VISIÓN ELECTRÓNICA *Preparación de Artículos revista VISIÓN ELECTRÓNICA: algo más que un estado sólido*
Fecha de envío: 25/11/2019
Fecha de recepción:
Fecha de aceptación:

ARQUITECTURA PARA ESCANEO DE PUERTOS USANDO AGENTES MÓVILES

ARCHITECTURE FOR PORTS SCAN USING MOBILE AGENTS

July Andrea Rocha Hernández*Roxana Andrea Jaramillo Cobos. Luis Felipe Wanumen Silva. *****

Resumen*:* En este artículo se propone una arquitectura de software para el escaneo de puertos usando agentes móviles, con el fin de facilitar el monitoreo de tráfico en la red o detección de vulnerabilidades de los puertos lógicos del sistema operativo. La arquitectura expuesta permite la ejecución de procesos residentes en cada una de las estaciones a ser monitoreadas; incluye mecanismos de comunicación por lo cual entrega un alto grado de autonomía a los agentes, mecanismos de planeación de rutas, puesto que así se elige la ruta más óptima para cada agente que se asigne a ejecutar una tarea en un host de la red, cabe resaltar que la correcta asignación de tareas en el lugar y momento concreto son la clave para el buen desempeño de un sistema de agentes móviles, teniendo en cuenta que el nivel de detalle de las arquitecturas depende del tamaño del sistema.

*Estudiante de Ingeniería Telemática, Universidad Distrital Francisco José de Caldas, Colombia. Pmo Analyst: TransUnion Colombia. E-mail: JulyAndrea.Rocha@transunion.com. ORCID https://orcid.org/0000-0001-5730-5284
**Estudiante de Ingeniería Telemática, Universidad Distrital Francisco José de Caldas, Colombia. Semi senior Developer: Experian. E-mail: Roxana.Jaramillo@experian.com. ORCID https://orcid.org/0000-0003-1207-0092
***Ingeniero de Sistemas, Universidad Distrital Francisco José de Caldas. Maestría en Ingeniería de Sistemas y Computación, Facultad Tecnológica, Proyecto curricular Sistematización de Datos e Ingeniería Telemática, Colombia. Docente: Universidad Distrital Francisco José de Caldas, E-mail: lwanumen@udistrital.edu.co. ORCID https://orcid.org/0000-0002-8877-5681

Este tipo de soluciones daría alta disponibilidad de la información en ambientes distribuidos y disminución del tráfico de red, de esta se apoya en las ventajas de los agentes inteligentes, permitiendo así informar a otros agentes del sistema la probabilidad de ataque en un puerto determinado. La arquitectura es validada por medio del prototipo “Scan Mobile UD” desarrollado en Java haciendo uso de Aglets, realizando pruebas sobre un entorno real, para detectar la probabilidad de vulnerabilidad de seguridad lógica en la arquitectura propuesta.

Palabras clave: Escaneo de puertos, agentes móviles, Arquitectura de software, Aglets.

Abstract: This article a software architecture approach for scanning ports is done using mobile agents, aiming network traffic monitoring detecting vulnerabilities in operating systems’ logic ports. The exposed architecture allows local executions in every computing device; it aggregates communication mechanisms to the agents, routes planning, choosing best route for each assigned agent in a network host, given correct task assignment in best moment and scenario achieving best mobile agent system possible takin into account that the amount of architectural details depends on each system size. This solution gives high availability of distributed environment and diminished network traffic by using intelligent agents allowing a communication to other agents informing a probability of a security attack to a certain port. The architecture is validated through Scan Mobile UD prototype, being developed in Java language technology implementing Aglets, and tested in a developed network, proving agents’ mobile effectivity to detect logical network vulnerabilities risks in proposed architecture.

Key Words: Ports scanning, mobile agents, Software Architecture, Aglets.

Fecha de envío: 25/11/2019
Fecha de recepción:
Fecha de aceptación:

1. Introducción

Las medidas de seguridad en las redes organizacionales es un tema que debe estar en constante revisión con el fin de hacer frente a los nuevos ataques; Es necesario identificar, prevenir y protegerse de numerosas vulnerabilidades, principalmente, el acceso a la red por medio de los puertos lógicos, ya que estos son una puerta para el ingreso y salida de datos (incluso de datos no autorizados) que pueden afectar los sistemas de seguridad, así como también, los equipos en la red.

Uno de los problemas que aún se evidencia en las actuales arquitecturas implementadas en las organizaciones para el monitoreo de tráfico en una red o detección de vulnerabilidades en la misma a través de los puertos del sistema operativo (puertos lógicos), es que dichas aplicaciones se continúan ejecutando, normalmente en una arquitectura cliente – servidor. Esto ocasiona que sí por medio de un virus o alguna otra técnica, la aplicación central es vulnerada y/o desactivada, la red entera quedaría desprotegida; por otra parte, el escaneo se vuelve tedioso cuando se tienen que escanear una gran cantidad de puertos en distintas máquinas y no se cuenta con una arquitectura descentralizada que garantice que los detalles del escaneo en una máquina, son conocidos por parte de otra estación. Este tipo de soluciones daría alta disponibilidad de la información en ambientes distribuidos, en la cual, si una parte falla, otra pueda tomar el control y mantener su funcionamiento o reportar la falla para su debido análisis y solución.

Planear arquitecturas que permitan ejecutar procesos residentes en cada una de las estaciones a ser monitoreadas y posteriormente transportar esta información en forma colaborativa a través de la red cuando un coordinador lo decida, garantiza bajar el tráfico de red [1]. Obviamente este coordinador debe implementar los mecanismos necesarios que garanticen que siempre habrá disponibilidad. Para mitigar en gran medida los problemas anteriores, se plantea una arquitectura basada en agentes móviles que se apoya en las ventajas que estos ofrecen [2] y las plataformas multiagente [3] para desarrollar soluciones de software que garantizan la disponibilidad de la información en un ambiente distribuido.

2. Marco Conceptual

2.1 Agente móvil

Un agente móvil es un programa que puede moverse a través de una red bajo su propio control, capaz de navegar a través de la red subyacente y realizar varias tareas en cada nodo de forma independiente [4]; Otra definición aportada a la literatura: los agentes móviles son programas que pueden enviarse desde una computadora y entregarse a una computadora remota para su ejecución, al llegar a la computadora remota, presentan sus credenciales y obtienen acceso a servicios y datos locales [5]; Aunque en la literatura existen varios definiciones para agentes móviles las citadas son la más acertada de acuerdo a nuestro enfoque. Estos presentan ciertos atributos como: autonomía, habilidad social, reactividad, pro actividad, movilidad, operación continua y adaptabilidad como se detalla en [6], no siendo en la práctica obligatoria la presencia de todas estas.

