2. Habilidades por agente

Agente	Habilidades	Recursos
Chief high atack	Saber contratar a los agentes medios atacantes. Recibir respuestas de contratación. Tener la capacidad de dar la orden de empezar a atacar el sistema. Saber leer informes de los resultados obtenidos de los ataques.	Número de atacantes por contratar. Archivo de configuración. Agentes activos. Variables que almacenen la información enviada por los agentes.

Agente	Habilidades	Recursos
Chief middleware atack	Recibir petición de contratación. Saber enviar una repuesta adecuada acerca de la contratación. Saber contratar a los agentes de nivel bajo que van a atacar. Recibir respuestas de contratación de agentes bajos. Saber recibir la orden del jefe de iniciar ataques. Poder informar a los agentes de bajo nivel iniciar los ataques. Saber recibir la información de los resultados enviados por parte de los agentes bajos. Poder enviar informes de resultados al jefe mayor.	Variable de estado. Tener activa la variable Value Número de agentes bajos por contratar. Variables de lectura. Variables de escritura. Variables activas de recepción de información. Métodos para enviar informes activos.
Agent atack	Recibir petición de contratación. Saber enviar una respuesta adecuada acerca de la contratación. Poder recibir órdenes de iniciar ataques. Atacar el sistema. Obtener resultados de los ataques ejecutados. Realizar informe de los resultados y enviarlos al jefe medio.	Variable de estado activa. Variable value activa. Variable position configurada. Variable Exce. Archivo de configuración. Variables de escritura. Variables de lectura. Método enviar.
Chief high detection	Saber contratar a los agentes medios detectores. Recibir respuestas de contratación. Tener la capacidad de dar la orden de empezar a defender el sistema. Saber leer informes de los resultados obtenidos de las detecciones hechas durante el tiempo que dura la simulación.	Número de agentes atacantes por contratar. Archivo de configuración. Agentes activos. Variables que almacenen la información enviada por los agentes.
Chief middleware detection	Recibir petición de contratación para ser agente medio detector. Saber enviar una respuesta adecuada acerca de la contratación. Saber contratar a los agentes de nivel bajo que van a defender el sistema. Recibir respuestas de contratación de agentes bajos. Saber recibir la orden del jefe de empezar a defender el sistema. Poder informar a los agentes de bajo nivel iniciar las detecciones de los posibles ataques que se presenten. Saber recibir la información de los resultados acerca de las detecciones hechas y enviadas por los agentes bajos. Poder enviar los informes de resultados al jefe mayor.	Variable de estado. Tener activa la variable value: -Número de agentes bajos por contratar. -Variables de lectura. -Variables de escritura. -Variables activas de recepción de información. -Métodos para enviar informes activos.
Agent detection	Recibir petición de contratación para ser agente de detección. Saber enviar una respuesta adecuada acerca de la contratación. Poder recibir órdenes de empezar a detectar. Poder detectar los posibles ataques que están ingresando al sistema. Obtener resultados de las detecciones hechas. Realizar informes de resultados y enviarlos al jefe medio.	Variable de estado activa. Variable value activa. Variable Exce. Archivo de configuración. Variables de escritura. Variables de lectura. Método para enviar.

Fuente: elaboración propia.

Tabla 3. Tareas y metas para el agente mayor de detección

Id	Tarea	Descripción de la tarea	Meta	Agente
1	Contratar agentes coordinadores de detección	Buscar en la organización los agentes que pertenezcan al nivel intermedio y enviarles una solicitud de contratación	Detectar ataques	Agente jefe mayor de detección
2	Recibir confirmación de contratación del agente de eliminación coordinador	Recibir de los agentes intermedios de eliminación su aprobación su aprobación con respecto a la solicitud de contrato hecho por el agente mayor de detección	Detectar ataques	Agente jefe mayor de detección
3	Recibir confirmación de contratación del agente de inserción coordinador	Recibir de los agente intermedios de inserción su aprobación con respecto a la solicitud de contrato hecho por el agente mayor de detección	Detectar ataques	Agente jefe mayor de detección
4	Recibir confirmación de contratación del agente de selección coordinador	Recibir de los agente intermedios de selección su aprobación con respecto a la solicitud de contrato hecho por el agente mayor de detección	Detectar ataques	Agente jefe mayor de detección
5	Iniciar detecciones	Buscar en la organización agentes que pertenezcan al nivel intermedio y enviarles una orden de iniciar detecciones	Detectar ataques	Agente jefe mayor de detección

Fuente: elaboración propia.

