Artículo de Investigación Revista Ciencia e Ingeniería. Vol. 30, No. 3, pp. 183-192, agosto-noviembre, 2009. ISSN 1316-7081

Sistema multiagente para el manejo de incidentes de seguridad

Multiagents systems for the management of security incidents

Aguilar, José* y Abraham, Blanca
CEMISID, Departamento de Computación
EISULA, Universidad de Los Andes
Mérida, 5101-Venezuela
*aguilar@ula.ve

Recibido: 10-05-2008 Revisado: 03-09-2009

Resumen

Este trabajo tiene como propósito implementar los servicios básicos de manejo de incidentes para un CERT. Para ello, se desarrolla un sistema que mantiene un repositorio centralizado de información sobre incidentes de seguridad informática. El sistema esta compuesto de agentes que realizan búsquedas a través de Internet, de incidentes que hayan ocurrido en otros sitios, para conocer las formas de respuestas a estos. Dichos agentes utilizan un mecanismo de selección de los incidentes de interés para el CERT, basados en algoritmos de inteligencia artificial colectiva.

Palabras clave: Manejo de incidentes, sistemas multiagentes, seguridad informática.

Abstract

This work proposes a service of incidents management for a CERT. We develop a system composed by a repository with information over computing security. In addition, the system has agents which search in Internet incidents to know the response to them. These agents use a selection mechanism based on swarm intelligence.

Key words: Incidents management, multiagents systems, computing security.

1 Introducción

En la actualidad, la necesidad de seguridad es un factor muy importante en todos los ámbitos de las tecnologías de información. Cualquier dispositivo conectado a Internet corre el riesgo de ser atacado desde computadores, servidores, PDA's, routers, hasta teléfonos celulares. Específicamente, desde que Internet fue aceptado como el medio de interconexión global, una gran cantidad de transacciones de negocios se realizan por este medio, por lo cual son necesarios mecanismos de seguridad para protegerlas.

Así, la necesidad de dar una respuesta rápida a incidentes de seguridad es un aspecto fundamental en cualquier organización, para evitar pérdidas irreversibles. Por esta razón, se han creado grupos, denominados CERT (Equipo de Respuesta a Emergencias Informáticas, pero cuyas siglas vienen del ingles), distribuidos a nivel mundial. El primer CERT que se creó fue a finales de los años 80, en la Universidad de Carnegie Mellon. Era un grupo formado en su mayor parte por voluntarios calificados de la comunidad de informática, cuyo objetivo principal era facilitar una respuesta rápida a los problemas de seguridad que afectaban a Internet CERT; (Killcrece y col., 2003; Rueffle y col., 2003).

La presente investigación tiene como propósito implementar los servicios básicos de manejo de incidentes para un CERT. Para ello se desarrolla un sistema que mantiene un repositorio centralizado de información sobre incidentes de seguridad informática. El sistema esta compuesto por agentes que realizan búsquedas a través de Internet, de incidentes que hayan ocurrido en otros sitios, para conocer las formas de respuestas a estos. Dichos agentes utilizan un mecanismo de selección de aquellos incidentes que les interesan. El mecanismo esta basado en algoritmos de inteligencia artificial colectiva que tratan de emular el comportamiento observado en las colonias de hormigas (Abraham y col., 2006; Abraham y col., 2007; Aguilar y col., 2001).

2 Aspectos teóricos

2.1 Incidentes de seguridad informática

En general, los diferentes ataques que sufren los sistemas conectados a Internet son conocidos como incidentes de seguridad informática. Estos son una amenaza para el buen funcionamiento de cualquier organización, y son considerados como el acto de violar, implícita o explícitamente, las políticas de seguridad. Existe una gran variedad de amenazas, entre las cuales podemos citar (Killcrece y col., 2003; Rueffle y col., 2003): Intentos (exitosos o no) de ganar acceso sin autorización, a un sistema o a sus datos; Interrupciones no deseadas o denegación de servicio; Uso desautorizado de un sistema para procesar o almacenar datos; Cambio en las características del hardware, firmware o software del sistema; o instalación de software malicioso, sin el consentimiento o conocimiento del propietario.

Así, los incidentes de seguridad informática son cualquier evento que sea considerado una amenaza para la seguridad de un sistema, y se clasifican en manuales y automáticos. Los automáticos son aquellas herramientas de software que, sin interacción del usuario, ejecutan alguna operación para desequilibrar el funcionamiento de un sistema de computación. Entre estos tipos de incidentes se encuentran los conocidos virus, gusanos y troyanos. Otro grupo de este tipo de incidentes son los estáticos, es decir, los que no se reproducen, entre estos tenemos: bombas lógicas, denegación de servicio (DdoS por sus siglas en ingles), entre otros. Los incidentes manuales ocurren de manera intencional, cuando un atacante desea irrumpir en un sistema informático violando las restricciones de seguridad que este posea. Entre los ataques manuales mas conocidos encontramos: escaneo de vulnerabilidades, inyección SQL, hacking, cracking, ingeniería social, entre otros.

Cuando ocurre un problema de seguridad, es muy importante que la organización afectada tenga una forma rápida y efectiva de responder. La rapidez con la que una organización pueda reconocer un incidente o ataque, y luego, de manera exitosa analizarlo y generar una respuesta, limitará dramáticamente el daño causado y reducirá el costo de recuperación que dicho incidente acarrea (Rueffle y col., 2004).

A nivel mundial se han creado grupos de apoyo para el manejo de incidentes informáticos, los cuales son los encargados de combatir dichas amenazas y brindar soporte técnico a las entidades adscritas a ellos, además de prevenirlas ante posibles ataques. Dichos grupos son denominados CERT (siglas en ingles para: Computer Emergency Response Team) (Killcrece y col., 2003). Por lo general, estos equipos están coordinados y en constante colaboración entre ellos.

