FUSIÓN Y CORRELACIÓN DE ALERTAS BASADAS EN ONTOLOGÍAS SOBRE SISTEMAS MULTI-AGENTES

AN ONTOLOGY-BASED ALERT FUSION AND CORRELATION ON MULTIAGENT SYSTEMS

AUTOR
Ph.D GUSTAVO ISAZA ECHEVERRY
Doctor en Ingeniería Informática e Ingeniería del software
*Universidad de Caldas
Profesor Asociado
Grupo GITIR
Ci2Dt2 Facultad de Ingeniería
gustavo.isaza@ucaldas.edu.co
COLOMBIA

AUTOR
LUIS FERNANDO CASTILLO OSSA
Doctor en Informática y Automática
*Universidad de Caldas
Profesor Asociado
Grupo GITIR
Ci2Dt2 Facultad de Ingeniería
luis.castillo@ucaldas.edu.co
COLOMBIA

INSTITUCIÓN
*UNIVERSIDAD DE CALDAS
UCALDAS
Universidad Pública
Calle 65 Nº 26 – 10, Manizales
ucaldas@ucaldas.edu.co
COLOMBIA

INFORMACIÓN DE LA INVESTIGACIÓN O DEL PROYECTO: Modelo de supervisión y prevención de ataques informáticos basado en una arquitectura Honeynet Distribuida e Inteligente. 0574611 / 2011-2013

RECEPCIÓN: Febrero 19 de 2013

ACEPTACIÓN: Marzo 11 de 2013

TEMÁTICA: Gestión y Seguridad en Redes

TIPO DE ARTÍCULO: Artículo de Reflexión

RESUMEN ANALÍTICO

Los sistemas de detección de ataques o intrusiones evalúan tráfico a partir de un conjuntos de firmas predeterminadas para identificar posibles comportamientos anormales, sin embargo, este tipo de técnicas son insuficientes si la secuencia del evento no corresponde a alguno de los patrones previamente reconocidos. El uso de redes trampa/señuelo (honeynet) ha contribuido a identificar la taxonomía de los atacantes. Este artículo presenta una aproximación a un modelo de detección de ataques utilizando sistemas multi-agentes en modo señuelo que incorpora procesos de fusión y correlación de alertas sobre ontologías, conducentes a identificar comportamientos anómalos a partir de procesos de inferencia y razonamiento.

PALABRAS CLAVES: Ontologías, Fusión de Alertas, Correlación de Ataques, Sistemas Multi-agentes, Detección de Intrusiones Inteligente

ANALYTICAL SUMMARY

Attacks Detection systems evaluate traffic attacks or intrusions from a default signature sets to identify potential abnormal behaviors, however, these techniques are insufficient if the sequence of events does not correspond to any of the previously recognized patterns. The use of honeynet aims to identify the taxonomy of attackers. This papers presents an approach to attack detection model using multi-agent systems incorporating honeynet mode mergers and alert correlation on ontologies to identify abnormal behavior leading from inference and reasoning processes.

KEYWORDS: Ontologies, Fusion Alerts, Correlation Attacks, Multiagent Systems, Intelligent Intrusion Detection

INTRODUCCIÓN.

Los estándares y taxonomías que actualmente soportan los sistemas de detección de intrusiones están basados en modelos sintácticos y no semánticos. El uso de ontologías permite describir objetos, conceptos y relaciones en un dominio de conocimiento. La definición de ontologías de firmas y reglas de ataques en la detección de intrusiones permite minimizar las dificultades de representación y heterogeneidad del conocimiento en los sistemas distribuidos de detección de intrusiones.

Las ontologías especifican rigurosamente un esquema conceptual en un dominio, con el objetivo de comunicar, interactuar, intercambiar y compartir información entre diferentes recursos computacionales, es por lo tanto común encontrar en el ambiente de tecnologías el uso de técnicas de inteligencia artificial y representaciones formales del conocimiento para optimizar tareas de clasificación, y razonamiento inductivo para resolver diferentes problemas. Las ontologías representan formalmente especificaciones de conceptos que ofrecen un conocimiento compartido en un dominio definido sobre un lenguaje semántico. Una ontología está compuesta por conceptos (elementos básicos de los dominios, que se organizan en taxonomías), instancias (específicas de los conceptos), relaciones (entre los conceptos del dominio), funciones, axiomas. Las ontologías se han convertido en elementos fundamentales de los sistemas multi-agentes, ya que permiten la comunicación entre agentes heterogéneos y hace posible la creación, la transmisión y el almacenamiento del conocimiento.

En este trabajo se presentan los resultados de un proyecto que utiliza un modelo ontológico para representar la detección y prevención de intrusiones a través de una arquitectura basada en agentes y el uso de las ontologías para razonamiento, clasificación y reconocimiento de comportamiento de atacantes. Los elementos del artículo están organizados de la siguiente manera: Se presentan en la sección 1, los trabajos anteriores relevantes en este campo, en la sección

2, el sistema multi-agente y el modelo ontológico que representa las firmas ataques y las normas de reacción para apoyar el sistema de prevención, en la sección 3 se evidencian los resultados obtenidos. Finalmente resumimos nuestro trabajo de investigación y analizamos las contribuciones futuras.

1. ESTADO DEL ARTE.

En los últimos años diferentes organizaciones han dedicado grandes esfuerzos en el ámbito de la estandarización y normalización de los IDS (Intrusion Detection Systems) en proyectos como IDWG y CIDF.
En [1] se provee una Taxonomía de Ataques de Redes que incluye categorías para clasificar atacantes, herramientas, accesos, resultados y objetivos, en esta investigación los atacantes son Hackers, Espías, Terroristas o Adolescentes que buscan objetivos como ganancia financiera.

El uso de lenguajes para describir ataques ha sido explorado en proyectos como [2] que provee un marco STATL para categorizar Eventos, Respuestas, Reportes, Correlación y Reconocimiento. Una instancia de STATL permite describir un escenario de ataque que puede ser usado por un IDS para analizar un flujo de eventos y detectar posibles intrusiones. Este lenguaje provee una composición de los estados y transiciones de un ataque.

Otras iniciativas relevantes y sus aportes en el problema de detección de intrusiones como el IETF (Internet Engineering Task Force) creó un subgrupo de trabajo enfocado a la detección de intrusiones el IDWG (Intrusion Detection Working Group), sus principales objetivos eran:

- Componer un documento que determine los requisitos para la comunicación entre Sistemas de Detección de Ataques y sistemas de gestión
- Diseñar, definir e implementar una especificación de lenguaje de intrusiones común, que permita estandarizar y homogenizar el formato de los datos que cumplan con los requerimientos de un IDS
- Redactar un documento que describiera los protocolos para comunicar IDS y su relación con los formatos de especificación y representación.

Adicionalmente se desarrolló un modelo de intercambio de datos y mecanismos de transporte que le permiten a los IDS apropiar un sistema de mensajería para compartir información procedente de diferentes nodos. Este modelo es una representación orientada a objetos de los datos de alertas enviados a gestores de detección de intrusos por un analizador [3]. En este grupo se implementaron dos versiones del modelo IDMEF (Intrusion Detection Message Exchange Format) definido en la RFC4765, una basada en SMI (Structure of Management Información) para describir recursos MIB basados en el protocolo SNMP (Simple Network Management Protocol) y otra usando DTD (Document Type Definition) para describir documentos XML, se decidió que la versión basada en XML cumplía con más requerimientos y puede ser más escalable. El estándar define una representación basada en clases de cada componente y su respectiva especificación en XML. Uno de los principales problemas de IDMEF es el uso de XML, ya que es una representación sintáctica del modelo de datos, este diseño requiere que cada IDS interprete e implemente el modelo, así como definir su propio motor de análisis y razonamiento.

El uso de ontologías en seguridad informática y en particular en detección de intrusiones ha sido un campo hasta la fecha poco explorado, si bien hay unos esfuerzos importantes, se evidencia una gran oportunidad de generar nuevos aportes en esta línea, particularmente en la evolución de los nuevos tipos de ataque presentes en los modelos de las nuevas tecnologías Web y de la computación distribuida, así como en la incorporación de razonamientos semánticos e integración con otras tecnologías derivadas de la inteligencia computacional distribuida. A continuación se hará referencia a algunos proyectos relevantes y que pueden tener elementos semejantes a las representaciones ontológicas en el problema de detección de intrusiones.

En la iniciativa de [4] se creó SOID (Simple Ontology for Intrusion Detection) que permite razonar sobre un modelo de incidentes de alto nivel, pero no representa formalmente un modelo de las firmas, solo la estructura general de un escenario de intrusión.

Tal vez uno de los proyectos de más relevancia en este campo, es el esfuerzo del grupo de investigación de la Universidad de Maryland hecho en la investigación de [5] en una ontología centrada en el objetivo "target-centric" para detección de intrusiones, donde se demostró experimentalmente la migración de un sistema de clasificación taxonómico y sus representaciones sintácticas a un modelo ontológico y semántico, a partir de un modelo de representación de datos , algunas

relaciones e instancias de ataques e intrusiones con DAML-OIL (DARPA Agent Markup Language + Ontology Interface). El concepto "target-centric" se define como el concepto del punto de vista de la víctima que requiere la detección de intrusiones y que la información pueda ser medible y observable.