Los agentes móviles, tienen la principal habilidad de ejecutarse en varias máquinas sin necesidad de que en estas se localice su código. Como su nombre lo indica su código es móvil y pueden ejecutarse sin conexión trabajando en la red aun si no está funcionando,

Fecha de envío: 25/11/2019
Fecha de recepción:
Fecha de aceptación:

esperando a que se reanude; las principales ventajas, identificadas en el uso de agentes móviles, radican en: la potencial reducción de costos de comunicación global mediante la migración de las unidades de computación a los datos; y la posibilidad de distribuir computaciones complejas en diferentes hosts, posiblemente heterogéneos, además de capacidades de eficiencia, adaptación al cliente, reducción del tráfico de la red, gestión de gran volumen de información y comunicación en tiempo real [7].

2.2 Sistema de agentes móviles

Los sistemas de agentes móviles se enfocan principalmente en dos componentes: el agente y la plataforma de agentes. Aquí, un agente consta de código y de la información de estado necesaria para realizar alguna tarea. La tarea es típicamente realizada de forma autónoma por el agente, quizás con la cooperación de otros agentes. La movilidad permite al agente migrar o saltar entre distintas plataformas. Un agente móvil puede por tanto ser definido como un invitado que temporalmente visita varias plataformas de agentes, las cuales soportan sus actividades [8].

En cuanto a la plataforma deberá ofrecer, aparte del soporte de creación, cuando menos tres servicios básicos: movilidad, localización y comunicación como se detalla en [9]; Para ello es importante que existan mecanismos que permiten a los agentes organizarse para realizar su trabajo en forma colaborativa. La comunicación entre agentes es indispensable en tales circunstancias. Los agentes podrán formar grupos de trabajo que les permitan trabajar en forma coordinada con un objetivo en común.

2.2 Aglets

Los aglets son objetos Java que pueden moverse de un host en Internet a otro. Es decir, un aglet que se ejecuta en un host puede detener repentinamente la ejecución, enviar a un host remoto y reanudar la ejecución allí. Cuando el aglet se mueve, trae consigo su código de programa y su estado (datos) [10].

El sistema de agente móvil debe ser capaz de proporcionar el entorno de ejecución para agentes móviles. Algunos de los agentes móviles comunes que se han desarrollado en el pasado son: Telescript, Concordial, Ara, Jade, Semoa, Tacoma, Tracy, Ajanta, Aglets, entre otros. Aglet fue desarrollado por el IBM Tokyo Research Laboratory [11]. Por lo tanto un aglet requiere una aplicación Java host, un "host aglet" que se ejecuta en una computadora antes de que pueda visitarla [12].

2.2 Escaneo de puertos

Un escaneo de puertos es una técnica de inspección que intenta descubrir qué servicios están siendo ofrecidos por una red o servidor, consiste en efectuar conexiones o intentos de conexión a diferentes puertos (TCP o UDP) en la víctima esperando obtener respuesta de alguno o algunos de ellos y deducir qué aplicación o servicio está escuchando en determinado puerto [13] [14].La metodología de ataque está compuesta por seis etapas, a saber: reconocimiento, escaneo, enumeración, explotación, mantenimiento del acceso y eliminación de rastro [15] , organizado de forma en que el éxito de una garantiza un mejor escenario para la siguiente etapa, es por ello que es sumamente importante conocer todas las herramientas, métodos y tiempo disponible para realizar cada una de ellas. La segunda etapa, el escaneo, tiene como objetivos primordiales cuatro puntos: encontrar host activos,

Fecha de envío: 25/11/2019
Fecha de recepción:
Fecha de aceptación:

puertos abiertos, versiones de aplicaciones y sistemas operativos, y otras vulnerabilidades de la red que se quiere atacar.

Es por eso que, con el objetivo de enriquecer el proceso de protección de puertos lógicos, se plantea el uso de sistemas de agentes móviles puesto que son sistemas ampliamente distribuidos y heterogéneos que permiten crear, interpretar, ejecutar, transferir y terminar a los agentes, estos sistemas involucran dos conceptos básicos: servidores de recursos que pueden contener uno o más sistemas de agentes y estos a su vez pueden contener uno o más lugares, y los agentes que pueden estar disponibles en uno o más lugares los cuales poseen la capacidad de trasladarse para acceder a sus recursos, o contactarse con otros agentes. La movilidad de los agentes, es decir su capacidad de migrar de un sitio al otro, establece la principal diferencia que este enfoque tiene con respecto a otras alternativas existentes para el desarrollo de sistemas distribuidos, como los de remote procedure call (RPC) o remote evaluation [7].

El presente artículo consta de las siguientes fases: un Introducción a la arquitectura propuesta, marco teórico en la sección 2, metodología, implementación mediante la descripción de cada uno de los mecanismos en la sección 4, verificación de la arquitectura sobre un prototipo en la sección 5, resultados y conclusiones.

3. Metodología

La arquitectura se presenta mediante el modelo 4+1 de Kruchten [16]motivado por escenarios y desarrollado iterativamente; Propone cuatro vistas: lógica, desarrollo, procesos y física; también una vista adicional llamada escenario utilizada para vincular a las demás. La validación de la arquitectura se hace por medio del prototipo Scan Mobile UD siguiendo las

actividades: Visualización de interfaces principales, ejecución sobre un escenario de red real y evidencia de resultados del escaneo de puertos en las estaciones de trabajo con el fin de mostrar probabilidades de ataque.

4. Implementación de la Arquitectura propuesta

Entre las diferentes plataformas disponibles para crear, proporcionar movilidad y comunicación a los agentes móviles podemos encontrar el framework “AGLETS” descrito en la figura 1, el cual permite desarrollar diversos agentes, tiene un espacio de trabajo que corre sobre un servidor de IBM llamado Tahití y utiliza el protocolo ATP para el paso de mensajes entre agentes. Este framework es uno de los más usados para el desarrollo de aplicaciones basadas en agentes móviles.