Killcrece, y otros, en (Killcrece y col., 2003), plantean los pasos a seguir para el establecimiento de un equipo de respuesta a emergencias informáticas. Entre otras cosas, definen los procesos de manejo de incidentes, manejo de vulnerabilidad, diseminación de alertas de seguridad, manejo de herramientas de seguridad, auditoría, detección de intrusos, análisis de riesgos, entrenamiento, consultoría, entre otros. Así, dentro de la estructura de un CERT debe existir un equipo que se encargue del manejo de incidentes. Las funciones de este equipo son:

- Detección y revisión de reportes: revisar los reportes de incidentes ocurridos a nivel mundial y detectar amenazas, para luego documentarlas.
- Análisis: es el intento por determinar que ha sucedido, que impacto o daño ha sido causado, y que pasos de mitigación o recuperación se deben tomar, para cada incidente ocurrido.
- Categorización y establecimiento de prioridades: es un procedimiento mediante el cual se hace una revisión y clasificación de los incidentes para establecer su gravedad, y de acuerdo a esta, asignar prioridades en las acciones a tomar.
- Respuesta a incidentes: son las acciones tomadas para mitigar o resolver el incidente, diseminar información de lo que se hizo, o implementar estrategias para que el incidente no ocurra de nuevo.

Existen pocas propuestas de CERT que han utilizado técnicas de inteligencia artificial en sus desarrollos, además, de forma distinta a la planteada en este trabajo. En la literatura los trabajos se orientan a proponer esquemas de reconocimiento de incidentes usando redes neuronales artificiales y sistemas inmunológicos artificiales (buscar semejanzas entre incidentes) (White, 2007), a desarrollar sistemas de gestión de bases de conocimiento de incidentes usando lógica difusa (Burch y col. 2008; White, 2007), a optimizar procesos de búsquedas de incidentes usando técnicas de computación evolutiva (Rueffle y col. 2003), entre otras. Nosotros proponemos usar la IAC para buscar respuestas a incidentes de seguridad desconocidos por el CERT.

2.2 Incidencia artificial colectiva (IAC)

Existen varios trabajos que estudian un tipo de conducta en una variedad de especies de animales, llamada "conducta colectiva" (Abraham y col., 2006; Abraham y col., 2007; Aguilar y col., 2001). Ejemplos de esa conducta son: una bandada de pájaros recorriendo el cielo, un grupo de hormigas en busca de comida, etc. Dichos trabajos han estudiado, entre otras cosas, cómo estos tipos de animales que actúan recíprocamente logran metas colectivas, evolucionan, etc. Estos trabajos han conformado el área llamada inteligencia colectiva (IC), la cual ha sido aplicada en problemas en las telecomunicaciones, robótica, transporte, etc. La idea principal de la IC sugiere que N agentes en una colonia cooperan mutuamente para lograr alguna meta. Los agentes usan reglas simples para gobernar sus acciones, y por medio de las interacciones del grupo entero logran sus objetivos. Un tipo de auto-organización surge de la colección de acciones del grupo. La IC resuelve problemas de

manera flexible, adaptativa y descentralizada.

Así, en la IC los agentes se ubican en grupos, llamados colonias, donde ellos hacen un trabajo cooperativo. Los agentes procesan información, modulan su conducta de acuerdo a estímulos, y toman la mejor decisión basada en la información del ambiente que les rodea. Pero el desafío más grande es hacer que los agentes trabajen de manera colectiva, que integren sus actividades individuales para generar resultados más complejos y eficaces.

En los estudios actuales de IC, la conducta inteligente surge frecuentemente a través de la comunicación indirecta entre los agentes. La fuente de inspiración son los sistemas de insectos. Individualmente, los insectos tienen comportamientos simples con memoria limitada. Sin embargo, colectivamente los insectos realizan tareas complicadas con un grado alto de consistencia. Algunos ejemplos de comportamiento sofisticado son: formación de puentes, construcción y mantenimiento de nidos, cooperación para cargar objetos grandes, búsqueda de la ruta más corta entre el nido y fuentes de alimentos, etc. En los modelos estudiados se han identificado dos tipos de comunicación indirecta, la primera involucra un cambio en las características físicas del ambiente. La construcción del nido es un ejemplo de esta forma de comunicación, en la que un insecto observa el desarrollo de la estructura y agrega su pelota de barro a la cima de ella. La segunda esta "basada en señales". En este caso algo es depositado en el ambiente que no hace ninguna contribución directa a la tarea, pero se usa para influir en la conducta subsiguiente. La comunicación indirecta basada en señales esta muy desarrollada en las hormigas. Las hormigas usan un químico muy volátil, llamado "feromona", para proporcionar un sistema de señalización sofisticado. La IC ha inspirado técnicas como optimización colectiva de partículas, sistemas artificiales de hormigas, modelos ecológicos, entre otros (Aguilar y col., 2001).

2.3 Teoría de agentes

¿Qué es un agente?, la respuesta a esta pregunta es compleja ya que los investigadores utilizan distintas acepciones de dicho término, y no existe una definición académica ampliamente aceptada. Pero daremos algunas (Jennings y col., 1998; Nwana, 1996; Nilsson, 2001):

- "Un agente es un programa de ordenador que actúa autónomamente en nombre de una persona u organización".
- "Un agente puede verse como aquello que percibe su entorno a través de sensores y que actúa sobre él mediante efectores".

