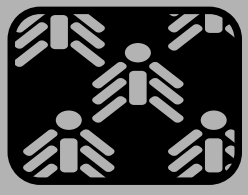
SISTEMA PARA EL MANEJO DE INCIDENTES DE SEGURIDAD INFORMATICA USANDO COLONIAS ARTIFICIALES DE HORMIGAS



AUTOR
Jose Aguilar
Doctor
Universidad de Los Andes, Facultad de Ingeniería, CEMISID
aguilar@ula.ve
VENEZUELA

AUTOR
Blanca Abraham
Msc.
Fundacite-Mérida
blanca@fundacite-merida.gob.ve
VENEZUELA

Fecha de Recepción: 21 de Julio de 2007
Articulo Tipo 1

Fecha de Aceptación: 11 de Agosto de 2007

RESUMEN.

Este trabajo presenta el desarrollo de un sistema que presta los servicios básicos de manejo de incidentes de seguridad informática. Particularmente, plantea el uso de agentes para realizar 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 búsqueda y selección de incidentes basados en los Sistemas Artificiales de Hormigas.

PALABRAS CLAVE

Seguridad Informática
Manejo de Incidentes
Sistemas Artificiales de Hormigas
Inteligencia Artificial Distribuida

ABSTRACT

This work presents the development of an informatic security incidents management system. Particularly, the system is composed by search agents over Internet, which find incidents in other sites in order to know the responses to these. These agents use a search and selection mechanism based on the ant artificial systems.

KEYWORDS

Informatic security
Incidents management
Ant artificial systems
Distributed Artificial Intelligence

INTRODUCCIÓN

En la actualidad la necesidad de seguridad es un factor muy importante en todos los ámbitos de las tecnologías de la información. Desde que Internet fue aceptado como el medio de interconexión global, la mayoría de los negocios se realizan por este medio, por lo cual nace la necesidad de que sean protegidos, ya que cualquier

dispositivo conectado a Internet corre el riesgo de ser atacado desde computadores, servidores, PDA's, routers, hasta teléfonos celulares.

La necesidad de dar una respuesta rápida a incidentes de seguridad, es uno de los factores importantes para cualquier organización para evitar pérdidas irreversibles. En base a lo anterior se han creado grupos denominados CERT (Equipo de Respuesta a Emergencias Informáticas, pero cuyas siglas vienen del termino en ingles), distribuidos a nivel mundial. A finales de los años 80 la Universidad de Carnegie Mellon creó el primer CERT, un grupo formado en su mayor parte por voluntarios cualificados de la comunidad informática, cuyo objetivo principal era facilitar una respuesta rápida a los problemas de seguridad que afectaban a Internet [1, 2, 4].

La presente investigación tiene como propósito implementar los servicios básicos de manejo de incidentes para un CERT. Para ello se desarrollara una aplicación que mantenga una base de datos centralizada de información de incidentes de seguridad informática. En la aplicación estarán integrados agentes que utilizarán un método de búsqueda y selección basado en algoritmos de inteligencia artificial colectiva que tratan de emular el comportamiento observado en las colonias de hormigas [6, 14]. Dichos agentes realizan búsquedas a través de Internet, de incidentes que hayan ocurrido en otros sitios, para conocer las formas de respuestas a estos. Para el desarrollo del modelo distribuido de búsqueda basado en agentes, usaremos conceptos derivados del trabajo [14].

1. MARCO TEÓRICO

1.1 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 la operatividad y buen funcionamiento de cualquier organización, y son considerados como el acto de violar implícita o explícitamente las políticas de seguridad. Existe una gran variedad en el comportamiento de estas amenazas, entre las cuales podemos citar [1]:

- Intentos (exitosos o fallidos) de ganar acceso sin autorización a un sistema o sus datos.
- Interrupciones no deseadas o denegación de servicio.
- Uso desautorizado de un sistema para procesar o almacenar datos.
- Cambiar las características de hardware, firmware, software del sistema o instalar 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 [1], 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, ataques de 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 [4].