Tabla 4. Tareas y metas para el agente mayor de ataque

Id	Tarea	Decripción de la tarea	Meta	Agente
6	Contratar agentes coordinadores de ataque	Buscar en la organización los agentes que pertenezcan al nivel intermedio y enviarles una solicitud de contratación	Realizar ataques	Agente jefe mayor de ataque
7	Recibir confirmación de contratación del agente de eliminación coordinador	Recibir de los agentes intermedios de eliminación su aprobación con respecto a la solicitud de contrato hecho por el agente mayor de ataque	Realizar ataques	Agente jefe mayor de ataque

Id	Tarea	Decripción de la tarea	Meta	Agente
8	Recibir confirmación de contratación del agente de inserción coordinador	Recibir de los agentes intermedios de inserción su aprobación con respecto a la solicitud de contrato hecho por el agente mayor de ataque	Realizar ataques	Agente jefe mayor de ataque
9	Recibir confirmación de contratación del agente de selección coordinador	Recibir de los agentes intermedios de selección su aprobación con respecto a la solicitud de contrato hecho por el agente mayor de ataque	Realizar ataques	Agente jefe mayor de ataque
10	Iniciar ataques	Buscar en la organización los agentes que pertenezcan al nivel intermedio y enviarles una orden de iniciar ataques	Realizar ataques	Agente jefe mayor de ataque

Fuente: elaboración propia.

Tabla 5. Tareas y metas del agente intermedio de ataque

Id	Tarea	Descripción de la tarea	Meta	Agente
11	Aceptar la solicitud de contratación del jefe mayor de ataque	Enviar al agente mayor la aceptación acerca de la solicitud de contratación	Realizar ataques	Agente de nivel medio coordinador de ataques tipo select
12	Aceptar la solicitud de contratación del jefe mayor de ataque	Enviar al agente mayor la aceptación acerca de la solicitud de contratación	Realizar ataques	Agente de nivel medio coordinador de ataques tipo insert
13	Aceptar la solicitud de contratación del jefe mayor de ataque	Enviar al agente mayor la aceptación acerca de la solicitud de contratación	Realizar ataques	Agente de nivel medio coordinador de ataques tipo delete
14	Contratar agentes bajos de ataque tipo select	Buscar en la organización los agentes que pertenezcan al nivel bajo y enviarles una solicitud de contratación	Realizar ataques	Agente de nivel medio coordinador de ataques tipo insert
15	Contratar agentes bajos de ataque tipo inset	Buscar en la organización los agentes que pertenezcan al nivel bajo y enviarles una solicitud de contratación	Realizar ataques	Agente de nivel medio coordinador de ataques tipo insert

Id	Tarea	Descripción de la tarea	Meta	Agente
16 tipo	Contratar agentes bajos de ataque tipo delete	Buscar en la organización los agentes que pertenezcan al nivel bajo y enviarles una solicitud de contratación	Realizar ataques	Agente de nivel medio coordinador de ataques tipo delete
17	Recibir confirmación de contratación del agente de eliminación coordinador	Recibir de los agentes bajos de eliminación su aprobación con respecto a la solicitud de contrato hecho por el agente mayor de ataque	Realizar ataques	Agente de nivel medio coordinador de ataques tipo delete
18	Recibir confirmación de contratación del agente de inserción coordinador	Recibir de los agentes bajos de eliminación su aprobación con respecto a la solicitud de contrato hecho por el agente mayor de ataque	Realizar ataques	Agente de nivel medio coordinador de ataques tipo insert
19	Recibir confirmación de contratación del agente de selección coordinador	Recibir de los agentes bajos de eliminación su aprobación con respecto a la solicitud de contrato hecho por el agente mayor de ataque	Realizar ataques	Agente de nivel medio coordinador de ataques tipo select

Fuente: elaboración propia.