Hay multitud de clasificaciones de agentes que dependen del punto de vista del investigador. Aunque podemos considerar la más común la realizada a partir de la enumeración de las características que cumple un agente, las cuales son:

- Autonomía: los agentes son autónomos en la medida en que actúan sin la intervención humana ni de otros sistemas externos. Es la capacidad que tiene un agente de tener un comportamiento propio, y reaccionar a los estímulos externos basándose en su estado interno. Cada agente recibe y percibe señales del ambiente o de otros agentes, las analiza utilizando sus mecanismos internos, que pueden ser desde sencillas sentencias *sí/entonces* hasta complejos sistemas expertos dinámicos que utilizan reglas difusas.
- Comunicación: es la capacidad de cada agente de conversar utilizando un lenguaje basado en ontologías, y realizar intervenciones asíncronas. Constituye un paso adelante en llevar el concepto real de conversación al ámbito computacional. Una ontología es una colección de conceptos, predicados, secuencias, términos y relaciones entre estos elementos, que son entendibles por una sociedad de agentes. Cada agente comprende esta intervención, en consecuencia, afecta su estado interno y, eventualmente, reacciona con un conjunto de intervenciones. Existe otro tipo de comunicación, que es indirecta, que no está basada en el pase de mensajes. Este tipo de comunicación deriva de las investigaciones en IC, está basada en variables compartidas por todos los agentes, y por mediciones en el ambiente donde estos se desenvuelven. Esto le permite al agente Socializar o cooperar.
- Reactividad: la capacidad de emitir una acción inmediata al recibir una señal o percibir un estado en el ambiente, es lo que caracteriza a los agentes reactivos. Los agentes, por lo general, no reaccionan de inmediato, ya que deben procesar la información y "pensar" sus acciones.
- Proactividad: toma la iniciativa para alcanzar sus objetivos.
- Inteligencia: generalmente, la cualidad de inteligencia es asociada directamente con el concepto de agente. Debido a que un agente debe analizar y tomar una acción de forma autónoma, es necesario implementar esta característica utilizando alguna tecnología o técnica computacional, para lo cual generalmente se utilizan técnicas inteligentes (redes neuronales artificiales, etc.). El Aprendizaje en este caso es entendido como el comportamiento basado en la experiencia previa.
- Carácter: inclusión de estados de creencia, deseo e intención (modelo BDI, Nilsson, 2001).
- Movilidad: es la capacidad que tiene un agente de mover su estado y código de ejecución de un nodo a otro en un sistema distribuido. Esta capacidad posibilita una computación menos centralizada y más distribuida. Un agente puede alojarse en cualquier nodo y realizar sus tareas utilizando los recursos locales, para después volver a su nodo origen llevando la información procesada.

Un ejemplo de clasificación de agentes es la siguiente (Jennings y col., 1998; Nwana, 1996; Nilsson, 2001).

Agente cognitivo, Agente reactivo y Agente deliberativo.

Normalmente los agentes no actúan en solitario, sino que se localizan en entornos o plataformas con varios agentes, donde cada uno de ellos tiene sus propios objetivos, toma sus propias decisiones, y puede tener la capacidad de

comunicarse con otros agentes. Dichos entornos se conocen con el nombre de Sistema Multiagentes (SMA).

Los SMA se caracterizan por la interacción de varios agentes en un mismo entorno, ya sea este físico o virtual. Un concepto principal de los SMA es la interacción y coordinación entre agentes, que no se limita a la comunicación ni al envío de mensajes entre estos, sino a la forma como un agente se relaciona con otros agentes.

3 Desarrollo de la propuesta

Un proyecto CERT debe contar con una herramienta que pueda ser utilizada para determinar que tipo de incidente esta ocurriendo en un sistema que este siendo atacado, en base a los síntomas presentados. Esto permite proporcionar soluciones inmediatas y ejecutar medidas automáticas de respuesta para controlar dicho incidente (Dorofee y col., 2004). Para esta problemática se propone el desarrollo de un sistema que realice los servicios de manejo de incidentes para un proyecto CERT. Este sistema debe ser capaz de aportar la mejor solución para tratar un incidente, en base a soluciones encontradas en la web. En nuestro caso, nosotros proponemos usar "agentes de búsqueda", los cuales estarán localizando constantemente información sobre incidentes en la red, para almacenarla en un repositorio. Todo esto le permite al sistema mantenerse actualizado. En este trabajo se propone un algoritmo de inteligencia artificial colectiva que será usado para la selección de los incidentes de interés para el CERT. Igualmente, se propone una forma de estandarizar la información de incidentes de seguridad en un archivo XML (Wikipedia, 2007), que será la información a almacenar en el repositorio de incidentes.

3.1 Formato de archivos XML usado

El sistema hará uso de archivos XML (ver Tabla 1) para almacenar la información que caracteriza a cada incidente. Estos archivos contendrán información, tanto estática como dinámica, para describir a los incidentes. Entre la información estática se incluyen datos como: nombre, descripción, tipo, fecha de descubrimiento, entre otros. Entre el grupo de datos dinámicos se tiene información relacionada a aspectos técnicos como: las plataformas que afecta, nivel de peligrosidad, nivel de daño, síntomas, entre otros. Además, se incluye una nueva variable que está asociada al mecanismo de selección de incidentes, llamada "feromona".

Tabla 1. Descripción de etiquetas XML

<incident_listing>	Indica el comienzo de la lista de incidentes.
<incident>	Indica el comienzo de la información del incidente de seguridad informática.
<general_information>	Contiene información general del incidente. Dentro del rango de esta etiqueta se encuentran las etiquetas: incident_id, incident_name, etc.
<incident_id>	Especifica un identificador para cada incidente.
<incident_name>	Contiene la información del nombre del incidente.
<incident_type>	Dentro de esta etiqueta se especifican los distintos tipos de incidentes de seguridad informática, tales como: virus, troyanos, gusanos, spyware, adware, hoaxes, spam, entre otros.
<date_discovered>	Indica la fecha de cuando fue descubierto dicho incidente por primera vez.
<date_updated>	Indica la fecha de la última vez en la que fue modificada la información del incidente.
<affected_platform>	Especifica las plataformas informáticas o sistemas operativos en los cuales el incidente causa efecto.
<desc_full>	Contiene información descriptiva y detallada del incidente.
<detalles>	Dentro de esta etiqueta se presenta información detallada acerca del incidente.
<method_infected>	Especifica el método de infección de incidente, es decir, como el incidente lleva a cabo la infección de un sistema informático.
<sintoms>	Detalla los síntomas más comunes que presentan los sistemas informáticos cuando son afectados por este incidente.
<other>	Indica el inicio de etiquetas opcionales de información de los incidentes. Dentro de estas etiquetas se encuentran method_distribution, efects y more.
<method_distribution>	Especifica el método de propagación del incidente.
<efects>	Detalla los efectos que acarrea la presencia del incidente en un sistema informático.
<more>	Contiene información adicional del incidente.
<solution>	Dentro de esta etiqueta se presentan varias etiquetas que especifican formas de solucionar el incidente. Contiene a las etiquetas removal, protected y know.
<removal>	Especifica la manera o maneras de eliminar el incidente de un sistema informático.
<protected>	Presenta información sobre como prevenir a un