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, al igual que prevenir sobre posibles ataques. Dichos grupos son denominados CERT (siglas en ingles para: Computer Emergency Response Team) [2]. Estos equipos son conocidos también por otros nombres: Computer Incident Response Team (CIRT), Computer Security Incident Response Team (CSIRT), System Security Incident Response Team (SSIRT), entre otros. Por lo general, estos equipos están coordinados y en constante colaboración con equipos de diferentes países, expertos de seguridad e instituciones legales [1].

Killcrece, Kosakowsky y otros (2003), en [2], plantean los pasos a seguir para el establecimiento de un equipo de respuesta a emergencias informáticas. Entre otras cosas, define 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.
- 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.

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 [3]. Para esta problemática se propone el desarrollo de una aplicación que realice los servicios de manejo de incidentes para un proyecto CERT. Esta aplicación deberá ser capaz de aportar la mejor solución para tratar un incidente en base a soluciones encontradas en la web por unas herramientas denominadas “agentes de búsqueda”, que formarán parte de la aplicación y facilitarán el proceso de detección, reporte y respuesta de incidentes. Estos agentes estarán localizando constantemente información sobre incidentes en la red para estructurarla, organizarla y almacenarla en una base de datos. Todo esto, con la finalidad de cooperar y hacer comunidad con los diferentes CERT's distribuidos en el mundo.

1.2 SISTEMAS ARTIFICIALES DE HORMIGAS

Existen varios trabajos que estudian un tipo de conducta en una variedad de animales bastante interesante, llamada “conducta colectiva” [6, 13]. Ejemplos de esa conducta son: una bandada de pájaros recorriendo el cielo, un grupo de hormigas en busca de comida, etc. Recientemente, investigaciones han estudiado esa conducta, particularmente cómo estos tipos de animales actúan recíprocamente, logran metas colectivas, evolucionan, etc. La inteligencia colectiva (IC) ha sido aplicada en distintas áreas como telecomunicaciones, robótica, transporte, aplicaciones militares, 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 [6, 13].

En 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 al cargar objetos grandes; Conseguir la ruta mas corta del nido a Fuentes de alimentos; Regulación de temperatura del nido; 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 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”. Aquí 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 [6, 13]. La IC ha inspirado técnicas como optimización colectiva de partículas, sistemas artificiales de hormigas, modelos ecológicos, entre otros [6].

2. PROPUESTA

Se propone un algoritmo de inteligencia artificial colectiva que será aplicado para la búsqueda de incidentes. Además, proponemos una forma de estandarizar la información de incidentes de seguridad en archivos XML [12].

2.1 Formato de archivos XML usado

El sistema hará uso de archivos que caracterizan a los incidentes, y en el presente trabajo dichos archivos tendrán un formato XML. Estos archivos contendrán tanto información estática como dinámica que describen a los incidentes. Entre la información estática se incluyen datos como: nombre, descripción, tipo, fecha de descubrimiento, entre otros, en la que se detalla una breve reseña del incidente, y cualquier otra información que sea permanente en el tiempo. 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 variable que está asociada al nivel de versatilidad del incidente, llamada "feromona".

Las etiquetas XML usadas fueron seleccionadas debido a que son las que mejor se adaptan a las características propias de los incidentes de seguridad, dando así mayor consistencia al momento de tomar la información referente a cada uno de ellos desde un archivo en formato XML (Ver Tabla 1).

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, incident_type, date_discovered, date_updated y affected_platform.
<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

<pheromone>	selección desarrollados en base a inteligencia artificial colectiva.

2.2 ALGORITMO DE BUSQUEDA Y SELECCIÓN BASADO EN SISTEMA ARTIFICALES DE HORMIGAS

El algoritmo de selección está inspirado en el uso de "trazas de feromona", que permiten identificar a los incidentes mejor adaptados a los requerimientos a través de un proceso de retroalimentación positiva o negativa [6, 13].

2.2.1 Modelo de Selección

El sistema utilizará tantos agentes como desee, y cada uno deberá seleccionar incidentes del repositorio. Las etapas que componen el proceso de selección son las siguientes:

1.Cada agente realiza un recorrido, independiente al seguido por el resto de los agentes, por los repositorios buscando incidentes de acuerdo a las características solicitadas

1.1.La trayectoria seguida en cada recorrido consistirá en visitar, aleatoriamente, 1, 2 ó N repositorios, que son los que contienen los incidentes y sus posibles soluciones.