Tabla 6. Técnicas de ataque

Técnica analizada	Descripción de la técnica	Razones para su inclusión o no en el SMA propuesto
Variante 1: variante de inyección SQL basada en creación dinámica de caracteres de escape	Esta técnica usa algoritmos de fuerza bruta, en los que se establecen diversas posibilidades de combinación de caracteres especiales de escape. Estos caracteres se incorporan aleatoriamente en consultas SQL con el fin de lograr registros de log erróneos en el servidor de base de datos	Esta técnica no fue escogida a pesar de que su algoritmo es sencillo de implementar. El problema radica en que dado que el número de posibilidades necesarias para realizar una inyección es exponencialmente. Para el caso de construir con este método las sentencias de inyección se tendrían que realizar grandes cantidades de operaciones y esto disminuiría el rendimiento del sistema
Variante 2: variante de inyección SQL basada en creación aleatoria de caracteres especiales	Esta técnica es un poco más avanzada que la anterior; presenta grandes ventajas, por cuanto el tipo de ataques que realiza es de mayor impacto. El problema radica en que este tipo de ataques es muy fácil de detectar: basta con tener una tabla de caracteres especiales e impedir su ingreso en el sistema	No se escogió esta técnica debido a que al implementar los algoritmos de detección, muy seguramente todas las sentencias arrojadas con este método de inyección serían detectadas y no se podría llegar a tener un gran impacto en las conclusiones del presente documento

Técnica analizada	Descripción de la técnica	Razones para su inclusión o no en el SMA propuesto
Variante 3: variante de inyección SQL basada en ensamblado de consultas	Esta técnica se basa en la experiencia obtenida por otros programadores, en la cual se tiene previamente un listado de consultas que han funcionado en algunos motores de bases de datos y han provocado ataques en dichos sistemas. Se basa en la selección a priori de consultas potencialmente peligrosas por su ubicación estratégica de las comillas, en la cual se engaña al servidor remoto lo que hacer que las sentencias where predominantemente se vean engañadas por suplantaciones o validaciones que siempre y forzosamente serán válidas. Es útil la aplicación de esta técnica en desarrolladores con alta experiencia en desarrollo de sistemas que tienen a su alcance una gran cantidad de posibles sentencias que han sido probadas en otros sistemas	Dado el avance de la informática y el conocimiento sobre sistemas de bases de datos como SQL server, se querían probar sentencias que en otros sistemas habrían sido fatales y habían provocado su colapso. Se tiene amplia documentación sobre la forma como se ataca, pero es poca la documentación sobre cómo prevenir este tipo de consultas. Por otra parte, estas consultas son las que en su mayoría causan de ataques a servidores de bases de datos. Por lo anteriormente descrito esta técnica fue la escogida para el desarrollo del sistema

Fuente: elaboración propia.

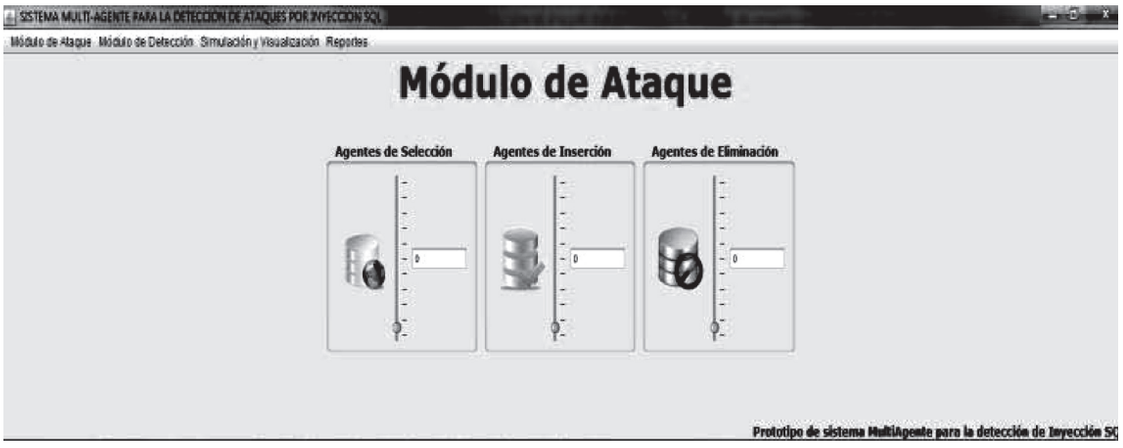
5. Fase de implantación y despliegue

En el módulo de ataque se pueden seleccionar el número de agentes que van a realizar ataques de selección, de inserción y de eliminación.