	sistema de ser atacado por este incidente.
<know>	Contiene información adicional sobre formas de determinar si un sistema informático esta siendo afectado por dicho incidente.
<incident_wild_level>	Especifica el nivel de peligrosidad del incidente.
<incident_damage_level>	Especifica el nivel de daño que puede causar el incidente.
<distribuction_level>	Especifica el nivel de propagación del incidente.
<url_homepage>	Contiene la información correspondiente a la dirección electrónica de donde se obtuvo la información del incidente.
<source>	Especifica la fuente donde se obtuvo la información acerca del incidente.
<pheromone>	Esta etiqueta contiene información utilizada por los agentes de búsqueda y selección desarrollados en base a inteligencia artificial colectiva.

3.2 Algoritmo de inteligencia artificial colectiva

El algoritmo de selección está inspirado en el uso de "trazas de feromona", que le permiten identificar los incidentes mejor adaptados a los requerimientos de búsqueda (derivados de los incidentes que frecuentemente ocurren en los entes adscritos al CERT), a través de un proceso de retroalimentación positiva y negativa (Abraham y col., 2006; Abraham y col., 2007; Aguilar y col., 2001).

3.2.1 Modelo de Selección

El sistema utilizará tantos agentes como desee. Las etapas que componen el proceso de selección son las siguientes:

- Cada agente realiza un recorrido, independiente al seguido por el resto de los agentes, por repositorios sobre la web, buscando incidentes de acuerdo a las características solicitadas (requerimientos de búsqueda).
- La trayectoria seguida en cada recorrido consistirá en visitar, aleatoriamente, 1, 2 ó N repositorios, que contienen incidentes y sus posibles soluciones.
- Al final de la trayectoria seguida, en cada uno de los recorridos realizados, el agente habrá acumulado un grupo de incidentes, llamado "cc", que son candidatos a ser seleccionados.
- Finalmente, cada agente realiza la selección del incidente encontrado que mejor se adapte a los requerimientos de búsqueda, desde el grupo de incidentes "cc" conseguido.

Luego de que cada agente creado complete todo el proceso de selección y tenga a disposición todos los incidentes requeridos, cada uno de los ellos procederá a actualizar la traza de feromona de los incidentes encontrados. La decisión sobre realizar la inserción de dichos incidentes al repositorio del CERT local que hace la búsqueda, esta a cargo del administrador del CERT. El administrador selecciona, según algún criterio, los incidentes que requiere del grupo de incidentes seleccionado por cada agente, y los incorpora al repositorio. Las premisas generales para el modelo se muestran a continuación:

- Se supone que cada incidente cuenta con un archivo XML que lo caracteriza.
- Se asume que se tiene la información exacta de los incidentes que se desean buscar.

La fórmula para establecer el grado de correspondencia entre un incidente buscado y un incidente encontrado, viene dada por:

$$X_{lj}{}^{i} = 1 + \sum H_i \text{ . } \sum N_{lj} \quad (1)$$

donde Xlji es el grado de correspondencia entre el incidente ideal i, y el incidente j, ubicado en el repositorio l. ∑Hi identifica la sumatoria de las características que debe tener el incidente i que se está buscando, por ejemplo: las plataformas que afecta, síntomas presentados, entre otras. Por otro lado, ∑Nlj representa la sumatoria de las características reales encontradas, similares a las deseadas, del incidente preseleccionado (candidato). Mientras más cercano a 1 sea el valor de Xlji, mayor será la similitud entre las características ideales y las características reales del incidente j encontrado. Cada agente evaluará un conjunto de incidentes de seguridad, por cada incidente i requerido, considerando:

- El valor ideal de Xlji es 1, el agente escogerá incidentes cuya correspondencia entre el incidente deseado i y el encontrado j sea cercano a ese valor.
- El monto de feromona Ylj(t) relacionado a cada incidente j, ubicado en el repositorio l, que conforman cada uno de esos conjuntos, será actualizado.

La ecuación de transición que calcula la probabilidad de que un agente k seleccione un incidente j, ubicado en el repositorio l, de entre un grupo cc_{ik} de posibles incidentes a seleccionar, es [6, 13]:

$$P_{lj}^{ik}(t) = \left[Y_{lj}(t)\right]\left[X_{lj}^{i}\right] / \sum_{rsn \in cc_{ik}} \left[Y_{rsn}(t)\right]\left[X_{rsn}^{i}\right] \quad (2)$$

donde $Y_{lj}(t)$ representa la cantidad de feromona para el incidente j que ha sido encontrado por el agente k en el repositorio l. Los agentes son creados e inician el proceso de selección al azar. Es importante notar que el valor de la probabilidad $P_{lj}^{ik}(t)$ pudiera ser diferente para dos agentes evaluando un mismo incidente, ya que esta depende del grupo de incidentes cc_{ik} que cada agente ha encontrado.