1.2.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.

2.Finalmente, cada agente realiza la selección del incidente encontrado que mejor se adapte del 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 seleccionados. La decisión sobre realizar la inserción de dichos incidentes a la base de datos estará a cargo del usuario de la plataforma que utilice la librería de manejo de incidentes. Dicho usuario seleccionará, según algún criterio, los incidentes que requiere del grupo de incidentes seleccionado por cada agente, y los incorporará a la base de datos.

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 los requerimientos iniciales son perfectos y se tiene información exacta de los incidentes que se desean seleccionar. Esta información esta asociada a archivos XML, los cuales agrupan la caracterización de los incidentes.

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

$$X_{lj}^{i} = 1 + \sum H_i \; - \sum N_{ljz}$$

donde X_{lj}^{i} es el grado de correspondencia entre el incidente ideal i, y el incidente j, ubicado en el repositorio l. $\sum H_i$ 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, $\sum N_{lj}$ 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 X_{lj}^{i}, 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 requerido, considerando:

1.El valor ideal de X_{lj}^{i} es 1, el agente escogerá incidentes cuya correspondencia entre el incidente deseado y el encontrado es cercana a ese valor.

2.El monto de feromona $Y_{lj}(t)$ relacionado a cada incidente j, ubicado en el repositorio l, que conforman cada uno de esos conjuntos, será actualizado luego de que el usuario escoja los incidentes a ingresar en la base de datos. El monto se incrementará o decrecerá, dependiendo del resultado de la evaluación del incidente, y de la tasa de evaporación de feromona.

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]:

$$P_{lj}^{ik}\ t \;=\; Y_{lj}(t)\ X_{lj}^{i} \;/ \sum_{rsn\ cc_{ik}} Y_{rsn}(t)\ X_{rsn}^{i} \qquad (1)$$

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 que cada agente ha encontrado.

Como se dijo anteriormente, cada agente k deposita una cantidad de feromona $\ddot{A}Y_{lj}\ t$ en cada uno de los incidentes conseguidos, producto inverso de la correspondencia $X_{ij}k$.

$$\ddot{A}Y_{lj}\ t\ = (X^{k}_{\mathrm{lj}})^{1}$$

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 por el usuario o no. En general, la retroalimentación positiva y negativa es realizada a través de la actualización de la traza como sigue [6, 13]:

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

El monto inicial de feromona de los incidentes se asume como un número aleatorio. a, 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:

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

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

2.2.2 Macro Algoritmo

1.Definición e identificación del perfil deseado de los incidentes. Crear k agentes de selección.
2.Cada agente realiza la selección de los incidentes requeridos usando el procedimiento "selección".
3.Actualizar la traza de feromona para cada incidente .
4.Seleccionar un incidente de entre la selección hecha por los distintos agentes, e ingresarlo a la base de datos de incidentes.

Procedimiento de "selección":

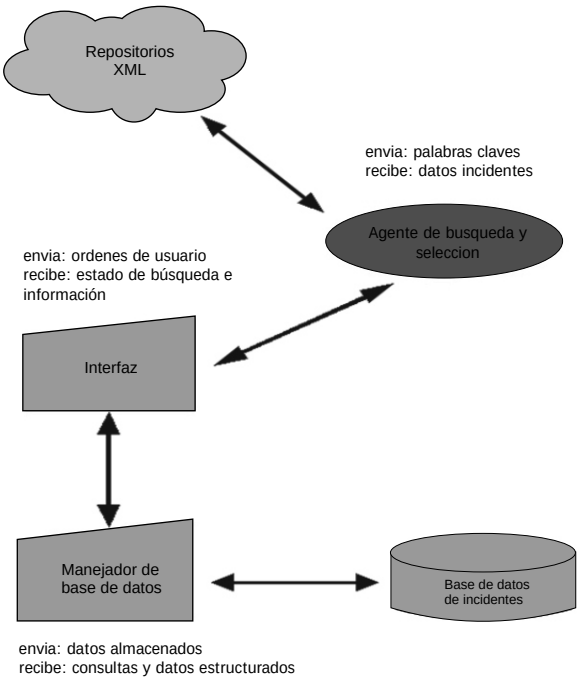
1.Búsqueda de incidentes similares al incidente i requerido (estos incidentes conforman al conjunto
2.Seleccionar uno de ellos usando la fórmula 2.