En [5] se propone una ontología para clasificar y representar un modelo de incidentes de seguridad basada en el usuario en un contexto de más alto nivel, tomando como referencia otros modelos genéricos.
En [6] se analizan diferentes tipos de ontologías y conocimientos particularmente en sistemas de seguridad informática y detección de intrusos y en [7] se presenta nuevo acercamiento para diseñar aplicaciones de detección de intrusos donde el dominio de los expertos está representado en ontologías para proveer razonamiento inteligente.

Investigaciones más recientes como [8] proponen un enfoque basado en ontologías para crear instancias de las políticas de seguridad para trazar descripciones de ataques que permitan identificar las políticas que deben aplicarse en la configuración de la red hacia la minimización de amenazas. Adicionalmente, el proyecto desarrollado por [9] describe el sistema de Scyllarus, que realiza la fusión de IDSs, con redes Bayesianas y una ontología extendida para expresar el modelo de referencia de intrusiones (IRM), que contiene información sobre la configuración de sitios a ser protegidos, las políticas de seguridad del mismos, los objetivos y los eventos de intrusión. En [10] se amplió el trabajo hecho en [5] basado en el concepto del objetivo. La investigación evidenciada en [19] presenta una arquitectura basada en agentes para correlación de alertas basada en ontología a partir de una propuesta extendida de XSWRL, la diferencia fundamental con la propuesta que se describe en este articulo radica en la inclusión de agentes de correlación y de técnicas híbridas inteligentes aplicadas en los agentes para procesos de clasificación, adicionalmente, nuestra arquitectura conduce a un modelo de prevención de intrusiones, dada las capacidades del agente de reacción para reconfiguración dinámica de reglas de filtrado; la propuesta referenciada opera sobre un conjunto de datos cargados del Dataset DARPA 2000 y normalizados al estándar IDMEF -entrenados y etiquetados- con atributos de alertas de alto nivel, en nuestro caso, operamos con datos obtenidos vía redes trampa/señuelo (*honeynet*), capturados en tiempo real y cargados a la ontología con una estructura de atributos que opera sobre la taxonomía del paquete (normal o anómalo), el modelo de correlación también es diferente en virtud de los sujetos, objetos y predicados utilizados para definir las reglas conducentes a procesos de inferencia.

2. MATERIALES Y MÉTODOS.

2.1 MODELOS ONTOLÓGICO.

El problema que se presenta en esta arquitectura requiere tener una visión clara de su dominio de conocimiento para definir un entendimiento común de conceptos, relaciones y aún más de la integración de los significados con otro tipo de tecnologías como cortafuegos y analizadores de vulnerabilidades. Dada la cantidad de datos que necesitan ser analizados, es indispensable filtrar y optimizar los contenidos de la comunicación con el fin de agilizar estas tareas. Una Ontología ayuda a reducir los tamaños de los mensajes intercambiados, limitar la redundancia y permite incorporar más inteligencia en el análisis de la información.

El Grupo de Ingeniería Ontológica de la Universidad Politécnica de Madrid creó *MethOntology* [11], esta metodología permite construir ontologías en el nivel de conocimientos, y tiene sus raíces en las actividades identificadas por el proceso de desarrollo de software propuesto por la organización IEEE y en otras metodologías de ingeniería de conocimientos.

A continuación se describe brevemente en qué consiste cada una de estas actividades:

- La especificación determina el por qué se construye la ontología, su posible uso así como sus posibles usuarios finales.
- La conceptualización organiza y convierte las percepciones informales en especificaciones semi-formales, aplicando un grupo de representaciones intermedias, que se basan en notaciones de tabulación y gráficas; como resultado se obtiene un modelo conceptual del dominio ontológico.
- La formalización transforma el modelo conceptual en un modelo formal o semi-computable.
- La implementación permite construir estos modelos en un lenguaje de ontologías (OWL, RDF, Schema, entre otros).
- El mantenimiento facilita la actualización y depuración de la ontología en caso de requerirse.

Las ontologías sirven como soporte para la implementación de sistemas basados en agentes, ofreciendo una estructura y estandarización en la representación de los objetos y de los mensajes que son enviados y recibidos entre ellos. En FIPA cada agente publica la ontología que está en capacidad de comprender, este servicio permite tener acceso a las ontologías públicas, traducir las expresiones en diferentes ontologías, responder a diferentes consultas sobre términos particulares, y facilitar la identificación y descubrir nuevas ontologías que puedan estar al servicio de los agentes. Algunas investigaciones han evidenciado el uso de ontologías en modelos de seguridad, algoritmos criptográficos y en sistemas de representación de anomalías. La utilización de estas representaciones permite estandarizar y unificar un lenguaje de comunicación homogéneo en la definición de firmas y eventos de los IDS. La evolución de una taxonomía tradicional a ontologías en el dominio de los ataques informáticos y de las intrusiones optimiza la gestión en un ambiente de IDS distribuido, ya que permite disminuir las dificultades de representaciones y comunicaciones heterogéneas. Las ontologías pueden ser definidas con base en estándares para lograr la definición de los meta-datos, entre otros RDF y OWL son los más difundidos, estos permiten transformar la información en una estructura global donde se puede compartir y reutilizar datos, documentos y otros recursos.

RDF proporciona información descriptiva simple sobre los recursos que pertenecen a un dominio particular.

OWL es un mecanismo para "desarrollar temas o vocabularios específicos en los que asociar esos recursos. Lo que hace OWL es proporcionar un lenguaje para definir ontologías estructuradas que pueden ser utilizadas a través de diferentes sistemas. Las ontologías, que se encargan de definir los términos utilizados para describir y representar un área de conocimiento, son utilizadas por los usuarios, las bases de datos y las aplicaciones que necesitan compartir información específica, es decir, en un campo determinado como puede ser el de las finanzas, medicina, deporte, etc.

OWL permite establecer formalismos y representaciones de conocimiento, algunas de las influencias más relevantes en el diseño de OWL proviene de su predecesor DAML+OIL de la Lógica Descriptiva, del paradigma de marcos y de los modelos RDF/XML.
Se ha decidido seleccionar a METHONTOLOGY como la metodología para el desarrollo de la ontología, dado que en múltiples estudios se considera como una de las metodologías más maduras que trata de seguir el ciclo de vida del software propuesto en el estándar IEEE 1074, por supuesto, susceptible de mejorar especificaciones para algunas tareas, en especial un lenguaje estandarizado de representación de conceptos, axiomas y reglas; también se puede ver como un punto a favor de METHONTOLOGY, el que sea recomendada por la "Fundación para los agentes Físicos Inteligentes" (FIPA), la cual promueve la interoperabilidad entre las aplicaciones basadas en agentes.

Para definir las reglas se ha usado SWRL (Semantic Web Rule Language), que se define como un lenguaje de expresión de reglas basado en OWL, permite escribir reglas expresadas como conceptos OWL proporcionando capacidades de razonamiento. Una regla SWRL contiene un antecedente para describir el cuerpo de la regla y un consecuente que se refiere a la cabeza de la misma, cada uno compuesto por un conjunto (puede ser vació) de átomos. Tanto el cuerpo como la cabeza de la regla son conjunciones positivas de los átomos; SWRL puede entender como significado que si todos los elementos en el antecedente son Verdaderos, entones el consecuente debe ser también cierto [12]. Las reglas SWRL están escritas en términos de clases, propiedades, instancias y valores de OWL. El siguiente código presenta una definición formal que describe el axioma *RootAccess* tipo de ataque por un total de 25 clasificaciones de intrusión obtenidas mediante el algoritmo de clustering aplicado (K-Means) y cerca de 2500 casos para nuestra ontología.