Como se dijo anteriormente, cada agente k deposita

una cantidad de feromona $\Delta Y_{lj}(t)$ en cada uno de los incidentes conseguidos, el cual es el producto inverso de la correspondencia Xljk.

$$\Delta Y_{lj}(t) = (X^{k}_{lj})^{-1} \quad (3)$$

En la presente propuesta es necesario aplicar la retroalimentación positiva y negativa. Esta última se hace a través de la tasa de evaporación de feromona, ya sea en el incidente seleccionado o no. En general, la retroalimentación positiva y negativa es realizada a través de la actualización de la traza como sigue (Abraham y col., 2006; Abraham y col., 2007; Aguilar y col., 2001):

$$Y_{lj}(t) = (1-\alpha) * Y_{lj}(t) + \Delta Y^{k}_{lj}(t) \quad (4)$$

El monto inicial de feromona de los incidentes se asume como un número aleatorio. α es una constante que controla la tasa de evaporación de feromona. Finalmente, para determinar el incidente "i" a seleccionar se define la siguiente regla de transición:

$$S^*_{ki} = \begin{cases} \arg\max_{rsn \in CC} \{P_{rsn}^{ik}\} \\ J \quad \text{(Valor Aleatorio)} \end{cases} \quad (5)$$

donde el valor de P^{ik}_{rsn} viene dado por la ecuación 1. Además, S^*_{ki} es el incidente i seleccionado por el agente k de un grupo de incidentes (cc_{ik}), mientras que J es un incidente seleccionado aleatoriamente.

3.2.2 Macro algoritmo

- Definir e identificar el perfil deseado de los incidentes a buscar (requerimientos de búsqueda).
- Crear k agentes de búsqueda.
- Realizar la búsqueda y selección de los incidentes requeridos usando el proceso "selección" antes explicado (cada agente lo hace).
- Actualizar la traza de feromona para cada incidente de los distintos cc_{ik} .
- Seleccionar un incidente por incidente requerido, de entre la selección hecha por los distintos agentes, e ingresarlo al repositorio de incidentes.
- Un agente podría buscar varios incidentes en un mismo proceso de búsqueda.

3.3 Descripción general del sistema

El sistema realizará la búsqueda de información de incidentes y soluciones de seguridad informática utilizando un algoritmo de selección basado en inteligencia artificial colectiva. El sistema sirve de apoyo a un CERT en el proceso de manejo y respuesta a incidentes, gracias al aporte de información actualizada sobre la manera de enfrentar a los incidentes ocurridos en las instituciones registradas en el CERT. En líneas generales el sistema permite:

- Realizar la búsqueda de información sobre incidentes de seguridad informática ubicados en repositorios web, usando agentes de búsqueda.
- Agilizar el proceso de respuesta a incidentes.
- Manejar un repositorio de archivos XML con información de incidentes.

3.3.1 Macro algoritmo

El cliente se dispone a utilizar la librería de manejo de incidentes. Esta librería exporta un conjunto de métodos que son utilizados para manejar la información de los incidentes. De estos métodos, el cliente puede elegir alguno de los que a continuación se explican:

- Buscar incidentes en los repositorios.
- Registrar incidentes en el repositorio local.
- Modificar incidentes en el repositorio local
- Consultar incidentes en el repositorio local.
- Eliminar incidentes en el repositorio local.

Si elije "Buscar Incidentes en los Repositorios", la librería se encarga de activar el siguiente proceso:

- Se agrupan varios de los atributos para formar un perfil de incidente deseado.
- Se organizan los elementos necesarios para activar los agentes que realizarán la búsqueda.
- Se activan los agentes que buscarán en los repositorios. Los agentes procederán a buscar y seleccionar los incidentes a partir de las características especificadas.

Si elije "registrar incidentes", el cliente deberá ingresar los campos de la manera correcta. Una vez realizado esto, la librería se encarga de insertarlos en el repositorio.

Si elije "modificar incidentes", el cliente deberá ingresar los campos de la manera correcta para consultar los datos. La información será mostrada en el cliente y este podrá modificar los que desee.

Si elije "Consultar incidentes", él tiene los siguientes criterios de búsqueda en sus opciones:

- Búsqueda por fecha y nombre.
- El cliente debe ingresar la fecha y el nombre del incidente.
- La aplicación le devuelve el resultado obtenido.
- Búsqueda por fecha, nombre, nivel de daño y nivel de peligrosidad.
- El cliente debe ingresar la fecha, el nombre, el nivel de peligrosidad y el nivel de daño del incidente.
- La aplicación le devuelve el resultado obtenido.

Si elije "eliminar incidentes", el cliente deberá ingresar los campos de la manera correcta para consultar los datos. La información será mostrada y el cliente decidirá si realiza la eliminación.

3.3.2 Pantallas de aplicación

Se presentarán a continuación algunas pantallas de interfaz gráfica del sistema. La aplicación se desarrollo como una librería en C++, que puede ser utilizada por cualquier tipo de aplicación de consola, gráfica o web. También, hace uso de un tipo de repositorio estandarizado basado en archivos XML, que contiene la información relevante a incidentes de seguridad informática. De igual manera, el sistema permite el acceso a una base de datos PostgreSQL a través de la librería libpq-fe, la cual pudiese contener la información de incidentes reportados por las instituciones registradas en el CERT – Venezuela. La Fig. 1 muestra la pantalla inicial del sistema, con las operaciones que se pueden hacer desde él. A continuación se presenta la pantalla donde se introduce la información para iniciar la búsqueda de incidentes Fig. 2, insumo necesario para que los agentes de búsqueda puedan realizar ese proceso. La Fig. 3 muestra la pantalla de consulta al repositorio de incidentes del CERT local, y la Fig. 4 la pantalla para introducir nuevos incidentes en él.