2.3 DESCRIPCIÓN GENERAL DEL SISTEMA

El sistema realizará la búsqueda de información de incidentes y soluciones de seguridad informática utilizando un algoritmo de búsqueda y selección basado en inteligencia artificial colectiva. El sistema debe servir de apoyo a un CERT en el proceso de manejo y respuesta a incidentes gracias al aporte de información actualizada. Además de esto, se pretende integrar en el sistema una base de datos que almacene la información recopilada de los repositorios de incidentes y de esta manera mantener un registro de los incidentes ocurridos en las instituciones registradas en el CERT. En líneas generales, el sistema permitirá:

- Realizar la búsqueda de información de incidentes de seguridad informática en archivos XML ubicados en repositorios web usando agentes de búsqueda y selección.
- Agilizar el proceso de respuesta a incidentes.
- Manejar una base de datos de registro de información de incidentes.

Figura 1. Componentes del manejador de incidentes



Sus componentes son (ver figura 1):

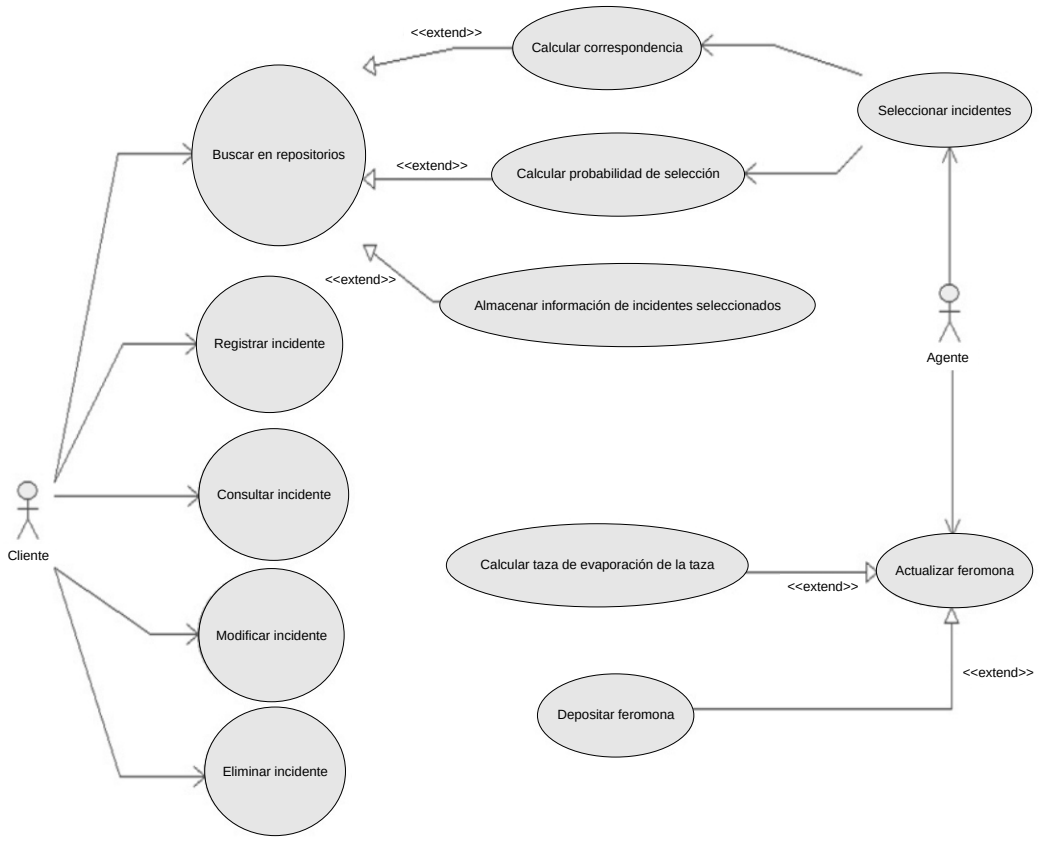
- Interfaz: se encargará de recibir las peticiones del cliente (programa o sistema) y realizarlas de manera transparente a este. La interfaz realizará la administración de los agentes y consultas a la base de datos.
- Agentes de búsqueda: Serán los encargados de buscar los incidentes que cumplan con las especificaciones dadas por el usuario en los