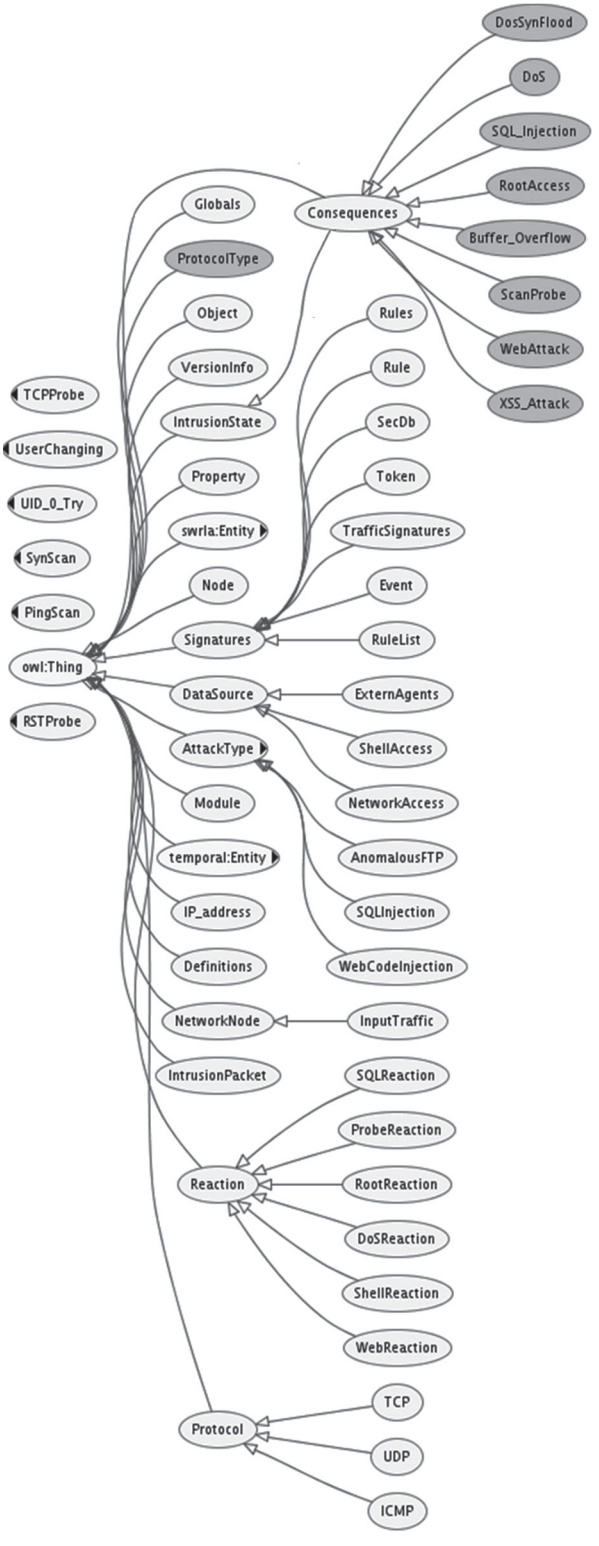
$$\begin{aligned} \text{RootAccess} \equiv\ & \ni (\text{IntrusionState} \cap \text{InputTraffic} \cap \\ & \text{NetworkNode}) \\ & \cap \ni \text{Generated\_by}(\text{AnomalousSSH} \cup \\ & \text{AnomalousFTP} \\ & \cup \text{WebAttack} \cup \text{TelnetAccess}) \cap \\ & \text{UID\_Try}(\text{UID\_0}) \end{aligned}$$

Nuestra Ontología Detección y Prevención de Intrusiones (OntoIDPSMA), ha sido diseñada utilizando Protégé como editor, incorporando los plugins para representación, así como las librerías de razonamiento, se han aplicado los tests de validación, en complemento, se envían peticiones al razonador DIG ejecutado con la plataforma RacerPro. Para poder especificar comportamientos de intrusión es conveniente identificar restricciones y reglas de comportamiento, políticas de identificación y reacción, así como instancias que disparan eventos anómalos. Por el momento hemos definido cerca de 3150 firmas de ataques y aproximadamente 1450 reglas de reacción, en complemento, se describen la secuencia de eventos que suceden en el sistema. En la fase de implementación del Sistema Multiagente se describe como se integra esta Ontología y como es el proceso de captura y carga de datos de paquetes e intrusiones.

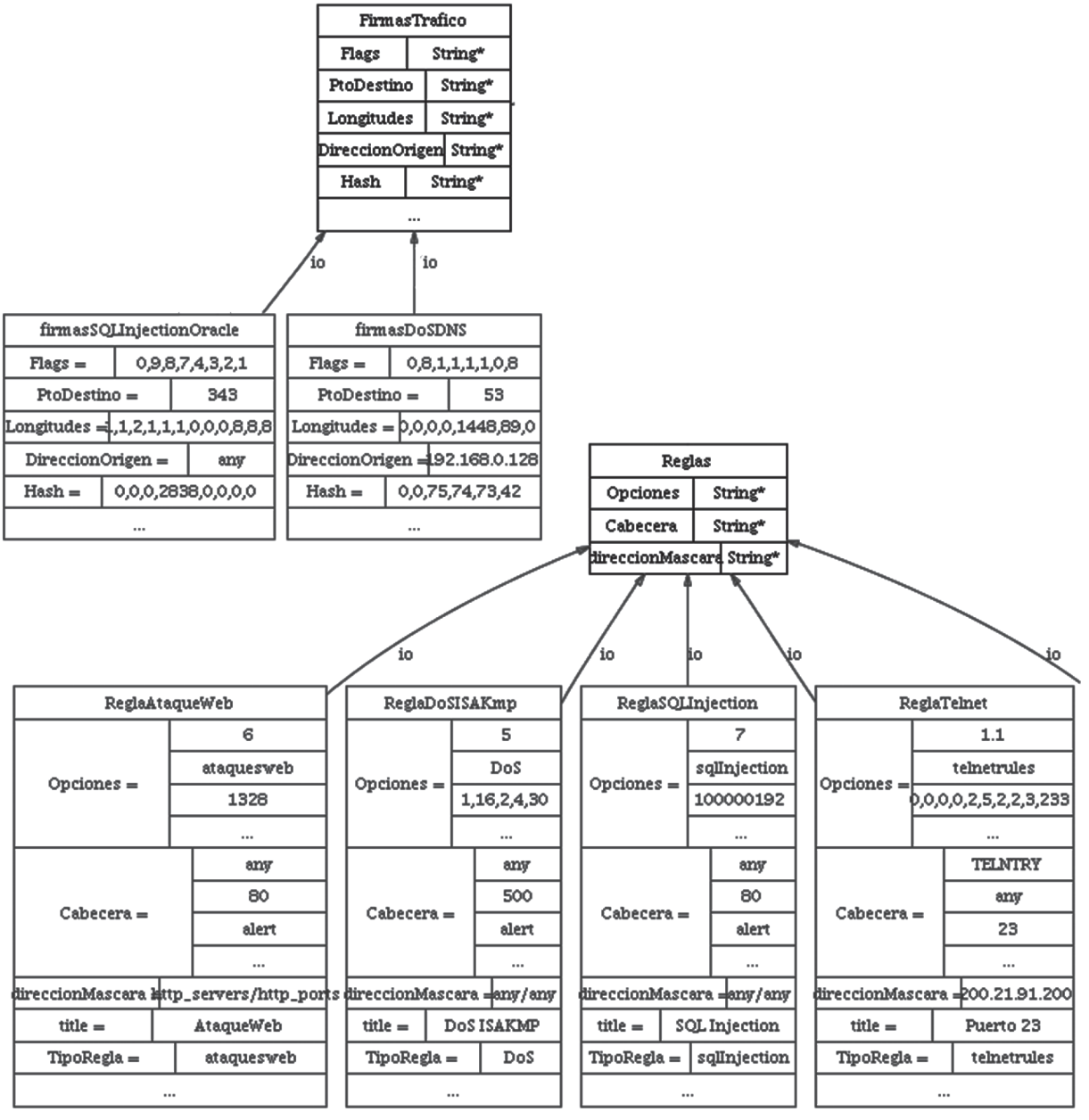
En [14] describimos el proceso de reacción a partir de un motor de correlación implementado en uno de los agentes. El modelo de carga de información y de representación de ataques de nuestra propuesta difiere de esfuerzos anteriores, ya que para nuestro caso se procesa información capturada directamente por los agentes y en el contexto de ubicación de un Nodo Sensor/Supervisor en modo señuelo y no desde el punto de vista del atacante o de la víctima, por otra parte, la transformación de información de ataques y tráfico normal hacia firmas es descrito con una secuencia semántica que ya ha sido explorada para identificar comportamientos anómalos, pero para nuestro caso, se incluye las tareas de reacción que permiten enfocar nuestra solución hacia un modelo de prevención de intrusiones y no solo de detección.

Para el modelo de clasificación que desarrollamos en esta investigación [15], se utilizó un conjunto de datos híbrido basado en el formato *tcpdump*. Inicialmente se realizó una captura de datos de tráfico normal e indebido usando la librería libpcap y jpcap (implementación de libpcap para Java) obteniendo un conjunto mezclado de paquetes normales y anormales (detectados con el motor de reglas de Snort); los paquetes anormales fueron etiquetados siguiendo el estándar del Dataset KDD'99. Sobre los datos reales capturados de la red y valorados como ataques por el motor de detección, se seleccionan los atributos relevantes para ser convertidos al formato del Dataset.

FIGURA 1. Ontología de Detección y Prevención de Intrusiones.