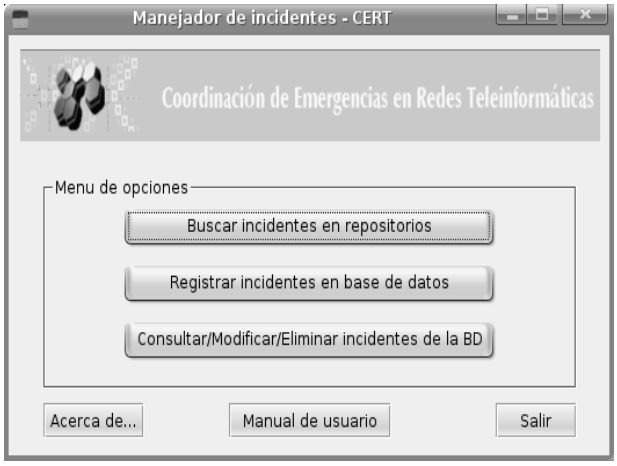

Fig 1. Pantalla inicial

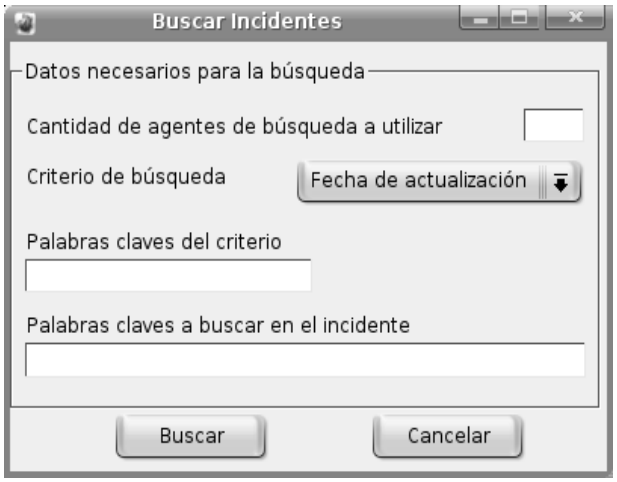

Fig. 2. Formulario para realizar la búsqueda de incidentes

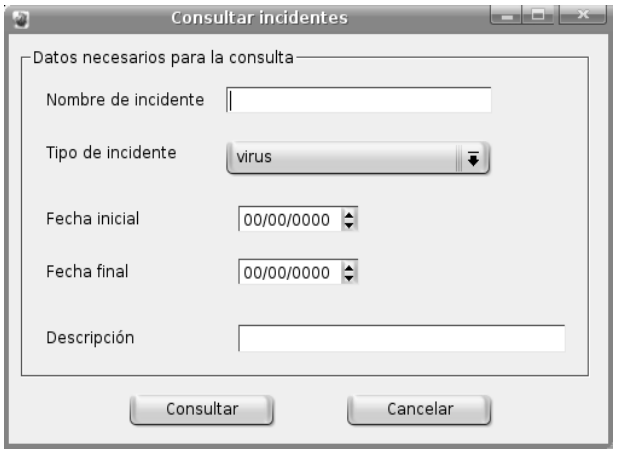

Fig. 3. Formulario para consultar incidentes en la base de datos

Fig. 4. Formulario para registrar incidentes en la base de datos

4 Experimentos

4.1 Descripción de los experimentos

Las pruebas se realizan con tres (3) repositorios que almacenan información de incidentes, recopiladas de fuentes de internet como panda (www.pandasoftware.com), symantec (www.symantec.com) y mcafee (us.mcafee.com). Los archivos XML contenidos en estos repositorios son construidos en base al diseño propuesto en etapas anteriores. Estos repositorios serán ubicados localmente, en donde podrán ser manipulados por la aplicación.

4.1.1. Experimento 1

Se procedió a ejecutar la aplicación en tres (3) oportunidades buscando un (1) incidente diferente en cada corrida, variando los criterios de búsqueda, pero manteniendo el

número de agentes en uno (1). Los incidentes a buscar son los especificados en la tabla 2. De esta manera se puede verificar la funcionalidad de los repositorios XML y de los agentes en la búsqueda y selección a partir de un perfil dado. Para este experimento, el agente buscará al incidente requerido en todos los repositorios. Así mismo, la evaluación de los incidentes se considera como un valor aleatorio decimal entre cero (0) y uno (1), esta evaluación significa en que medida la información recopilada por los agentes se corresponde con la solicitada por el usuario.

Tabla 2. Descripción de incidente requerido

Incidente a buscar #1	
Nombre	Exploit-ANIfile
Plataforma que afecta	Windows
Síntomas presentados	System crashing unexpectedly
Tipo de incidente	Troyano
Palabras descriptivas	exploit a microsoft windows kernel ANI file parsing vulnerability
Incidente a buscar #2	
Nombre	Trj/Gagar.CC
Plataforma que afecta	Windows 2003/XP/2000/NT
Síntomas presentados	no muestra mensajes o avisos que alerten sobre su presencia
Tipo de incidente	Troyano
Palabras descriptivas	descarga un archivo que corresponde al troyano Alanchum.MU
Incidente a buscar #3	
Nombre	W32.Sagevo
Plataforma que afecta	Windows 2000/95/98/Me/NT/2003/XP
Tipo de incidente	Gusano
Palabras descriptivas	spreads by exploiting the Symantec Client Security and Symantec AntiVirus Elevation of Privilege

Los datos de los incidentes extraídos de Symantec y Mcafee se especifican en ingles debido a que los archivos XML que contienen la descripción de estos se encuentran en ese idioma. Los archivos XML extraídos de Panda contienen su descripción en español. Para el inicio de las pruebas se determina que el valor inicial de la feromona sea uno (1) para todos los incidentes.

4.1.2. Experimento 2

Se procedió a ejecutar el sistema en 4 oportunidades consecutivas, variando el número de agentes, y buscando los dos (2) incidentes explicados en la tabla 3. Para este experimento, la asignación de los repositorios fue aleatoria, lo que permitió realizar búsquedas independientes en distintos repositorios. De la misma manera que en el experimento #1, la evaluación de los incidentes se considera como un valor aleatorio decimal entre cero (0) y uno (1), esta evaluación significa en que medida la información recopilada por los agentes se corresponde con la solicitada por el usuario.

Tabla 3. Descripción de incidente requerido

Incidente a buscar #1	
Nombre	Exploit-ANIfile
Plataforma que afecta	Windows
Síntomas presentados	System crashing unexpectedly
Tipo de incidente	Troyano
Palabras descriptivas	exploit a Microsoft windows kernel ANI file parsing vulnerability
Incidente a buscar #2	
Nombre	MalwareAlarm
Plataforma que afecta	XP
Síntomas presentados	Aparece un icono en forma de alarma en la bandeja del sistema
Tipo de incidente	Spyware
Palabras descriptivas	es un programa de tipo adware que intenta engañar al usuario

El orden de ejecución del sistema, junto con los parámetros asignados en cada caso, se puede ver en la tabla 4.