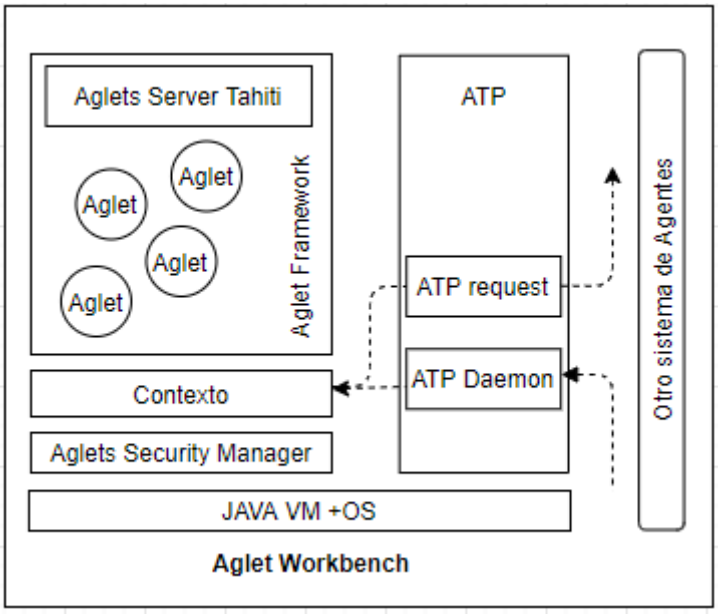

Figura 1.Descripción del framework Aglets. Fuente: Desarrollada por los Autores.

Sobre este framework se implementa la arquitectura de la figura 2 mediante el desarrollo de varios subsistemas, tal como se muestra a continuación:

Fecha de envío: 25/11/2019
Fecha de recepción:
Fecha de aceptación:

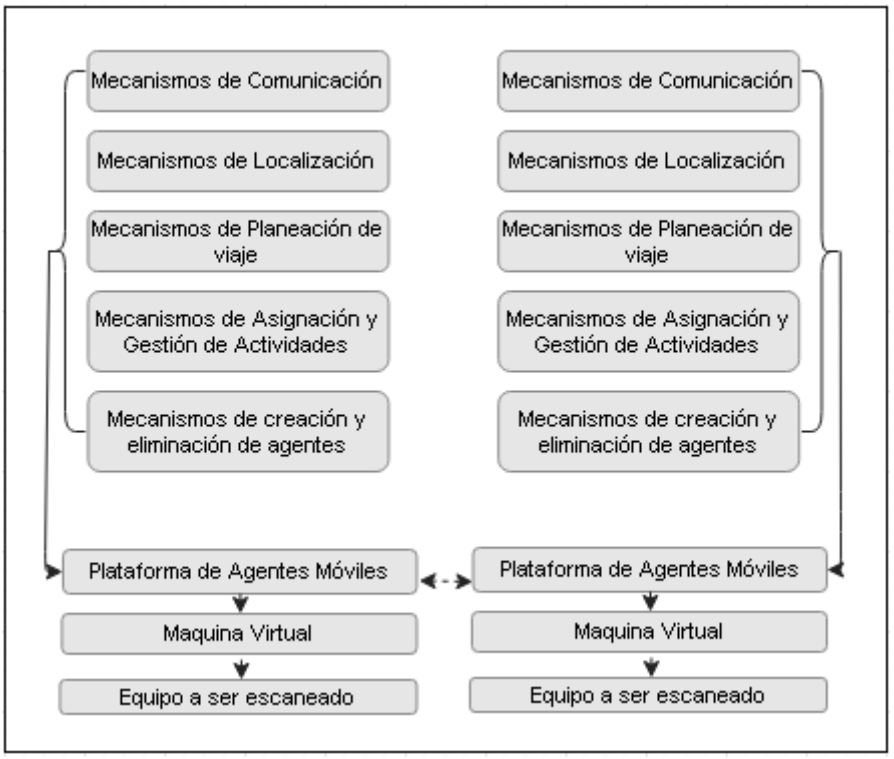

Figura 2.Arquitectura Propuesta. Fuente: Desarrollada por los Autores.

4.1. Subsistema de comunicación

Los mecanismos de comunicación son indispensables en cualquier sistema tanto de agentes móviles como de agentes en un sistema multiagente [17]. Entre otras cosas, el incluir estos mecanismos en la arquitectura entrega al sistema un grado alto de autonomía a los agentes [18].Este subsistema permite informar a otros agentes del sistema la probabilidad de ataque en un puerto determinado, se apoya idea de obtener un contexto de trabajo para el aglet, es decir define el medio con el que se relaciona un agente con otros, en pocas palabras si un agente tiene el mismo contexto que otro agente, los dos agentes se pueden comunicar en el mismo lugar de trabajo [19].

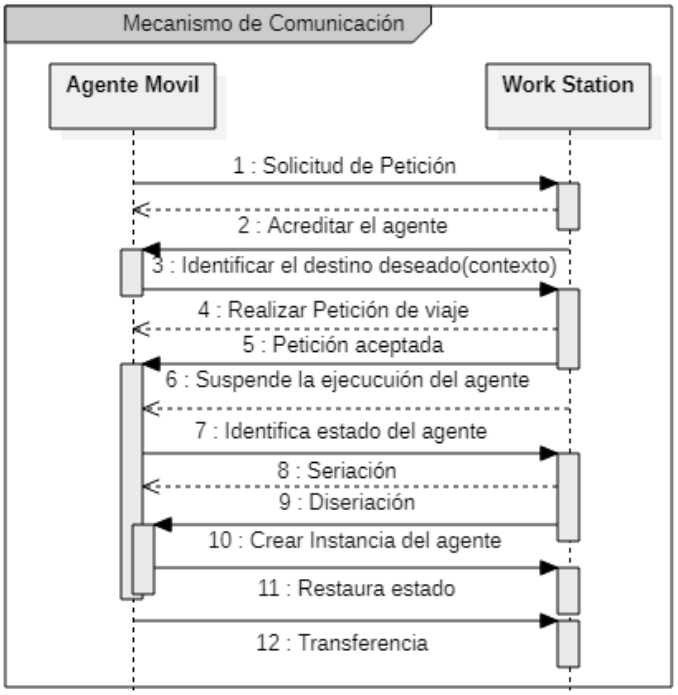

Figura 3.Descripción de transferencia en el enfoque de contexto de aglet. Fuente: elaboración propia de los autores.

Los agentes se transfieren de un sistema a otro, como se muestra en la figura 3. El agente establece el destino deseado, realiza una petición de viaje, si es aceptada recibe autorización para realizar la transferencia, para esto el sistema suspende la ejecución del agente e identifica el estado y las partes que serán enviadas, se realiza la conversión del código y estado del agente (seriación) y se codifica según el protocolo seleccionado, el sistema destinatario acredita al cliente [20]. Se realiza la descodificación del agente y la conversión del código y estado del agente (diseriación), el sistema crea la instancia del agente, restaura su estado, continúa su ejecución y procede con la transferencia.