(a) Ontología



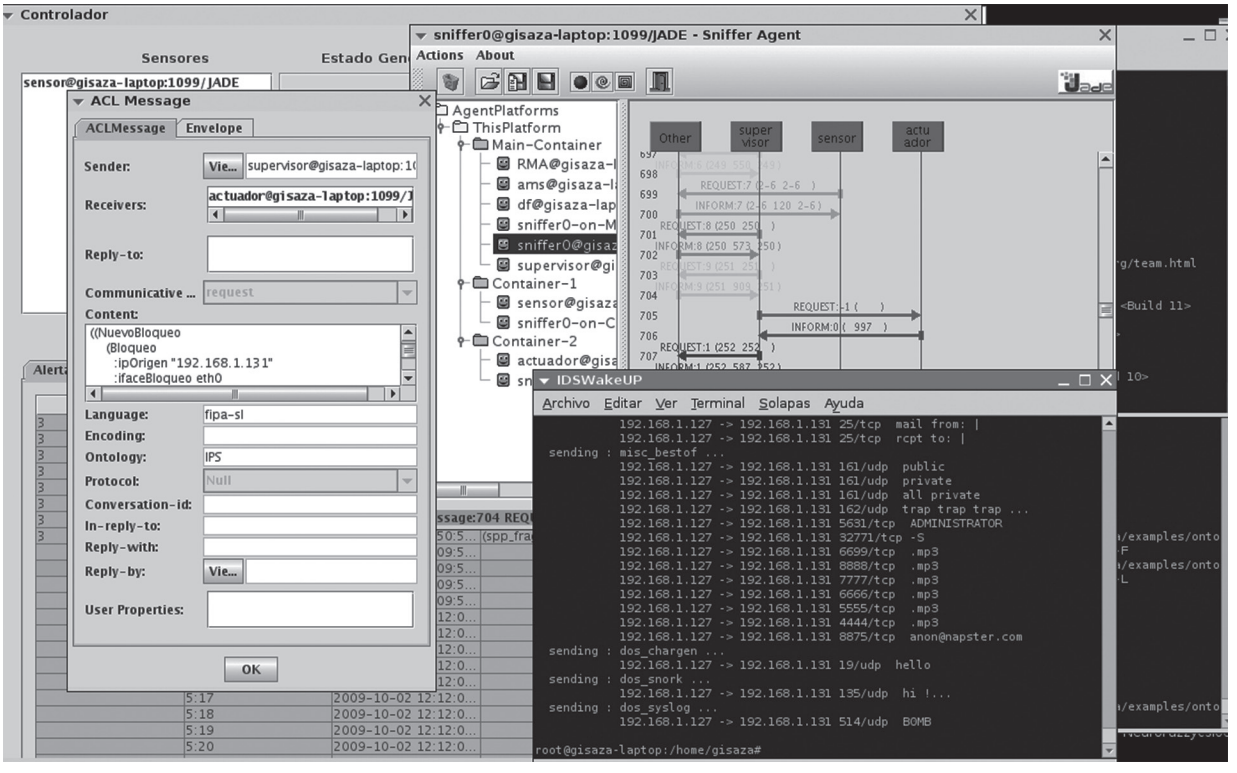
(b) Fragmento de representación ontológica par firmas de inyección SQL, Dos y Acceso Telnet No Autorizado.

La ontología representa las firmas de ataques conocidos y ataques nuevos, el comportamiento inteligente utiliza el modelo de inferencia y razonamiento herramientas de integración de redes neuronales en el sistema multi-agente, estas tareas se describen en [13] y [15], lo que la aplicación proporciona es un modelo ontológico de las normas de reacción y la creación del sistema de prevención a partir de la fusión y correlación de las alertas. Un fragmento de la ontología que implementa la detección de intrusiones y conocimiento de la prevención se representa en la Figura 1 (a) que presenta una vista de alto nivel y la Figura 1 (b) muestra un fragmento de la representación ontológica de las firmas de Tráfico para inyección SQL y reglas para diferentes ataques.

El comportamiento para la detección y prevención de intrusiones multi-agente ha sido diseñado utilizando un modelo ontológico integrado con la ontología a partir

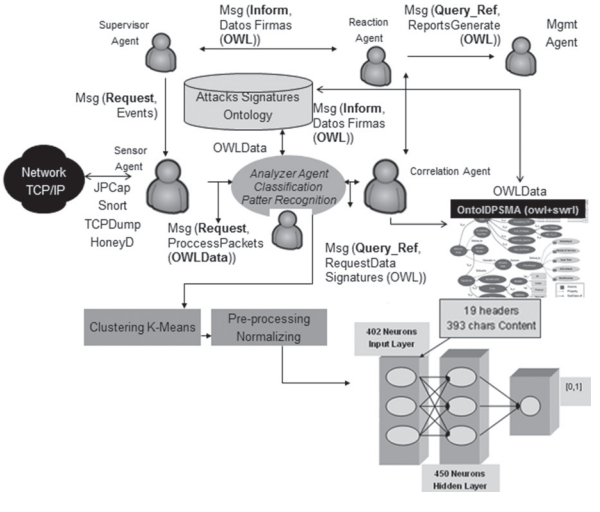
de reglas de reacción. Las capacidades de prevención se generan en el *Agente de Reacción* que espera mensajes para realizar un bloqueo o desbloqueo de una IP, protocolo o servicio. Inmediatamente, el supervisor es notificado por una acción, por lo que informa visualmente la Regla de Bloqueo y toma el control sobre las reglas aplicadas en cada uno de estos agentes, como se ve en la Figura 2.

FIGURA 2. Agente de Reacción y Mensaje ACL de Bloqueo.



El Agente Ontológico actualiza el *OntoIDPSMA.owl* y gestiona los OWL dentro de los mensajes de ACL. Esta interfaz permite observar los mensajes que se intercambian. La Figura 3 presenta la arquitectura completa para el sistema propuesto incorporando el comportamiento sensorial vía redes señuelo (OntoIDPSMA).

FIGURA 3. Arquitectura del Sistema Multiagente *OntoIDPSMA.*



2.2 FUSIÓN, CORRELACIÓN Y PROCESOS DE INFERENCIA.

En el problema de detección de intrusos, se utilizan diferentes tipos de técnicas de fusión de sensores. Algunas priorizan las alertas para la reducción de eventos, otras agrupan a partir de la identificación de patrones comunes de ataque y algunas correlacionan alertas para identificar diferentes fases de los ataques. En nuestro modelo, la técnica inteligente usa la correlación de los grupos de intrusiones identificadas para la fusión de alertas homogéneas, así mismo, se invoca el proceso de interpretación, combinación y análisis de las alarmas reportadas por los sensores, la correlación de alertas pretende eliminar alertas redundantes mediante la agregación o fusión de alertas similares.

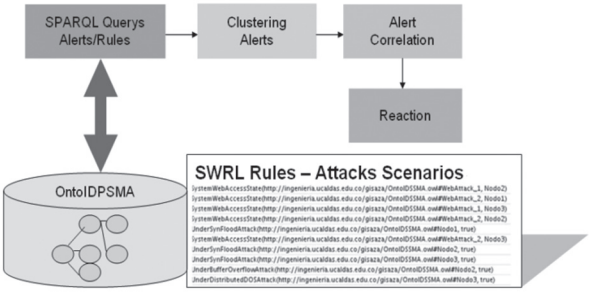
El modelo de fusión implementado propone un enfoque que correlaciona la semántica de ataques a partir de eventos generados por los acontecimientos heterogéneos que conducen a analizar escenarios anómalos a partir de la representación de un lenguaje como SWRL sobre el cual se ejecutan procesos de inferencia y razonamiento. Los grupos de alertas de cada Agente Sensor se asignan a estas categorías (grupos identificados por la técnica de Clustering, se detalla en [15]), basado en descripciones de eventos. Este modelo se basa en el modelo de reglas propuesto por Cuppens [17].

El propósito de la correlación de intrusiones tiene como objetivo integrar y correlacionar alertas para proporcionar el reconocimiento y dar información sobre la secuencia de los ataques para generar intenciones de intrusión o de manifestar posibles ataques implícitos que no se identifican explícitamente.

El uso de la correlación depende del contexto de múltiples fuentes, para nuestro caso, múltiples *Agentes Sensores* distribuidos, con el conocimiento de las representaciones ontológicas. La verificación de alertas y la fusión puede minimizar las alertas totales en un entorno distribuido, sin comprometer la capacidad de interpretación en cada nodo de detección, estas técnicas permiten incorporar un modelo más comprensible e inteligente para la secuencia del intruso. La gestión del proceso de correlación integrado en nuestro sistema multi-agente se presenta en la Figura 4.

El modelo de correlación utilizado combina las alertas de detección de diferentes fuentes y aplica una técnica híbrida de similitud para el análisis de los atributos prioritarios, la similitud de atributos se deriva de una medida de correspondencia para elementos que son agrupados a partir de la técnica de clustering aplicada, en nuestro caso el algoritmo K-Means en el *Agente Analizador* con capacidades inteligentes.

FIGURA 4. Modelo de Correlación para la Ontología [14].