Tabla 4. Ciclo de corridas

Incidente 1 y 2	# de agentes a utilizar
Primera corrida	2
Segunda corrida	3
Tercera corrida	4
Cuarta corrida	6

4.2. Análisis de los resultados

Antes de presentar los resultados obtenidos, a continuación se describe la nomenclatura utilizada:

- Arch/Prob/Pher/Rep = Arch indica el nombre del archivo seleccionado por el agente, junto con Prob que corresponde a la probabilidad de selección, Pher la cantidad de feromona depositada y Rep que indica el repositorio donde fue ubicado.

Es de importancia saber que, además de los experimentos que a continuación se detallan, previamente se realizaron pruebas para comprobar el comportamiento esperado en los agentes, es decir, que su comportamiento fuera acorde a lo establecido en el diseño del algoritmo de búsqueda y selección, y así saber que no se cuenta con fallas funcionales en el sistema. En estas pruebas, a diferencia de las otras, se establecieron los perfiles de componentes de software deseados (especificaciones técnicas y funcionales), y se identificaron su presencia en algunos de los repositorios. Después, se inicio el proceso búsqueda. Al final de este proceso se constato si los perfiles seleccionados por los

agentes fueron los ideales previamente identificados. Esto nos permitió comprobar si el algoritmo siguió el comportamiento esperado. Todos los resultados obtenidos se muestran en (Aguilar y col., 2006). Aquí solo analizaremos los de los dos experimentos antes explicados.

4.2.1. Experimento 1

En la tabla 5 se muestran los mejores perfiles encontrados de acuerdo al criterio de probabilidad, asociado al incidente requerido, y el repositorio de donde provienen.

Tabla 5. Resultados de la búsqueda para el experimento 1

Incidente #1	Incidente #2	Incidente #3
Arch/Prob/Pher/Rep	Arch/Prob/Pher/Rep	Arch/Prob/Pher/Rep
3.xml/1/14.4499/1	1.xml/0.2/2.1352/2 5.xml/0.2/1.2906/2 2.xml/0.13/2.7220/2 7.xml/0.13/3.3112/2	8.xml/1/1.94579/3

Al analizar los resultados obtenidos, se observa que al buscar el incidente #1, el agente selecciono al incidente que corresponde al archivo "3.xml" con probabilidad 1, del repositorio McAfee, con un valor de feromona final de 14.4499 luego de su selección. La selección fue perfecta en cuanto a la correspondencia entre el perfil dado y el encontrado. Es importante aclarar que no seleccionó otro incidente debido a que no encontró otro que tuviera una correspondencia cercana al perfil deseado.

En cuanto al incidente #2, el agente seleccionó 4 incidentes del repositorio Panda que concuerdan con el perfil deseado, cada uno con una probabilidad de selección diferente y con feromona inicial de uno (1). El primer incidente seleccionado corresponde al descrito por el archivo "1.xml", con una probabilidad de selección de 0.2, con un valor de feromona final de 2.1352. El segundo corresponde al descrito por el archivo "5.xml", con 0.2 de probabilidad, con un valor de feromona final 1.2906. De la misma manera, el tercer incidente seleccionado corresponde al archivo "2.xml", con probabilidad de 0.1333, con un valor de feromona final de 2.72204. Finalmente, el último incidente seleccionado es descrito por el archivo "7.xml", con probabilidad de selección de 0.1333, con un valor de feromona final de 3.31123.

Continuando con el incidente buscado #3, se observa que el agente selecciono al incidente que corresponde al archivo "8.xml" del repositorio Symantec, con un valor de feromona final de 1.94579. En este caso, al igual que en la búsqueda del incidente #1, la selección fue perfecta en cuanto a la correspondencia entre el perfil dado y el encontrado, por esta razón no se realizo la selección de otro incidente.

4.2.2. Experimento 2

Los resultados obtenidos en el experimento 2 se muestran en la tabla 6.

Tabla 6. Resultados de la búsqueda para el experimento 2

	Corrida #1	
	Incidente 1	Incidente 2
	Arch/Prob/Pher/Rep	Arch/Prob/Pher/Rep
agente #1	3.xml/1/4.78585/1	No encontrado
agente #2	No encontrado	9.xml/1/14.5226/2
	Corrida #2	
	Incidente 1	Incidente 2
	Arch/Prob/Pher/Rep	Arch/Prob/Pher/Rep
agente #1	3.xml/1/6.48242/1	No encontrado
agente #2	No encontrado	9.xml/1/8.3831/2
agente #3	No encontrado	No encontrado
	Corrida #3	
	Incidente 1	Incidente 2
	Arch/Prob/Pher/Rep	Arch/Prob/Pher/Rep
agente #1	3.xml/0.88609/14.8786/1 7.xml/0.06834/2.88003/1 9.xml/0.04556/1.72975/1	No encontrado
agente #2	No encontrado	9.xml/1/7.89194/2
agente #3	No encontrado	No encontrado
agente #4	No encontrado	No encontrado
	Corrida #4	
	Incidente 1	Incidente 2
	Arch/Prob/Pher/Rep	Arch/Prob/Pher/Rep
agente #1	3.xml/0.851272/16.3263/1 9.xml/0.090265/9.86651/1 7.xml/0.036149/3.03881/1 4.xml/0.022311/1.28494/1	No encontrado
agente #2	No encontrado	9.xml/0.7978/10.6697/2 3.xml/0.1010/1.65539/2 7.xml/0.1010/5.7619/2
agente #3	3.xml/0.712473/16.1563/1 9.xml/0.215285/9.4443/1 7.xml/0.044204/3.86004/1 4.xml/0.028037/2.34125/1	No encontrado
agente #4	No encontrado	7.xml/0.5103/8.65146/2 9.xml/0.4269/10.6699/2 3.xml/0.0627/1.02155/2
agente #5	3.xml/0.489468/9.04667/1 9.xml/0.391458/18.5159/1 4.xml/0.067266/3.42173/1 7.xml/0.067266/4.03210/1	No encontrado
agente #6	9.xml/0.4334/14.72142/1 3.xml/0.4235/8.14648/1 4.xml/0.0800/4.69445/1 7.xml/0.0629/4.77927/1	No encontrado