4.2 Subsistema de localización

Permiten a un agente coordinador de un grupo de escaneadores de puertos encontrar a sus agentes trabajadores y viceversa, el cual define la forma como se puede localizar un agente en una máquina remota, a fin de establecer posteriormente una comunicación con estos agentes. Los mecanismos de localización funcionan bien gracias al desarrollo de estándares

Fecha de envío: 25/11/2019
Fecha de recepción:
Fecha de aceptación:

como FIPA, sin el que los agentes no se podrían comunicar [21], en el caso de las tecnologías actuales para modelar estos mecanismos se aconseja seguir un estándar basado en KQML u otro que cumpla con altos niveles de aceptación por la comunidad científica [22] [23].

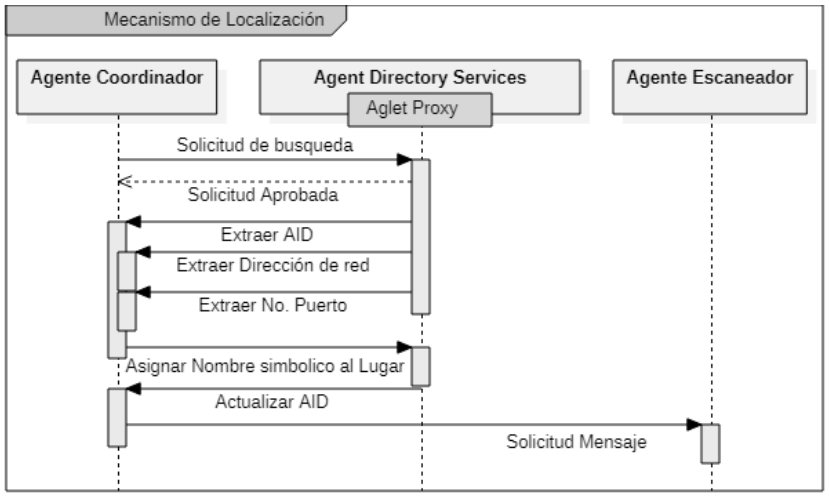

Figura 4.Descripción del enfoque Agent Directory Services, usado en el subsistema de localización. Fuente: elaboración propia de los autores.

En la figura 4 se puede apreciar que el mecanismo de localización implementa la funcionalidad de un directorio de servicios, utilizado por el agente coordinador/Boss para localizar los agentes escaneadores/empleados servicios con los que desea interactuar e invocar, este directorio obtiene datos mínimos como el AID (Agent Identifier), dirección de red, el puerto lógico objetivo, descripción del lugar, a su vez contiene el tipo de transporte (protocolo); el Aglet Proxy (Representante legal) realiza la comunicación [24], finalmente se efectúa la búsqueda de los agentes por matching simple de cadenas y pares atributo-valor.

4.3. Subsistema de planeación del viaje

Permiten a cada agente planear la ruta que debe escoger para llegar a una máquina remota y realizar posteriormente el escaneo en dicha máquina. Internamente estos mecanismos invocan los subsistemas de localización y comunicación con agentes remotos para saber si ya hay agentes que ejerzan los roles planeados en la máquina remota, a fin de tomar la mejor decisión al enviar agentes y no incursionar en el envío de agentes a máquinas remotas que ejecuten roles que ya se estén ejecutando. Es importante anotar que el establecimiento de una arquitectura específica para cada uno de los subsistemas ayudaría notablemente a la mejor construcción de un sistema basado en agentes [25], sin embargo, como se anota en otros documentos de la sociedad de agentes, el nivel de detalle de las arquitecturas depende del tamaño del sistema.

El subsistema de planeación de viaje se basa en los itinerarios de aglets, que incorporan la funcionalidad necesaria para realizar un plan de viaje, en el cual cuando un agente se despacha se le asocia a dicho agente un itinerario para que el agente conozca por donde debe desplazarse para llegar al destino deseado [26]. Para implementar la descripción de la figura 5, se crean itinerarios preestablecidos en la configuración de la topología de la red que se le instala al escaneador. Se desarrolló un archivo XML en el que el usuario puede configurar las posibles rutas que puede seguir un agente para llegar de un equipo a otro, para esto se requiere la ayuda de un experto en redes, porque de hacerse estas rutas con una mala planeación, se tendrían demoras injustificadas al momento de desplazarse un agente de un lugar a otro [27].

Fecha de envío: 25/11/2019
Fecha de recepción:
Fecha de aceptación:

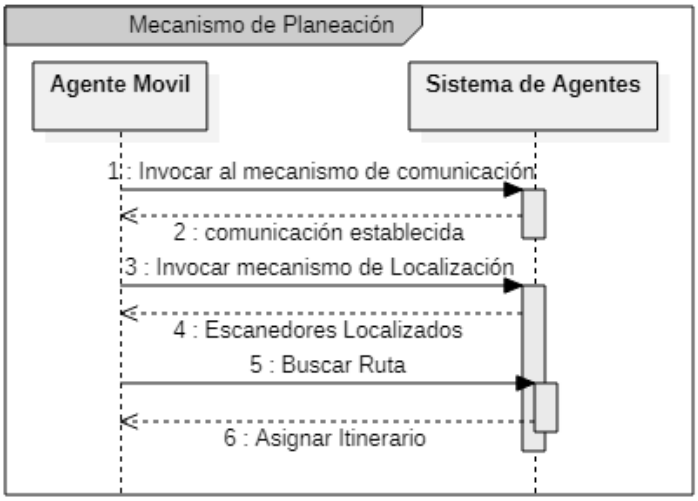

Figura 5.Descripción del enfoque de itinerario provisto por Aglet usado en el subsistema de planeación de viajes. Fuente: elaboración propia de los autores.

4.4. Subsistema de asignación y gestión de actividades

Permiten a un agente móvil una vez ya ha llegado al destino, comunicarse con los otros agentes móviles que ya están hospedados en dicha máquina y tomar un rol que le permita ser útil en el proceso de escaneado de puertos en la máquina establecida. En este punto es posible que los agentes tengan que tomar decisiones autónomamente sin la supervisión del coordinador general remoto. Los mecanismos de asignación y gestión de actividades son la clave para el buen desempeño de un sistema móvil basado en agentes [28]. Incluso hay quienes opinan que la correcta asignación de tareas, en el lugar y momento concreto es la clave para el buen desempeño de un sistema de agentes móviles.

Para ello se hace uso del AID, con el propósito de asignar actividades a un agente en específico, el agent directory service localiza y verifica el AID asociado, el cual es

sincronizado con el AgentBoss o Coordinador que ordena las actividades a realizar por el agente escaneador, tal como muestra la figura 6.