Las correlaciones alcanzadas evidencian patrones de comportamiento entre los diferentes atributos que No eran explícitos, así, cuando un IDS genera un nuevo evento, la función de agrupación determina las alertas existentes que pueden ser "vinculadas" a esta nueva alerta. Una relación de similitud define en qué casos dos alertas se consideran lo suficientemente cerca para ser agrupadas. Por lo tanto, la técnica de clustering define la función que tiene que pasar por la especificación de la relación de similitud entre las diversas entidades que conforman una alerta, en particular tiempo de detección, direcciones de origen/destino, carga de datos y otros atributos continuos y discretos relevantes para la clasificación. Esta aproximación facilita la agrupación de atributos para detectar clusters permitiendo la clasificación de un conjunto de elementos de muestra sobre grupos basados en similitudes y diferencias. Para este caso se ha usado el algoritmo *K-Means* que trata de un método aprendizaje no supervisado que apunta a minimizar la función objetivo, es decir la distancia interna (cercanía hacia los vectores de características comunes).

La descripción del algoritmo se presenta como:

#Algoritmo K-Means
Entrada : El número de Clusters K de un Dataset de Detección de Intrusiones
Salida: Un conjunto de K-clusters que minimizan el criterio del error cuadrático
Algoritmo:
1 Inicializar K clusters (seleccionar aleatoriamente k elementos de los datos)
2 Mientras la estructura del cluster cambia
Repetir desde 2.
3 Determinar los clusters a los que pertenece un dato
Usar la fórmula de la distancia Euclidiana
Adicionar elementos al cluster con la distancia mínima
4 Calcular el significado de los clusters
5 Cambiar el centro de los clusters a los significados obtenidos en el paso 3.

En nuestro sistema multi-agente, el motor de correlación/ reacción realiza un bloqueo de una secuencia sospechosa presente dentro de la red supervisada por los sensores. Estos bloqueos se propagan por todos los agentes de reacción que se registran dentro del contenedor principal del supervisor, independientemente de la red a la que pertenecen, con el fin de bloquear los ataques posibles en estas redes, incluso si dicho tráfico no ha estado allí. Esta capacidad de los agentes constituye el marco prevención.

El motor de correlación tiene una lista blanca o lista de direcciones privilegiadas y que no se bloquearán, independientemente de las medidas que se deban tomar, cuando se considera una advertencia originada de un host específico. La aplicación cuenta con cinco modos operativos funcionales, cada una cuenta con una amplia calificación [0 ... 5] o [0 ... 10], utilizando el *Agente Supervisor* es posible seleccionar la calificación mínima para generar un evento de bloque. Los 5 modos se describen en [14]. La diferencia fundamental entre el modo de correlación implementado en esta fase con respecto a la descrita en la publicación mencionada, radica en el *uso de las capacidades de razonamiento e inferencia dentro de la Ontología a partir de representaciones codificadas SWRL* para fusionar alertas y construir escenarios de ataques (Ver Figura 4), en la versión anterior los modos de correlación para el agente de reacción hacen parte del comportamiento de este agente, en esta nuevo desarrollo, estas capacidades se derivan de la invocación de tareas de inferencia y razonamiento a las reglas de los eventos de intrusión.
Para nuestro caso, el modelo matemático subyacente se describe como: Sea $A^h = \{A^h_1, A^h_2, A^h_3 \ldots\ldots A^h_n\}$ el número total de alertas de un host y $A = \{A_1, A_2, A_3 \ldots\ldots A_n\}$ el número total de alertas del sistema distribuido basado en agentes, la calificación pertenece en un rango de 0 a 10. El valor de *Confiabilidad* es calculado como:

$$\text{Confiabilidad} = Round \frac{\sum_{i=1}^{n} A^h{}_i}{\sum_{i=1}^{n} A_i} \times 10$$

Sea VA el Valor Activo, PS la prioridad del Sensor y R la confiabilidad, el valor del riesgo está dado por:

$$Riesgo = \text{Round}(\frac{VA \times PS \times R}{10})$$

En este enfoque, el nivel de confiabilidad tiene más relevancia, si un host está generando un mayor número de alertas, independientemente del valor activo donde los sensores se ejecutan. Cualquier alerta generada por el motor de correlación, debe tener un tiempo de bloqueo designado, que puede configurado en la interfaz del Agente Supervisor. El Agente Supervisor tiene un comportamiento que monitorea la lista de bloqueos y necesita de las cerraduras del motor de correlación en busca de eventos expirados, si se encuentra, se lo notifica a los Agentes de Reacción para llevar a cabo la liberación respectiva.

La representación lógica utilizando OWL para una definición de la instancia de correlación se presenta como:

```
<rdf :RDF
xmlns : j.0="http://ingenieria.ucaldas.edu.co/gisaza/
OntoIDPSMA.owl #"
.....
xmlns : rdf=
resource="Resource"/>
.....
 "Ontology IDSPSMA">
        </owl:Ontology>
        <owl:classrdf:ID="PortScan">
        <rdfs:Subclass of>
        <owl class rdf:ID="Alert">
                <Name> Port Scanning </Name>
                <AlertType>R2L</AlertType>
                <Reaction>Inform</Reaction>
                <Protocol>tcp</Protocol>
                <Source>home_net</Source>
                <Destination>external_net</
                Destination>
                .
                .
</rdf :RDF>"
```

Para estas representaciones, es posible construir un escenario de ataques que definen precondiciones y post-condiciones que determinan por ejemplo que: *A* es un subconjunto de relaciones de clases de ataque entre los ataques *12°* donde si un ataque es un tipo de ataque de A_2, se puede afirmar que A_1 es subclase de A_2 y superclase de A_1, y si un ataque tiene un conjunto de estados S_1 y S_2 pre-requisitos uno del otro con instancias en *A* con sus respectivos cuerpos, si S_1, S_2 son prerrequisitos de un estado, entonces se cumplen las condiciones para configurar el escenario. La representación SWRL de este estado, se describe como:

A(?p) ∧ S1(?q) ∧ PreconditionIntrusionState(?p, ?q) ∧ S2(?z) ∧ PreconditionIntrusionState(?p, ?q) → State(?p, true)

Como se menciona en [17], la representación es parte de un concepto de correlación indirecta, donde un ataque A_1 y A_2 están directamente correlacionados a través de las reglas ontológicas R_1 ... R_n si siguen las siguientes condiciones:

- El Ataque A1 está directamente correlacionado con la Regla R1 a través de algún unificador más general, en la lógica de primer orden, de todos los unificadores posibles siempre existe al menos uno que es el menos restrictivo, en el sentido que es el que menos restringe futuras unificaciones. Este tipo de unificador se conoce como unificador más general (UMG).
- Para cada *j* en *[1, n-1]*, la regla R_j está directamente correlacionada con la regla R_{j-1} a partir del unificador más general.
- La regla R_n está directamente correlacionada con el ataque A_2 utilizando un unificador más general.

El sistema multiagente se compone de un grupo de sensores $S = \{s_1, s_2, s_3, \ldots\ldots s_n\}$ donde cada Sensor se define como un conjunto de agentes autónomos que permite monitorear y reportar posibles intrusiones o anomalías que ocurren en la red a partir de técnicas de comparación de firmas. El sistema monitoreado por el IDPS (*Intrusion Detection and Prevention System*) puede ser representado como un conjunto de nodos $N = \{n_1, n_2, n_3, \ldots\ldots n_n\}$ que pueden ser posibles objetivos de los atacantes, el conjunto de intrusiones detectadas $I = \{I_1, I_2, I_3, \ldots\ldots I_n\}$y un conjunto de propiedades de cada tipo de intrusión $P = \{ P_1, P_2, P_3, \ldots\ldots P_n \}$, donde P_i representa un tipo de ataque (Tipo Web, Escalada de Privilegios, Denegación de Servicios), el conjunto de ataques $A = \{A_1, A_2, A_3 \ldots\ldots A_n\}$ representa las instancias de ataques marcados correctamente.

El Dataset de tráfico se define como $D = \{d_1, d_2, d_3, \ldots\ldots d_n\}$ a partir del cual se generan subconjuntos de entrenamientos $D_{t} = \{ dt_1, dt_2, dt_3, \ldots\ldots dt_n \}$.
La eficiencia del modelo determinada por la capacidad de interacción entre los agentes sensores $S = \{s_1, s_2, s_3, \ldots\ldots s_n\}$ enviando requerimientos a sus agentes analizadores, de correlación y reacción y determinando la precisión de clasificación en función de los datos de entrada, los patrones de entrenamiento y los valores correspondientes a falsos positivos, negativos está dada por:

$$Eficiencia = \sum_{i=1}^{N} S_i \times \frac{\sum_{i=1}^{n} I_i}{\sum_{i=1}^{n} A_i}$$

En [17], [18] se menciona que la Correlación y Fusión de Alertas dada una nueva alerta *A* y un conjunto *n* dada una nueva alerta *H* y un conjunto de ataques existentes m-attacks $S = \{M1, M2, \ldots, Mn\}$, el proceso de correlación apunta a encontrar el subconjunto apropiado *S' de S* y la combinación *H* con *S'* donde *S'* es un conjunto de m-attacks $S' = \{Mi1 \ldots Mik\}$ $(0 \leq k \leq n)$ tal que si *S'* es un conjunto No vacío, *H* y todos los m-attacks están en *S'* es posible que estén embebidos dentro del mismo escenario, porque *H* requiere ser satisfecho en la regla definida.