Teniendo en cuenta los elementos por utilizar se establece la funcionalidad de cada uno, cuales son:

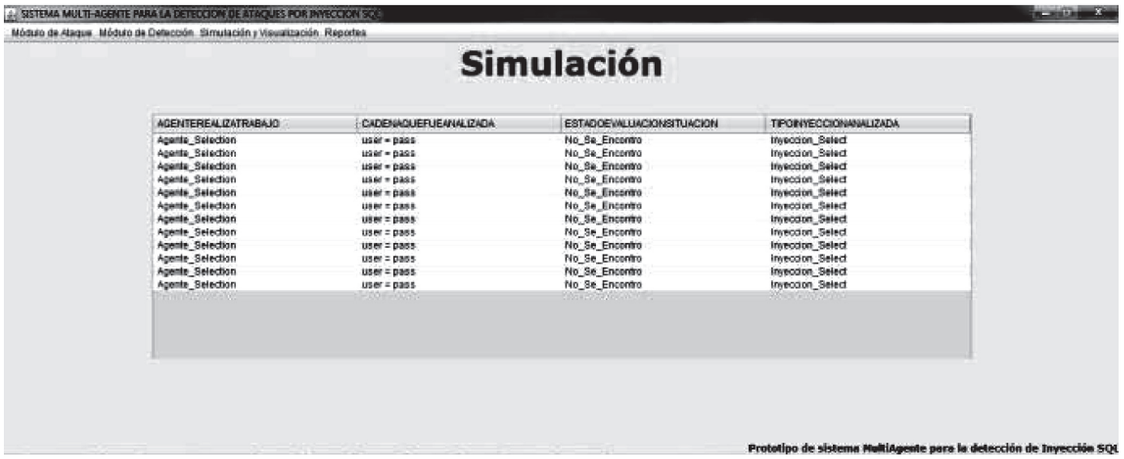
- *Agentes de selección*: en esta parte se selecciona el número de agentes que van a detectar ataques de tipo selección.

Figura 3. Módulo de ataque del SMA



Fuente: elaboración propia.

Figura 4. Módulo de visualización de resultado



Fuente: elaboración propia.

- *Agentes de inserción*: en esta parte se selecciona el número de agentes que van a detectar ataques de tipo inserción.

- *Agentes de eliminación:* en esta parte se selecciona el número de agentes que van a detectar ataques de tipo eliminación.

En este modulo se determina el tiempo que durara la simulación y luego con el botón "Iniciar" se da paso a la visualización dinámica de los datos que está arrojando el sistema.

En este módulo se determina el tiempo que durara la simulación y luego con el botón "Iniciar" se da paso a la visualización dinámica de los datos que está arrojando el sistema.

Se espera que en todo momento el número de ataques detectados sea inferior o igual al número de ataques realizados.

Teniendo en cuenta las siguientes variables:

- NAAS = número de agentes de ataques de selección

- NAAI = número de agentes de ataques de inserción

- NAAD = número de agentes de ataques de eliminación

- NADS = número de agentes de detección de selección

- NADI = número de agentes de detección de inserción

- NADD = número de agentes de detección de eliminación T = tiempo que dura la simulación

- ARAS = ataques realizados por agentes de ataques de selección

- ARAI = ataques realizados por agentes de ataques de inserción

- ARAD = ataques realizados por agentes de ataques de eliminación

- ADDS = ataques detectados por agentes de detección de selección

- ADDI = ataques detectados agentes de detección de inserción

- ADDD = ataques detectados agentes de detección de eliminación

- E = Efectividad del sistema

Se calculan unas matrices de confusión para observar el comportamiento del sistema para 2 escenarios

Matriz de confusión para el escenario 1: para la elaboración de ésta matriz se parte de las condiciones del escenario, descritas en la siguiente tabla

Tabla 7. Condiciones de entrada escenario 1

AAS	AAI	AAD	ADS	ADI	ADD	TE
3	3	3	4	4	4	15

Fuente: elaboración propia.

Matriz de confusión obtenida:

Tabla 8: Matriz de confusión escenario 1

	ND (select)	ND (insert)	ND (delete)	Total
ND (select)	8			8
ND (insert)		7		8
ND (delete)			5	6

Fuente: elaboración propia.

La anterior matriz muestra que de 8 ataques de selección, se detectaron 8, es decir el 100 % de los ataques fueron detectados por el sistema. De 8 ataques de inserción, 7 fueron detectados como ataques de inserción, es decir el 87 % de los ataques de inserción fueron detectados. Finalmente de 6 ataques de eliminación 5 fueron detectados, es decir el 83% de los ataques de eliminación fueron detectados

Matriz de confusión para el escenario 2

Condiciones de entrada del escenario:

Tabla 9. Condiciones de entrada escenario 2

AAS	AAI	AAD	ADS	ADI	ADD	TE
4	4	4	5	5	5	15

Fuente: elaboración propia.