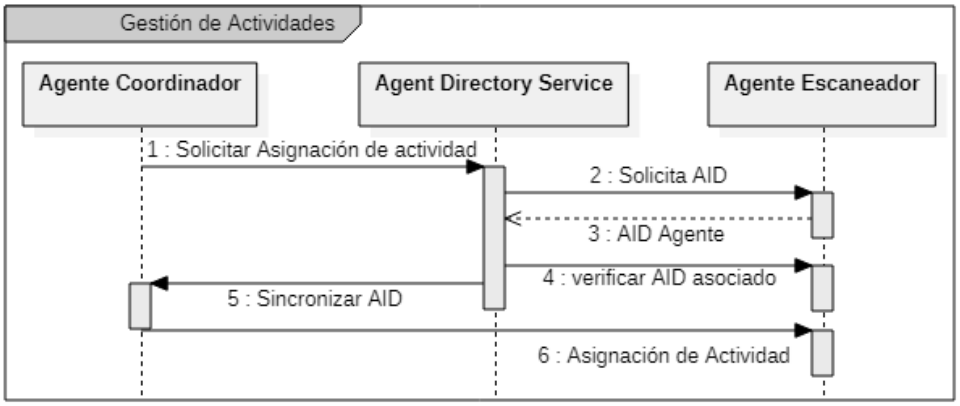

Figura 6.Descripción del enfoque del identificador provisto por Aglets, usado en el subsistema de asignación y gestión de actividades. Fuente: elaboración propia de los autores.

4.5Subsistema de creación y eliminación de agentes

En algunas ocasiones dependiendo las situaciones y la complejidad del escaneo del puerto se requiere la intervención de nuevos agentes que colaboren en el escaneo. De otra parte, crear muchos agentes que hagan la misma tarea sobre recarga el sistema y se presentan problemas de redundancia que finalmente causan otra serie de problemas en la red. En estas situaciones se tienen dos casos, un caso en el que el coordinador local detecta que debe enviar nuevos agentes para que ayuden a escanear los puertos remotos y otro en el que el agente que está escaneando el puerto es capaz de invocar la creación de nuevos agentes para que le colaboren con dicho escaneo; en ambos casos a nivel de aglets, se hace uso del “AgletContext” que proporciona el entorno de inicialización y ejecución además de los métodos, que un aglet puede invocar en su contexto actual permite crear, recuperar y eliminar aglets, entre otros, tal como muestra la figura 7. En ella se puede apreciar que la única forma de crear una instancia adecuada de un aglet es dentro de un contexto, se obtiene acceso a este mediante su método getAgletContext, se procede a la creación del

Fecha de envío: 25/11/2019
Fecha de recepción:
Fecha de aceptación:

aglet, se inserta en el contexto actual y se inicia invocación de onCreation () seguido de run (). El método devuelve un identificador (AgletProxy) para el nuevo aglet tan pronto como finaliza el método constructor del aglet.

Los mecanismos de creación y eliminación de agentes comúnmente se han desarrollado a bajo nivel con tecnologías como CORBA y RMI [29], sin embargo, pueden desarrollarse en un futuro nuevos mecanismos a bajo nivel que soporten la creación y eliminación de entidades de software remotamente en ambientes móviles.

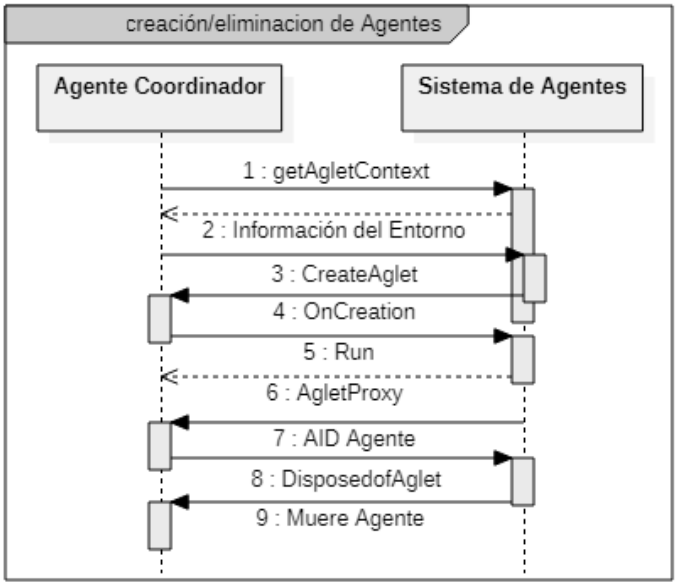

Figura 7.Uso de las funcionalidades del agletContext usadas en el subsistema de creación y eliminación de agentes. Fuente: elaboración propia de los autores.

Es importante mencionar que la clase AgletProxy es la que permite comunicar al AGLET con los representantes de los otros AGLETS, también permite despachar un agente hacia otro contexto, clonar un agente, eliminarlo y permitir que se devuelva al sitio donde estaba antes de despacharlo, todas estas capacidades de la clase AgletProxy se pueden ver resumidas en la siguiente figura 8.

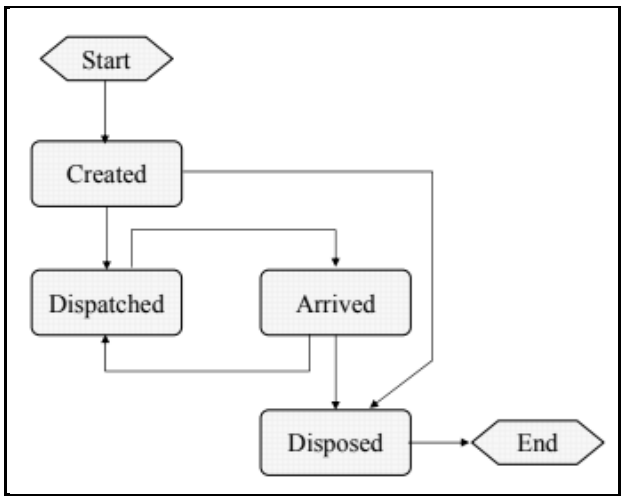

Figura 8.Capacidades de la clase AgletProxy usadas en el subsistema de creación y eliminación de agentes. Fuente: elaboración propia de los autores.

No se puede olvidar que, para lograr la implementación de las capacidades expuestas, la plataforma AGLETS usa las funcionalidades de RMI [29]para permitir la comunicación entre los agentes, usa MASIF para la gestión de agentes e implementa el protocolo ATP (Agent Transfer Protocol) para comunicar entre los diversos agentes, a modo de resumen en la figura 9 se expone el diagrama de componentes de la arquitectura propuesta [30].