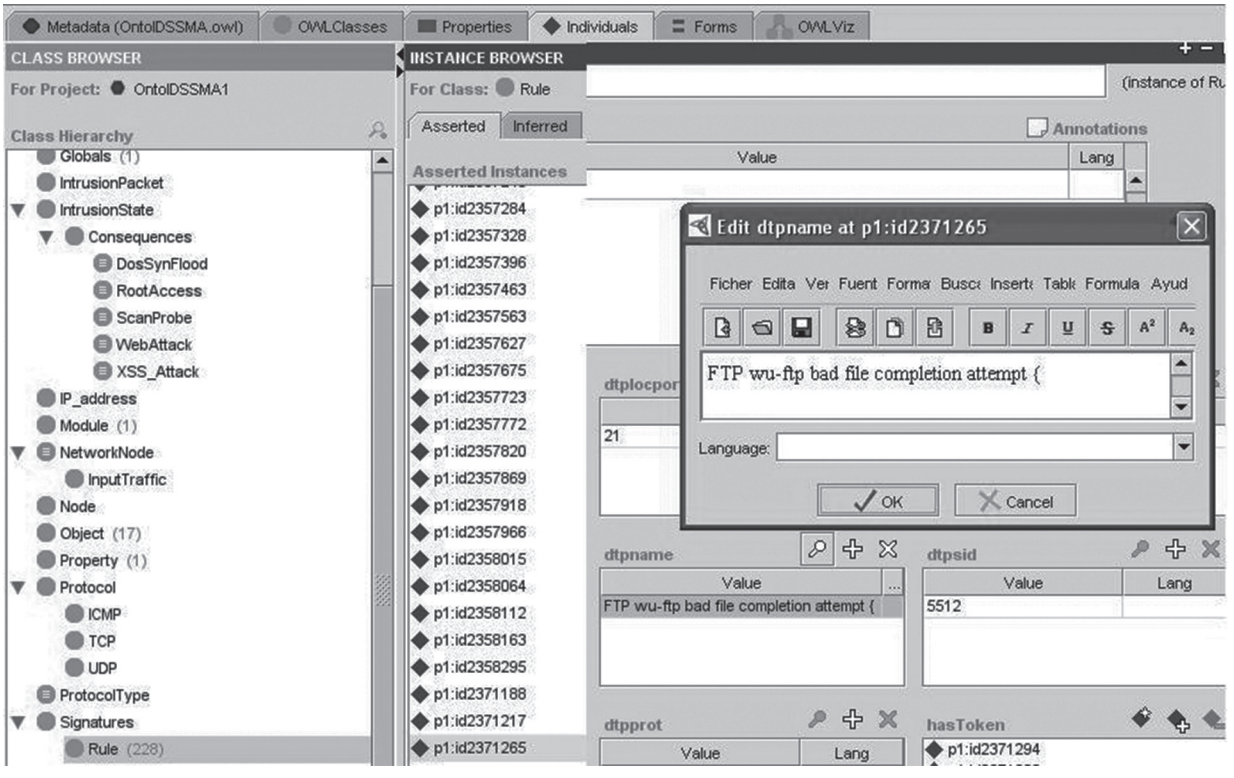
En síntesis, el componente de *Fusión* combina las alertas de detección que representan instancias independientes del mismo ataque por los diferentes sensores de intrusión y el componente de *Correlación* es un proceso multi-componente que recibe como entrada una corriente de alertas de los múltiples sistemas de detección de intrusiones (en modo señuelo para nuestro caso). En cada uno de los componentes del proceso, las alertas se fusionan de acuerdo con los comportamientos de intrusión de determinados procesos "inteligentes" y etiquetados como irrelevantes si no representan ataques significativos. En consecuencia, las alertas se priorizan de acuerdo con la política de seguridad antes de ser informado por el agente de reacción y supervisión.

3. RESULTADOS.

El proceso de correlación llama al *Razonador* JESS (basado en reglas) para correlacionar los elementos de las tripletas que corresponden a instancias de los ataques. La entrada del razonador SWRL es la ontología definida. Los ataques que están dentro del mismo grupo de intrusión se correlacionan entre sí relacionando los atributos comunes derivados de las técnicas de clasificación.

La Figura 5 muestra la representación Protégé de una instancia para un evento de comportamiento anormal FTP, para este caso se presenta el valor de atributo dtpname que identifica la cadena de alerta con información capturada por el sensor. La instancia de la clase Signature (Firma) definida aquí, es parte de las reglas que definen las clases de ataques (*Rule*).

FIGURA 5. Representación de instancia en Protégé para un FTP anómalo.



Usando el plugin de Protégé de Jambalaya se pueden obtener diferentes tipos de visualización de la ontología en un esquema interactivo, incluyendo las relaciones, propiedades e individuales, en la Figura 6 se presenta un esquema generado con esta utilidad donde se mapean tanto las clases como las instancias, se aclara que para nuestro caso la visualización que se genera a partir de las hojas del árbol es débil porque corresponden a las instancias de reglas y firmas de ataques.

La incorporación de reglas en la Ontología, definidas previamente en la aplicación de MethOntology, proporciona un enfoque semántico que brinda posibilidades de razonamiento e inferencia. Para el dominio de nuestra Ontología hemos definido un conjunto genérico de reglas utilizando el Motor y el plugin de SWRL, así como las APIS correspondientes para ser integrado en la plataforma de Agentes y procesado en tiempo de ejecución a través de JENA. En la Figura 7 se presenta la interfaz de definición de reglas en Protégé, para este diagrama se describe de manera extendida la Regla Nodo Bajo Ataque de Desbordamiento de Buffer (BufferOverFlow_To_NodeRule) que se describe como:

Si (x es Nodo de Red -> NetworkNode(?x) ^ y es un Estado de Intrusión -> IntrusionState(?y) ^ Generado a x por y -> GeneratedBY(?x, ?y) ^ z es un ataque de tipo BufferOverflow -> BufferOverflow(?z) ^ z es un ataque de tipo Intrusión y -> AttackTypeOf(?y, ?z) Entonces El Nodo x se encuentra bajo ataque de BufferOverflow -> UnderBufferOverflowAttack(?x, true)

FIGURA 6. Instancias Inferidas, Reglas y Axiomas.