archivos XML del repositorio (local a efecto de pruebas). De la misma forma, tendrá la tarea seleccionar los incidentes que mas se acercan a los requerimientos del usuario, realizando un filtrado de estos al utilizar los principios del algoritmo de inteligencia artificial colectiva explicado anteriormente.

- Manejador de base de datos: Se ocupará de realizar los intercambios de datos entre el sistema y el cliente. Manejará una base de datos diseñada para almacenar datos referentes a incidentes. Con esta base de datos se pretende centralizar información actualizada de incidentes para ayudar a los integrantes del CERT en su labor de respuesta a incidentes.
- Repositorio XML de incidentes: Son los repositorios de incidentes basados en archivos XML para la descripción de los incidentes.

Y el diagrama de casos de uso basado en UML es:

Figura 2. Diagrama de casos de uso del Sistema



En particular, si se elije Buscar Incidentes en los Repositorios, el cliente deberá ingresar los campos de la manera correcta. Una vez realizado esto, el sistema se encarga de activar el proceso de búsqueda basado en el algoritmo mostrado en la sección 2.2.

2.4 PANTALLAS DEL SISTEMA

Se presentarán a continuación algunas pantallas de interfaz gráfica del sistema. La figura 3 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 (figura 4), insumo necesario para que los agentes de búsqueda puedan realizar ese proceso. La figura 5 es la pantalla para introducir nuevos incidentes en la base de datos de incidentes.

Figura 3. Pantalla inicial

Figura 4. Formulario para realizar la búsqueda de incidentes

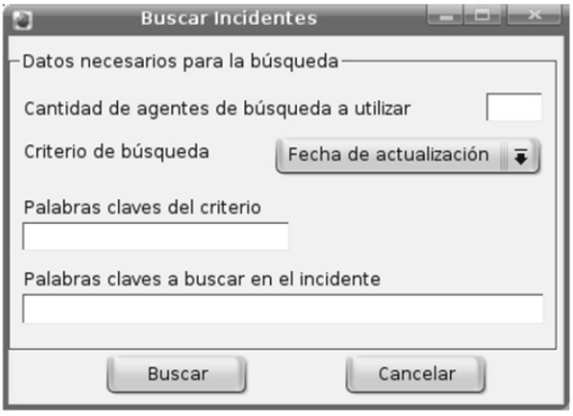


Figura 5. Formulario para registrar incidentes en la Base de Datos

3. EXPERIMENTOS

3.1 DESCRIPCIÓN DE LOS EXPERIMENTOS

Las pruebas se realizan con tres (3) repositorios que almacenan información de incidentes, recopiladas de fuentes de internet como panda [14], symantec [15] y mcafee [16]. 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.

3.1.1. 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 de 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 del 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

Incidente a buscar #3	
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.

3.1.2 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

3.2 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 [17]. Aquí solo analizaremos los de los dos experimentos antes explicados.

3.2.1 Experimento 1

En la tabla 4 se muestran los mejores perfiles encontrados de acuerdo al criterio de probabilidad, asociado al incidente requerido, y el repositorio de donde provienen.

Tabla 4. 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 seleccionó 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 2.8841 después de su evaluación. El segundo corresponde al descrito por el archivo “5.xml”, con 0.2 de probabilidad, con un valor de feromona final 1.2906 después de la evaluación dada. 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 después de la respectiva evaluación. 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 luego de la evaluación. Es de gran importancia resaltar el hecho de que el perfil encontrado con mayor probabilidad no es precisamente el incidente requerido, esto debido a que la asignación de probabilidad fue igual para los primeros 2 incidentes debido a la similaridad de sus perfiles.

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 luego de la respectiva evaluación. 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.