Fecha de envío: 25/11/2019
Fecha de recepción:
Fecha de aceptación:

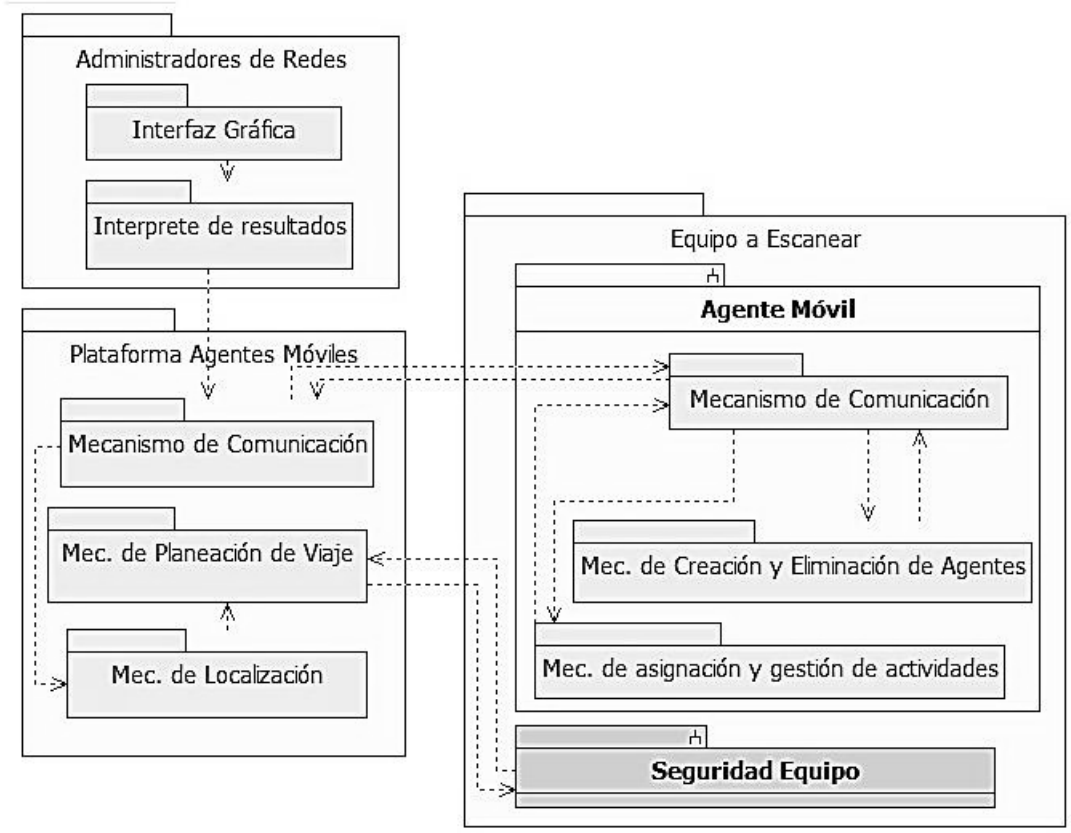

Figura 9.Diagrama de componentes de la arquitectura propuesta. Fuente: elaboración propia de los autores.

5. Escenario de Prueba

La validación de la arquitectura se realizó a través de la aplicación del prototipo SCAN UD de agentes móviles que soporta la funcionalidad de escaneo de puertos lógicos. Para el caso práctico, se definió como escenario de ejecución una red local compuesta por diez equipos de cómputo; en calidad de plataforma de agentes móviles un servidor (Tahití); un administrador de red donde se encuentra el intérprete de resultados junto con la interfaz gráfica de usuario; y los restantes actuando como nodos de la red. En este contexto el objetivo principal es realizar un escaneo de puertos en cada host perteneciente a la red con

la finalidad de evidenciar puertas lógicas abiertas que son vinculadas a vulnerabilidades del sistema, esto soportado bajo la arquitectura propuesta.

Se definió una pantalla en la interfaz de usuario donde se evidencian datos como: IP de red, puerto, protocolo, servicio ejecutándose actualmente por el puerto y estado (abierto o cerrado) del puerto; Los datos obtenidos por la herramienta permiten hacer una exploración de vulnerabilidades de la red y calcular probabilidades de ataque.

5.1Prototipo implementado: Scan Mobile UD

Para la implementación del prototipo se tuvo en cuenta el diagrama de despliegue de la figura 10 y la relación entre cada uno de los mecanismos explicada en la tabla 1.

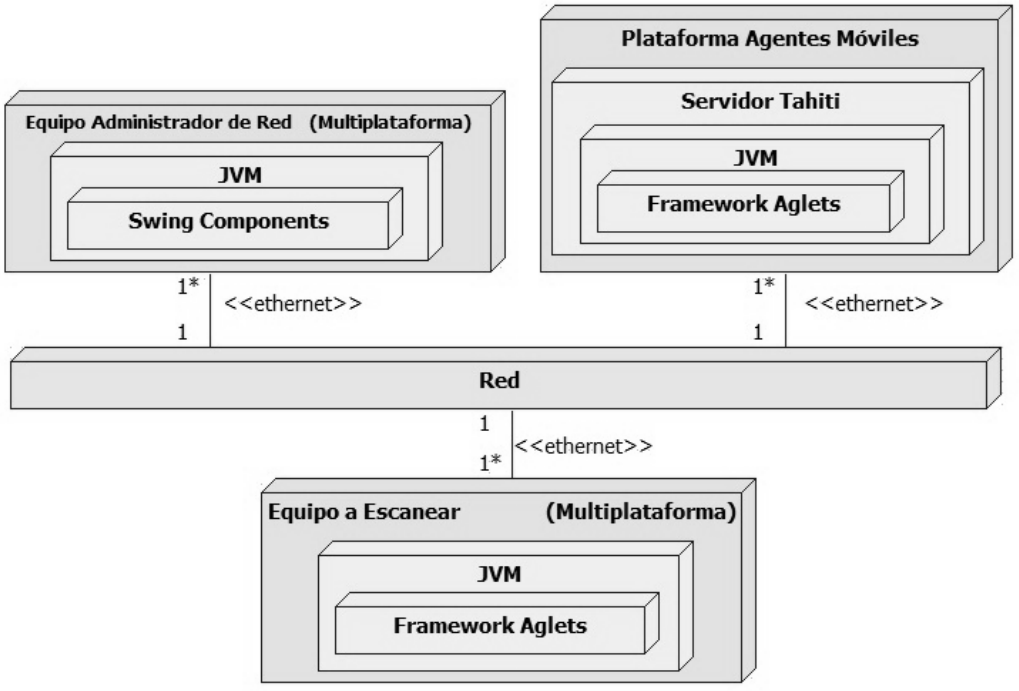

Figura 10.Diagrama de despliegue de la arquitectura propuesta. Fuente: elaboración propia de los autores.