Matriz de confusión obtenida:

Tabla 10. Matriz de confusión escenario 2

	ND (select)	ND (insert)	ND (delete)	Total
ND (select)	9			9
ND (insert)		9		10
ND (delete)			5	7

Fuente: elaboración propia.

La anterior matriz muestra que de 9 ataques de selección, se detectaron 9, es decir el 100 % de los ataques fueron detectados por el sistema. De 10 ataques de inserción, 9 fueron detectados como ataques de inserción, es decir el 90 % de los ataques de inserción fueron detectados. Finalmente de 7 ataques de eliminación 5 fueron detectados, es decir el 71 % de los ataques de eliminación fueron detectados.

6. Agradecimientos

Agradecimientos a la Universidad Distrital por su apoyo en la realización de este artículo y a la organización de CCOM por su convocatoria, que ha inspirado la elaboración del presente artículo.

7. Referencias

[1] C. Pinzón, J. F. De Paz, J. Bajo, A. Herrero y E. Corchado, "AIIDA-SQL: An adaptive Intelligent Intrution Detector Agent for Detecting SQL Injection Atacks", *Hybrid Intelligent Systems (HIS) 2010 10th International Conference on , pp.73-78, 23-25 Aug. 2010*

[2] T. M, Ahmed, "Using SMAC Agent Securely in Heterogonous Database Systems", Information, Communications and Signal Processing, 2005 Fifth International Conference on , pp.1264-1268.

[3] Hong Yu; Shao-Zhong Zhang; Nan-Hai Yang; Hua Ding; Xiu-kun Wang , "Intelligent agent-based distributed heterogeneous database system", Machine Learning and Cybernetics, 2003 International Conference on , vol.3, no., pp. 1932- 1935 Vol.3, 2-5 Nov. 2003

[4] Hongbin Zhang; Guixia Yang; Gang Li; Baorui Chen; Xin Xiaoping, "An intelligent inter database retrieval system based on Multi-agent", World Automation Congress (WAC), 2010 , vol., no., pp.101-105, 19-23 Sept. 2010

[5] I. Rudowsky, O. Kulyba, M. Kunin, D. Ogarodnikov, T. Raphan, "Managing a relational database with intelligent agents", Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on , vol.1, no., pp. 238-242. April 2005

[6] S. Ramanujam, S.; Capretz, M.A.M.; , "Design of a multi-agent system for autonomous database administration," Electrical and Computer Engineering, 2004. Canadian Conference on , vol.2, no., pp. 1167- 1170 Vol.2, 2-5 May 2004

[7] Tian-Lei Hu; Gang Chen; Yin-Jie Hong; Xiao-Long Zhang; Jin-xiang Dong; , "Collaborative Agents Supported Automatic Physical Database Design Based on Description Logics Reasoning," Computer Supported Cooperative Work in Design, 2006. CSCWD '06. 10th International Conference on , vol., no., pp.1-6, 3-5 May 2006

[8] Experimental Design and Analysis. Howard J. Seltman. January 30, 2012

[9] E. González, C. Bustacara and Ávila, J. (2003). BESA: Behavior-oriented, Event-driven and Social-based Agent ramework TA '03

[10] González E., Torres M., Rodríguez J. La Metodología AOPOA. . Grupo SIDRe, Departamento de Ingeniería de Sistemas, Facultad de Ingeniería, Pontificia Universidad Javeriana

[11] Gonzalez E., Torres M., AOPOA Organizational Approach for Agent Oriented Programming, in Proceedings of the 8th International Conference on Enterprise Information Systems, May 2006 Paphos – Cyprus

[12] González E., Bustacara C. ,Desarrollo de Aplicaciones Basadas en Sistemas MultiAgentes. Libro Producto de Investigación – Grupo SIDRe, Departamento de Ingeniería de Sistemas, Facultad de Ingeniería, Pontificia Universidad Javeriana.

[13] D. Ahogado, A. M. Reinemer. Gonzalez "AO OA: Aproximación Organizacional para Programación Orientada a Agentes" L I2003 La Paz, Bolivia. Septiembre de 2003.

[14] E. González y C. Bustacara. Metodologías OA y AO OA Desarrollo de aplicaciones basadas en sistemas MultiAgentes " (pp. 49-95), (pp.97- 167). Pontificia Universidad Javeriana, Bogotá, Colombia. Editorial Pontificia Universidad Javeriana.