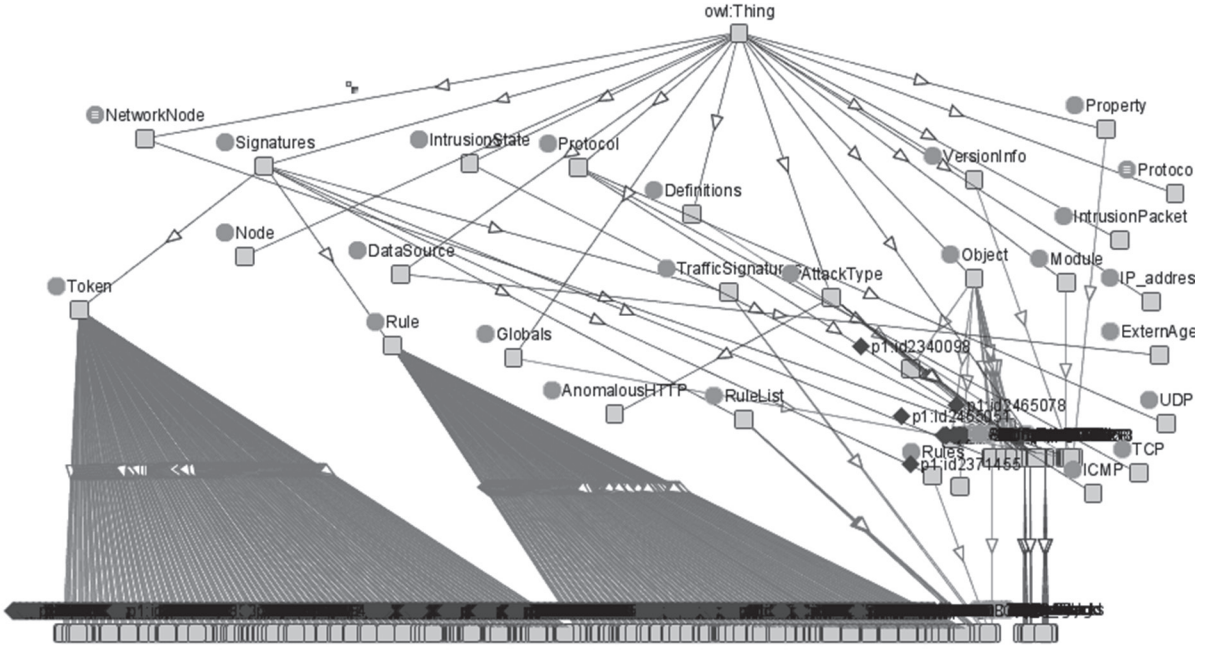
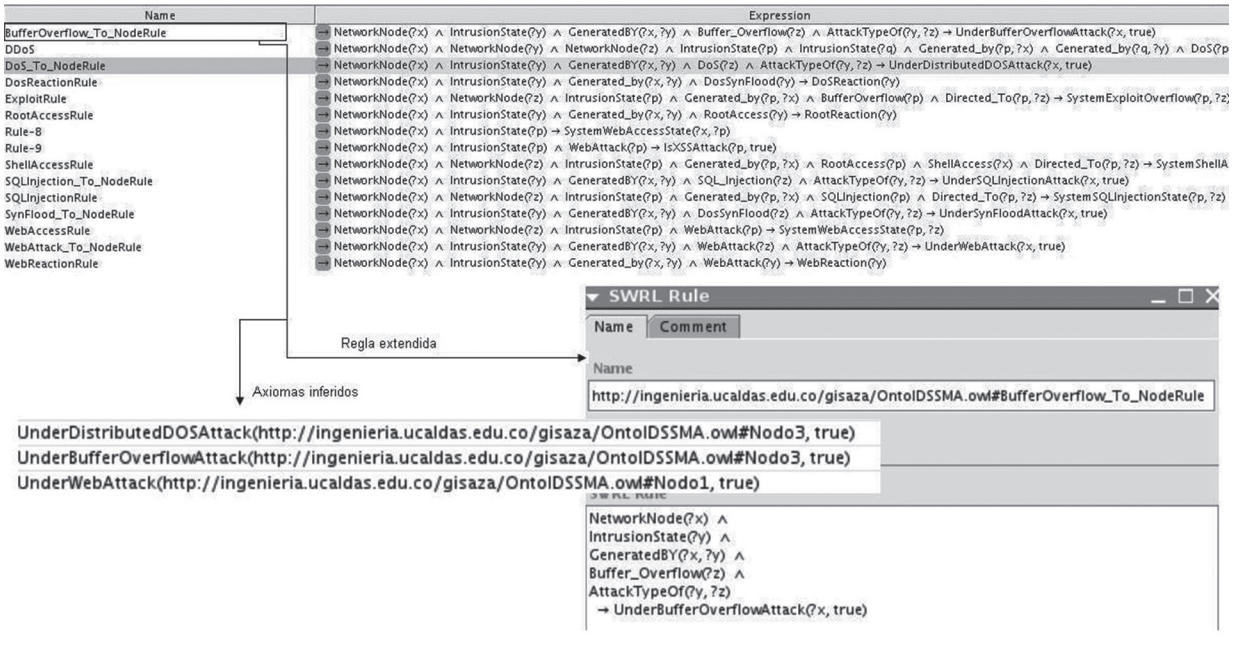


FIGURA 7. Definición de Regla usando SWRL.



El motor de correlación implementado permite correlacionar alertas a partir de un modelo basado en prioridades de los eventos de intrusión.

El modelo desarrollado permite definir un ataque dentro del contexto de un conjunto de características, basado en la propuesta hecha por [17] en el CRIM como parte del proyecto MIRADOR de la DGA (French Defense Agency) y ALCATEL para construir plataformas de IDS cooperativos y adaptativos donde se propone un módulo de cooperación que agrupa y asocia las alertas usando el formato IDMEF; para nuestro caso adaptado a la Ontología OWL de Detección y Prevención y representando las reglas con SWRL, así:

- Precondición de Ataque: Condición Lógica que especifica el conjunto de condiciones que se deben cumplir para que suceda un ataque.
- Postcondición de Ataque: Condición Lógica que determina el efecto que puede tener un ataque cuando este sucede.
- Escenario de Ataque: Se determina por la combinación de eventos que el intruso realiza cuando se ejecuta un ataque.
- Escenario de Detección: La combinación de eventos que son necesarios para detectar una ocurrencia de un ataque.
- Escenario de Verificación: Combinación de eventos que deben ser lanzados para verificar si un ataque ha sido realizado con éxito.

Con este tipo de representaciones es posible determinar correlaciones implícitas a partir de las relaciones entre diferentes eventos de intrusión. Esta aproximación se basa en la observación de grupos de alertas y en la extracción de relaciones entre estas. De igual forma el modelo puede también representar correlaciones explícitas, donde el usuario administrador "experto" está en capacidad de expresar conexiones entre diferentes eventos que puede reconocer. Cuando se cumplen las condiciones de propiedades y relaciones entre eventos se genera un predicado del tipo:

Correlación_Alerta(AtaqueTipo1, AtaqueTipo2): Determina que el Ataque de Tipo 1 puede estar correlacionado y ser antecedente del Ataque Tipo 2. Por ejemplo, un escaneo de puertos TCP puede estar correlacionado con el lanzamiento de exploits remotos a un servicio particular, dado que el proceso de escaneo de puertos, permite determinar servicios abiertos, así como niveles de vulnerabilidad que facilitan al atacante suficiente información para lanzar algún script anómalo al puerto asociado. En el siguiente esquema de ejemplo se representa una correlación entre dos posibles tipos de alertas genéricos de Escaneo de Puertos y accesos indebidos de máquina remota a entornos locales, para este caso en OWL y SWRL, se define genéricamente así:

```
Alert_Correlation(AttackType_1, AttackType_2)
   Alert(AttackType_1)
      Address(AttackType_1, Target)
      Service(AttackType_2, TCP)
      Classification(Alert, "PortScan")
   Alert(AttackType_2)
      Address(AttackType_2, Target)
      Service(AttackType_2, R2L)
      Classification(AttackType_1, "ExploitXYZ")
```

Como se ha mencionado, a partir de estas reglas es posible construir un motor de correlación basado en OWL tomando como referencia las reglas de antecedentes –precondiciones- y la clasificación provenientes de la regla consecuencia –postcondición-. El proceso de correlación incorporando las políticas que se definidas en este sistema se puede observar en el siguiente fragmento de pseudocódigo:

```
For A(a1, a2 . . . an)
Create_Alert(OWL)  //From Ontology
// Clustering duplicate alerts in each // Sensor
   For A(a1, a2 . . .  an)
      If Alerts_Sim(atributo_SigName) in Si then
         Correlate(A, Si) ->Correlated[A,Si]
   For S (s1, s2 . . . sn)
      If Sim_Alerts_Sensor(Correlated[]) in Si then
      // construct Attacks Scenaries
      New_AlertGroup(Correlated[],Sensor[])
```

En la tabla 1 se presenta un fragmento de las alertas correlacionadas dependiendo de las instancias de intrusión por Sensor y la Tabla 2 evidencia las alertas fusionadas y correlacionadas usando la arquitectura del SMA (sistema Multiagente).

TABLA 1. Alertas fusionadas y correlacionadas por sensor.

Firma	Timestamp (inicio-final)	IP Origen	IP Destino	Sensor	Tag
XSRF Flow	08:22/09:11	192.X.Y.Z	200.21.W.Y	Sensor01	Correlacionada
SQL Injection	02:21/02:32	192.P.Q.R	200.21.K.R	Sensor02	Correlacionada
XSS Ajax	16:23/17:04	192.A.V.X	200.21.X.Z	Sensor03	Correlacionada

TABLA 2. Alertas fusionadas y correlacionadas por el SMA.

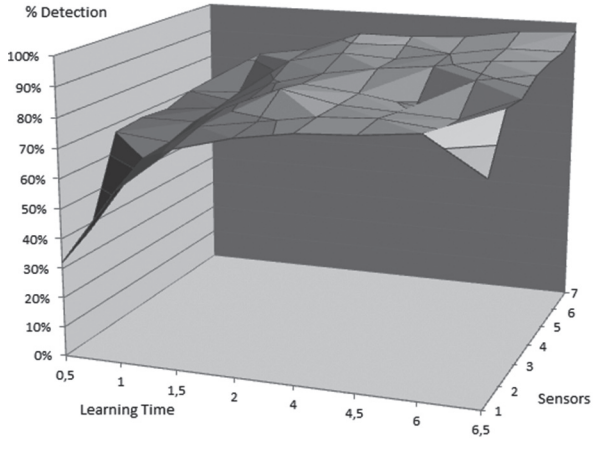
Sensor	Alertas Generadas	Alertas Correlacionadas
Ag_Sensor01	34789	224
Ag_Sensor02	16718	4123
Ag_Sensor03	9103	1180
Ag_Sensor04	28654	2431

El comportamiento del sistema en el proceso de aprendizaje basado en el número de sensores, el tiempo transcurrido (minutos) de correlación y fusión, así como la precisión de clasificación porcentaje que se muestra en la Tabla 3 representan el comportamiento de la superficie de la figura 8, la descripción de estas tareas se hace explícita en [13].

TABLA 3. Porcentaje de Detección en función del número de sensores.