Mecanismos implementados	Funcionalidades dentro del prototipo "Scan Mobile UD" donde se aplica el mecanismo
Mecanismos de creación y eliminación de agentes	Se aplica cuando se define el rango de puertos que el usuario quiere escanear en la figura 11 o cuando quiere hacer un escaneo general.
Mecanismos de comunicación	Cuando solicita iniciar un escaneo, se activan estos mecanismos.
Mecanismos de	Se activan cuando los agentes remotos quieren comunicarse con sus agentes

localización	coordinadores o viceversa. Estos mecanismos son transparentes para el usuario final, pero también se usan cuando en la interfaz gráfica de la figura 12, se totalizan los resultados para ser mostrados en el JTable.
Mecanismos de planeación de viaje	Son mecanismos transparentes al usuario final, pero se activan cuando los agentes planean su traslado a las máquinas locales en donde se va a realizar el escaneo específico.
Mecanismos de asignación y gestión de actividades	Son mecanismos transparentes al usuario final, pero se activan en las máquinas locales
Mecanismos de creación y eliminación de agentes	Son mecanismos transparentes al usuario final que se activan tanto en las máquinas locales, como en las máquinas que coordinan el proceso de escaneo.

Tabla 1: Relación entre los mecanismos de arquitectura con la interfaz gráfica de usuario de "Scan Mobile UD".

La figura 11, muestra la interfaz gráfica que el usuario en el equipo como rol de administrador de red percibe cuando desea comenzar un escaneo, para ello se puede observar que se requiere una serie de parámetros antes de iniciar el proceso. Para el escenario de prueba se indica al sistema que se ejecute sobre todas las IP's de la red junto con todos los puertos que se puedan escanear.

Figura 11.Interfaz gráfica de "Scan Mobile UD" para el inicio del escaneo. Fuente: elaboración propia de los autores.

6. Resultados

La figura 12, muestra la captura de resultados con los parámetros de la figura 11 en donde se puede apreciar que los agentes móviles reportan: puerto, IP, aplicación o servicio ejecutándose, tipo de protocolo y el estado del puerto; A su vez se indica la cantidad de agentes empleados en cada equipo escaneado.

Con los resultados obtenidos se calcula la probabilidad de ataque a nivel de puerto y a nivel general del sistema, aplicando las formulas (1) y (2) respectivamente.

$$\%Probabilidad\ de\ ataque\ por\ puerto = \frac{\sum puertos\ de\ tipo\ X\ en\ estado\ abierto\ *\ 100}{\sum puertos\ de\ tipo\ X} \quad (1)$$

$$\%Probabilidad\ de\ ataque\ al\ sistema = \frac{\sum puertos\ en\ estado\ abierto\ *\ 100}{\sum puertos} \quad (2)$$

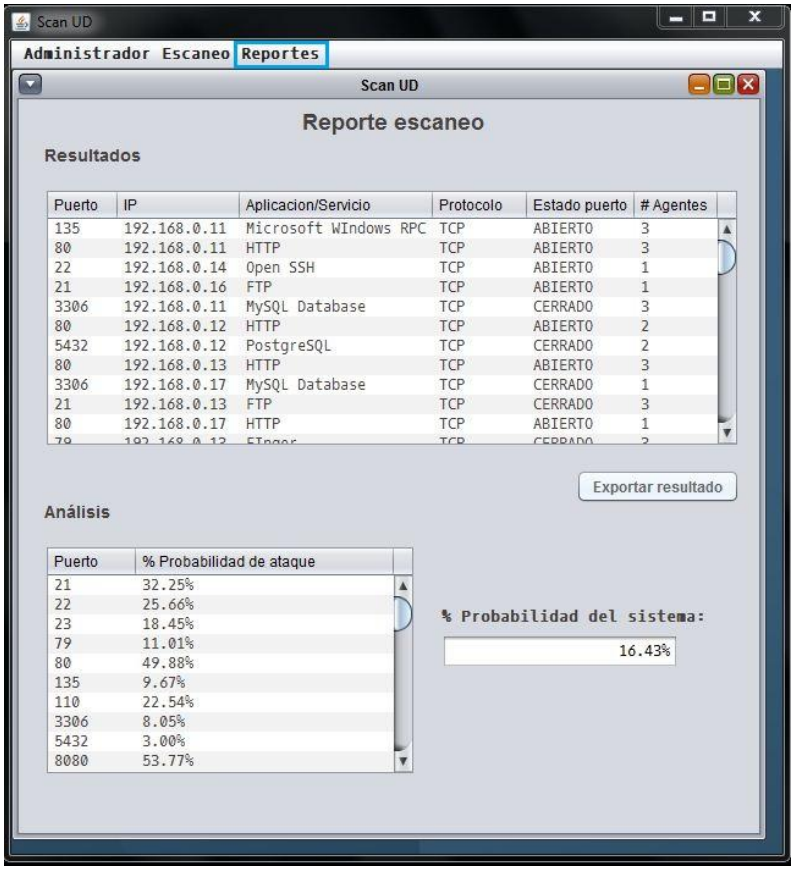

Figura 12.Interfaz gráfica de captura de resultados de "Scan Mobile UD". Fuente: elaboración propia de los autores.

Fecha de envío: 25/11/2019
Fecha de recepción:
Fecha de aceptación:

7. Conclusiones

La arquitectura propuesta basada en agentes móviles para el escaneo de puertos fue validada por medio de un prototipo desarrollado en Java con el uso del framework Aglets; En la ejecución del prototipo en el escenario planteado se comprobó la funcionalidad de los agentes cumpliendo con lo indicado en la arquitectura, además de proveer de una herramienta para la eficiencia en el escaneo de puertos en los equipos de una red, haciendo más fácil la detección y monitoreo de la red frente a vulnerabilidades.

Es importante señalar que todos estos resultados pueden ser afectados por el algoritmo que los agentes móviles usen para escanear puertos. No solamente la arquitectura es la clave para escanear puertos, sino también la forma como se implementen los mecanismos de comunicación entre agentes. De otra parte, es importante anotar que dado que las diversos frameworks de agentes móviles tienen sus cuestiones propias que las diferencian con las otras, es posible que el simple hecho de decidir implementar el modelo en otra arquitectura afecte positiva o negativamente los resultados mostrados en el presente artículo

Se requiere continuar trabajando en la generación de algoritmos para escaneo de puertos a fin de poder hacer una depuración más completa de la arquitectura propuesta con diversos algoritmos de escaneo.