3.2.2 Experimento 2

Los resultados obtenidos en el experimento 2 se muestran en la tabla 5.

Tabla 5. Resultados de la búsqueda para el experimento 2

Comida #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
Comida #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
Comida #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
Comida #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/0.712473/12.1563/1 9.xml/0.215285/19.4443/1 7.xml/0.044204/3.86004/1 4.xml/0.028037/3.34125/1	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/0.4334/4.72142 /1 3.xml/0.4235/4.14648 /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 5), 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. 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 5), 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 difiere entre las corridas, debido a que al finalizar un proceso de búsqueda y selección en una corrida del programa, todos los incidentes evaluados relacionados a los incidentes requeridos son sometidos al proceso de evaporación de su traza de feromona, y, los incidentes seleccionados reciben una cierta cantidad de feromona que dependerá de la evaluación (considerada aleatoria para el caso de pruebas). Esta actualización de feromona es lo que interviene en la selección de incidentes en corridas consecutivas del sistema, esto significa, que un incidente que posea un valor de feromona alto se considera que es “bueno” en correspondencia con los requerimientos del usuario, y por lo tanto, su probabilidad de selección es mayor que otro con feromona baja.

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.

4. CONCLUSIONES

Durante este trabajo se llegaron 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 validados, que contengan información detallada de las características de los incidentes.
- Para determinar el número ideal de agentes a usar para la búsqueda y selección, debe ser tomado en cuenta tanto la cantidad de incidentes requeridos como de repositorios a ser considerados. Es decir, el número de agentes a usar debe garantizar conseguir todos los incidentes solicitados, buscándolos a través de todos los repositorios disponibles, esto permitiría obtener mejores resultados.
- 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.
- 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.

5. REFERENCIAS

[1]US-CERT. https://forms.us-cert.gov/report/.
[2]Killcrece G., Kosakowsky K., Ruefle R., Zajicek M. Organizational models for computer security incidents response teams (CSIRT's). Carnegie Mellon Software Engineering Institute. Reporte Técnico No. IA-230, 2003.
[3]Alberts C., Dorofee A. Managing Information Security Risks, Addison Wesley, 2002.
[4]Killcrece G., Kosakowsky K., Ruefle R., Zajicek. State of the practice of Computer Security Incident Response Team (CSIRT's). Carnegie Mellon Software Engineering

Institute. Reporte Técnico No. IA-233, 2003
[5]Estrada M. (2003). Delitos informáticos. http://scholar.google.co.ve/scholar?num=100&hl=es&lr=&q=cache:3cWkjYSX5IEJ:unifr.ch/derechopenal/articulos/pdf/DELITOS.pdf
[6]Abraham B. Aguilar J. y Bastidas J. Selection algorithm using Artificial Ant Colonies, WSEA Transactions on Computers, Vol. 5, No. 10, pp. 2197-2203, 2006.
[7]Jennings S. Wooldridge V. A Roadmap of Agent Research and Development. Autonomous Agents and Multi-Agent Systems Institute, Reporte Técnico No. 321, 1.998.
[8]Nwana R. Software Agents: An Overview. Knowledge Engineering Review, Vol. 13, pp. 18-29. 1.996.
[9]Cortés L., Sánchez M. Agentes en la Red. Novatica, 1.996.
[10]Foundation for Intelligent Physical Agents. http://www.fipa.org/.
[11]Nilsson, Nils J. Inteligencia artificial: Una nueva sintesis". McGraw Hill, 2001.
[12]Wikipedia. XML. http://es.wikipedia.org/wiki/XML
[13]Aguilar J., Rivas F. Introducción a la Computación Emergente, MERITEC, 2001.
[14]Panda Software. http://www.pandasoftware.com/virus_info/encyclopedia/.
[15]Symantec Corporation. http://www.symantec.com/enterprise/security_response/threatexplorer/threats.jsp.
[16]McAfee Inc. http://us.mcafee.com/virusInfo/default.asp?id=calendar.
[17]Latorre E., Moreno G., Aguilar J., Abraham B. Aplicación para el manejo de incidentes del proyecto CERT Venezuela basado en teoría de agentes. Informe Técnico No. 10-2007, CEMISID-ULA.