	Sensores						
Tiempo (min)	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0,5	42 %	45 %	67 %	68 %	67 %	70 %	71 %
1	62 %	64 %	65 %	68 %	69 %	71 %	82 %
1,5	73 %	76 %	78 %	77 %	79 %	84 %	85 %
2	78 %	79 %	86 %	84 %	84 %	88 %	92 %
4	81 %	82 %	82 %	82 %	87 %	87 %	92 %
4,5	82 %	83 %	81 %	79 %	89 %	90 %	93 %
6	84 %	81 %	80 %	89 %	92 %	93 %	96 %
6,5	72 %	89 %	88 %	92 %	93 %	93 %	97 %

FIGURA 8. Tiempo de Correlación / Fusión y porcentaje de detección en función del número de sensores.



En el agente inteligente se aplica una técnica híbrida a partir de un modelo de neuro-clasificación, descrito en [15], los resultados obtenidos con las mejores configuraciones de rendimiento (para un modelo supervisado) se presentan en la Tabla 4, siendo *PkG* Número de Paquetes, *FP* falsos positivos, *FN* Falsos Negativos, *U* paquetes incertidumbre, *VP* Verdaderos Positivos, *ET* tiempo transcurrido.

TABLA 4. Comportamiento a partir de Técnica Inteligente Híbrida.

Paquetes	**PkG**	**FP**	**FN**	**U**	**% VP**	**ET**
Entrenamiento	1955	0	1	0		
Validación	101	0	0	0		
Normal	50000	752	-	573	97,94%	285 sec
Anómalos	246	-	17	4	99,05%	

4. CONCLUSIONES.

Las taxonomías y estándares de representación existentes para los IDSs son insuficientes para soportar una óptima identificación de ataques, definición y razonamiento predictivo de comportamientos anómalos. El uso de representaciones ontológicas en la detección de intrusiones facilita la interoperabilidad y comunicación en los IDS distribuidos, el desempeño de estos modelos puede mejorar significativamente integrando los componentes de formalización de su conocimiento, de autonomía y reacción a partir del uso de técnicas de inteligencia computacional en el proceso de análisis y de gestión de firmas, así como la comunicación e interacción distribuida basada en agentes. Los resultados logrados evidencian la optimización que puede lograrse en los modelos de detección y prevención a partir de técnicas de clasificación y reconocimiento de patrones embebidos en agentes inteligentes y la maximización de la representación homogénea de conocimiento de los IDS a partir de ontologías, ofreciendo a la arquitectura capacidades de razonamiento e inferencia. La ontología diseñada, implementada e integrada propone un esquema para modelar la conceptualización de un subconjunto del dominio de firmas de ataques y reglas, permitiendo a los agentes resolver dificultades de heterogeneidad en sus bases de conocimiento y ofreciendo una escalabilidad de la misma para lograr razonamientos e inferencias. En esta aproximación se ha definido las firmas de ataques, reglas de reacción, afirmaciones, axiomas usando el lenguaje OWL-DL integrando un modelo de correlación presente en el Sistema Multiagente.

Formalizar el comportamiento mediante ontologías, permite la creación de repositorios que pueden estar delimitados por un dominio, facilitando la reutilización de componentes basados en agentes con cierta homogeneidad en su funcionalidad. Por otra parte, el modelo planteado permite integrar nuevas capacidades de inteligencia, razonamiento y clasificación, esto se debe a que el componente que permite clasificar, razonar y ejercer un comportamiento inteligente, es independiente de la plataforma basada en agentes, para nuestro caso, es implementado como un proceso externo que se ejecuta por lotes mientras los Agentes sensores y de actualización ontológica continúan operando; esta característica provee a la arquitectura una óptima escalabilidad. El uso de un lenguaje de representación de reglas para modelar los escenarios de ataques proporcionan un marco sensible de aplicación de inferencias y razonamientos que pueden arrojar instancias inferidas para delegar el comportamiento de correlación en la ontología y no en un agente específico del sistema multi-agente expuesto, de esta manera se reduce significativamente la carga de trabajo en los agentes y se aprovechan las capacidades de las ontologías.

La relación de este componente de correlación y fusión con otros ya publicados no permite hacer un análisis comparativo dado el tipo de alertas procesadas (estructura interna de los atributos) y la arquitectura disímil basada en Agentes con comportamientos divergentes, adicionalmente, algunas de estas investigaciones no referencian los elementos cuantitativos de correlación y fusión, no obstante, es importante resaltar que algunas taxonomías [8],[10],[19] han sido un referente muy valioso para este proyecto. Otros resultados publicados por los autores en el marco de esta investigación si presentan comparaciones contra otros parámetros como tasas de detección, de precisión, de falsos positivos y negativos a partir de entrenamientos supervisados y No supervisados.

5. REFERENCIAS.

[1] HOWARD, Jhon and LONGSTAFF, Thomas. A Common Language for Computer Security Incidents. Sandia Lands Reports. 1998. pp. 16.

[2] ECKMAN, Steven, VIGNA, Giovanni and KEMMERER, Richard. STATL: An Attack Language for Statebased Intrusion Detection. Journal of Computer Security, No. 1-2, p. 71-104. 2002.

[3] CURRY, David y otros. Intrusion detection message exchange format. In Intrusion Detection Working Group – Internet Engineering Task Force, Internet Draft. 2004 http://www.ietf.org/rfc/rfc4765.txt

[4] MARTIN, Francisco and PLAZA, Enric. SOID: An Ontology for Agent-Aided Intrusion Detection. Springer Berlin / Heidelberg, Volume 2773, p. 1222-1229. 2003.

[5] MARTIMIANO, Luciana and DOS SANTOS, Edson. An OWL-based Security Incident Ontology. 8th Intl. Protégé Conference, Madrid, Spain. 2005, pp. 12, 16.

[6] JOTSOV, Vladimir. Ontology-Driven Intrusion Detection Systems. International Conference Knowledge-Dialogue-Solutions. 2007, pp. 1-11.

[7] HUNG, Shao-Shin and LIU, Damon. A user-oriented ontology-based approach for network intrusion detection, Elsevier Science Publishers B. Volume 30, p. 78-88. 2008.

[8] CUPPENS-BOULAHIA, Nora, CUPPENS, Freddery and other. An ontology-based approach to react to network attacks. Risks and Security of Internet and Systems. CRiSIS `08. Third International Conference on Publication, p. 27-35. 2008.

[9] GOLDMAN, Robert and HARP, Stephen. Model-based Intrusion Assessment in Common Lisp. International Lisp Conference, Association of Lisp Users and ACM SIGPLAN, Cambridge, MA. 2009.

[10] MANDUJANO, Salvador, GALVAN Arturo, and NOLAZCO, Juan. An ontology-based multiagent approach to outbound intrusion detection.in Computer Systems and Applications. The 3rd ACS/ IEEE International Conference, p. 94. 2005.

[11] CORCHO, Oscar, LÓPEZ, Mariano, GÓMEZ-PÉREZ, Asunción and LÓPEZ-CIMA, Angel. Building Legal Ontologies with METHONTOLOGY and WebODE, in Lecture Notes in Computer Science - Law and the Semantic Web, p. 142-157. 2005.

[12] OCONNOR, Martin, RABI Shankar and other. Developing a Web-Based Application using OWL and SWRL. Stanford Medical Informatics, Stanford University, Stanford, CA 94305-5479. 2008.

[13] ISAZA, Gustavo, CASTILLO, Andrés, LOPEZ, Manuel, and CASTILLO, Luis. Towards Ontology-Based Intelligent Model for Intrusion Detection and Prevention. Journal of Information Assurance and Security, No. 5, p. 376 – 383. 2010.

[14] ISAZA, Gustavo, CASTILLO, Andrés, LOPEZ, Manuel, and CASTILLO, Luis. Intrusion Correlation Using Ontologies and Multi-agent Systems. En Alemania Communications in Computer and Information Science, vol.76 fasc.1, p. 51 – 63. Heidelberg: Springer-Verlag, 2010.

[15] ISAZA, Gustavo, CASTILLO, Luis, LOPEZ, Marcelo, y MARULANDA, Carlos. Modelo híbrido de neuroclasificación y clustering en el problema de detección de intrusiones. En: Colombia Vector No. 5, p.69 – 77. Centro Editorial de la Universidad de Caldas, 2012.

[16] CUPPENS, Freddery. Managing alerts in a multi-intrusion detection environment. In Proceedings of the 17th annual computer security applications conference, p. 22–31. New Orleans, Louisiana, 2001.

[17] CUPPENS, Freddery and MIEGE, Alexandre. Alert correlation in a cooperative intrusion detection framework. In Proceedings of the 2002 IEEE symposium on security and privacy, p. 202–215. Oakland, California, 2002.

[18] ZHOU, Jinming, HECKMAN, Mark, REYNOLDS, Brennen, CARLSON, Adam and BISHOP, Matt. Modeling network intrusion detection alerts for correlation. ACM Trans. Info. Syst. Sec. 10, 1, Article 4. 2007.

[19] WAN, Li and SHNGFENG, Tian. An ontology-based intrusion alerts correlation system. Journal of Expert Systems with Applications. ISSN 0957-4174 Elsevier. Volume 37, Issue 10, p. 7138-7146, 2010.