Reconocimientos

Un trabajo como este, es siempre fruto de ideas, proyectos y esfuerzos previos que corresponden a otras personas. En este caso agradecemos al Grupo de Investigación en Informática Organizacional METIS de la Universidad Distrital Francisco José de Caldas

Facultad Tecnológica por su apoyo y asesoría en la implementación de la arquitectura y del prototipo “Scan UD”.

Referencias

[1] A. H. Mohamed and K. Marzouk, "Mobile Agents for Wireless Network Security," 2016.

[2] N. R. Jennings, K. Sycara and M. Wooldridge, "A Roadmap of Agent Research and Development," *Autonomous Agents and Multi-Agent Systems*, vol. I, 1998.

[3] D. Samet, F. Barika Ktata and K. Ghedira, "Securing Mobile Agents, Stationary Agents and Places in Mobile Agents Systems," in *Agents and Multi-Agent Systems: Technologies and Applications*, 2018.

[4] "Mobile Agents for Telecommunication Aplications," Paris, Springer, 2000.

[5] D. B. Lange, M. Oshima, G. Karjoth and K. Kosaka, "Aglets: programación de agentes móviles en Java," vol. 1274, 2005.

[6] L. Camarinha-Matos and W. Vieira, "Intelligent mobile agents," 1998.

[7] E. Belloni and M. Campo, "Una Arquitectura de Software Para Soportar Agentes Móviles Inteligentes," in *II Workshop de Investigadores en Ciencias de la Computación*, 2000.

[8] J. M. Garcia and O. Domingo Jabonero, "Seguridad en Sistemas de Agentes Móviles," 1999.

[9] O. L. Roa, "Agentes de software:tecnologías, herramientas y aplicaciones," vol. 3, no. 1, 2004.

[10] D. B. Lange, "Documento técnico de la interfaz de programación de aplicaciones Java Aglet (J-AAPI)," IBM Tokyo Research Laboratory, 1996.

[11] A. Mishra, R. S. Chowhan and A. Mathur, "Sniffer detection and load balancing using aglets in a cluster of heterogeneous distributed system environment," *2016 IEEE 7th Power India International Conference (PIICON)*, pp. 1-6, 2016.

[12] O. O., Oyewole, A. N.A, O. A and A. M.I, "Design and Implementation of a modified pull-all migration strategy in aglets mobile agent for effective network-load management," vol. 8, 2014.

[13] A. Siler , A. Arboleda and C. Bedón, "Utilizando Inteligencia Artificial para la detección de Escaneo de Puertos," in *VI Jornada Nacional de Seguridad Informática ACIS 2006*, Colombia, 2006.

[14] M. Anbar, A. Manasrah, S. Ramadass, A. Altaher, A. Aljmmal and A. Almomani, "Investigating Study on Network Scanning Techniques," vol. 4, no. 9, 2013.

[15] Chaverrí Pérez, José David; Picado Arias, Kevin, "Combinación Apropiada para Escaneo de Puertos Usando Algoritmos Genéticos," [Online]. Available: https://www.academia.edu/28350009/Combinaci%C3%B3n_Apropiada_para_Escaneo_de_Puertos. [Accessed 20 05 2019].

Fecha de envío: 25/11/2019
Fecha de recepción:
Fecha de aceptación:

[16] P. Kruchten, "Planos Arquitectónicos: El Modelo de "4+1" Vistas de la Arquitectura del Software," [Online]. Available: http://cic.puj.edu.co/wiki/lib/exe/fetch.php?media=materias:modelo4_1.pdf. [Accessed 10 2019].

[17] A. C. Jimenez, J. P. Anzola, G. M. Tarazona and S. J. Bolaños, "Modelo para la simulación de sistemas de multi-agentes robóticos en Python," *redes ing*, pp. 34-41, 2017.

[18] H. A. Diosa, *Propuesta para la conformación de grupo de trabajo en procesamiento basado en objetos distribuidos con CORBA y Objetos de Software Móviles Usando Java*, Bogotá: Universidad Distrital Francisco José de Caldas, 2000.

[19] E. Zapata Granada and C. E. Gómez Montoya, "Arquitecturas de Software para Entornos Móviles," *Revista de Investigaciones Universidad del Quindío*, vol. 1, no. 25, pp. 20-27, 2014.

[20] S. Franklin and A. Graesser, "Is it an Agent or just a Program? A Taxonomy for Autonomous Agents," *University of Memphis*, 1996.

[21] S. Hilde Houmb, "Security Issues in FIPA Agents," paper in progress.

[22] T. D. K. S. Initiative and E. I. W. Group, "Specification of KQML Agent-Communication Language plus example agent policies and architectures," 1993. [Online]. Available: http://www.cs.umbc.edu.

[23] S. Jain, "KQML - From Scenario to Technology," *International Journal of Advanced Studies in Computer Science and Engineering IJASCSE*, vol. 7, no. 3, pp. 30-34, 2018.

[24] J. Xu and S. Wu, "Intrusion detection model of mobile agent based on Aglets," in *International Conference on Computer Application and System Modeling ICCASM*, 2010.

[25] M. Monroy Rios, E. Reyes Torregroza and M. Macareno Mojica, "Arquitectura Basada En Agentes Inteligentes Para La Gestión de Configuración de Red," *Ciencias e Ingeniería al Día*, vol. 7, no. 1, pp. 39-48, 2012.

[26] V. Kumar Manupati, G. Putnik, M. Kumar Tiwari, P. Ávila and M. M. Cruz-Cunha, "Integration of process planning and scheduling using mobile-agent," *Computers & Industrial Engineering*, vol. 94, pp. 63-73, 2016.

[27] F. Al-akashi, "Architecture of a Commercialized Search Engine Using Mobile Agents," *Artificial Intelligence Advances*, vol. 1, no. 1, pp. 44-51, 2019.

[28] R. Pandey, N. Sharma and R. Rathore, *Aglet- A java based mobile agent and its security issues*, 2018.

[29] F. Fernández Moya, "Introducción a la arquitectura corba para sistemas distribuídos," *Grupo de Arquitectura y Redes de Computadores*, 2001.

[30] J. L. Posada Yagüe, Arquitectura de Procesos en sistemas Reactivos Distribuidos, Departamento de informática de sistemas y computadores, 2000.