Los resultados del experimento 2 demuestran que el comportamiento de los agentes observado en el experimento 1 es consistente en este experimento. Otra característica que se observa en el sistema de selección propuesto es el hecho de que varios agentes, relacionados a una misma corrida, presentan resultados similares en la evaluación de los incidentes encontrados. Tal es el caso de los agentes 1 y 3 de la cuarta corrida (ver tabla 6), los cuales obtuvieron los mismos resultados asociados al Incidente #1. Esto es debido

a que ambos siguieron una misma ruta de búsqueda del incidente solicitado, situación posible ya que los agentes actúan de manera autónoma al decidir en qué o en cuáles repositorios buscar los incidentes solicitados. Además, algo importante para resaltar es que el agente 1 de la tercera corrida, y los agentes 1, 3, 5 y 6 de la cuarta corrida (ver tabla 6), visitaron solo un repositorio (en este caso Repositorio 1, Mcafee) para buscar al Incidente #1, formando un mismo grupo de incidentes encontrados.

También importante a resaltar es que la mayoría de los agentes, al buscar los incidentes #1 y #2, realizaron la selección del mismo grupo de incidentes, siendo los incidentes con mayor probabilidad de selección los descritos por los archivos "3.xml" y "9.xml", respectivamente. Sin embargo, los resultados obtenidos de la evaluación hecha por estos agentes sobre estos incidentes difieren entre las corridas, debido a que al finalizar un proceso de búsqueda y selección un agente, todos los incidentes encontrados por él son sometidos al proceso de actualización de su traza de feromona. Esta actualización de feromona va influenciando el proceso de selección de incidentes en corridas consecutivas/paralelas de los agentes.

Otro punto importante es que los incidentes seleccionados por la mayoría de los agentes pertenecen a un repositorio específico, esto es debido a que un mismo incidente no se encuentra registrado en varios repositorios. Si se diera ese caso, la selección se realizaría en todos los repositorios en que dicho incidente fuera encontrado.

5 Conclusiones

Luego de realizar la investigación se llego a las siguientes conclusiones:

- Se requiere la creación de un repositorio de incidentes para un CERT que pueda llegar a convertirse en un estándar para intercambio y publicación de información de incidentes de seguridad informática. Para tratar de lograr esta estandarización, se plantea el uso de archivos XML que contengan información detallada de las características de los incidentes.
- Los agentes deben garantizar conseguir todos los incidentes solicitados, buscándolos a través de todos los repositorios disponibles.
- Gracias a la generalidad del algoritmo de búsqueda y selección de incidentes de seguridad, se concluye que este puede ser usado en la búsqueda y selección de otros elementos o recursos informáticos.
- Luego de realizar muchas corridas buscando un incidente específico, ocurre que el sistema de selección se vuelve determinista, arrojando como resultado, casi siempre, el mismo incidente. Esto es debido a que la marca de feromona de dicho incidente hace que la probabilidad de selección sea muy alta.

Cuando se usan muchos agentes en el programa, podría ocurrir el seguimiento de rastros no óptimos que llevarían a la selección de malas soluciones. De la misma manera, el uso de pocos agentes en el programa podría no producir el efecto de sinergia y cooperación esperado. Estas son propiedades propias de la IC, que por supuesto, se dan en nuestro sistema.

Agradecimientos

Este trabajo ha sido financiado por el CDCHT-ULA a través del proyecto I-820-05-02-AA, y el programa de cooperación Franco-Venezolano ECOS-NORD, a través del proyecto V04M01, a quienes se les extiende nuestro agradecimiento.

Referencias

Abraham B, Aguilar J y Bastidas J, 2006, Selection Algorithm Using Artificial Ant Colonies, WSEA Transactions on Computers, Vol. 5, No. 10, pp. 2197-2203.Abraham B y Aguilar J, 2007, Software Component Selection Algorithm Using Intelligent Agents", Lecture Notes in Artificial Intelligence, Springer-Verlag, Vol. 4496, pp. 82-91.

Aguilar J y Rivas F, (eds.), 2001, Introducción a la Computación Emergente, Editorial MERITEC.

Aguilar J, Abraham B, 2007, Aplicación para el manejo de incidentes del proyecto CERT Venezuela basado en teoría de agentes, Informe Técnico N. 10-2007, CEMISID-ULA.

Burch H, Manion A y Ito Y, 2008, Vulnerability Response Decision Asístanse, Technical Report (VRDA-2008.pdf, http://www.cert.org/).

Dorofee A, et all, 2004, Defining incident management process for CSIRT's, Technical Report, Computer Sciences Dept., University of Columbia.

Jennings J, Sycara L y Wooldridge M, 1998, A Roadmap of Agent Research and Development, Technical Report, Autonomous Agents and Multi-Agent Systems Institute.

Killcrece J, Kosakowsky K, et all., 2003, Organizational models for computer security incidents response teams (CSIRT's). Srpinger, USA.

Nilsson N, 2001, Inteligencia artificial: Una nueva sintesis. Editorial McGraw Hill.

Nwana L, 1996, Software Agents: An Overview". Knowledge Engineering Review, Vol. 7, No. 9, pp. 134-142.

Rueffle R, Killcreece J, et all, 2003, State of the practice of Computer Security Incident Response Team (CSIRT's), Carnegie Mellon Software Engineering Institute.

White D, 2007, CERT Resiliency Engineering Framework Technical Report (bits0703.pdf, http://www.cert.org/).

Wikipedia, XML. http://es.wikipedia.org/wiki/XML.