Sistemas multi agentes para la defensa de redes IoT

Alejandro Ramirez Garcia
Ingeniería de sistemas,
Escuela Colombiana de Ingeniería Julio Garavito
111031
alejandro.ramirez@mail.escuelaing.edu.co

***Resumen*- Este documento presenta diferentes conceptos del área de inteligencia artificial como sistemas multi agentes y machine learning que serán aplicados a la ciberseguridad para la defensa de cualquier tipo de red, pero centrándose en el internet de las cosas.**

I. INTRODUCCIÓN

En la actualidad los sistemas IoT están tomando auge sin embargo el aspecto de seguridad no esta tan avanzado como en otros sistemas lo que los hace más vulnerables a ataques, esto hace que las personas, organizaciones o gobiernos requieran de mayores recursos para defender su información distribuida en equipos. En este documento se presenta una solución para tener mayor resguardo ante ataques en los dispositivos. El documento menciona tecnologías y términos usados como JADE, FIPA, etc. Y como la combinación de las herramientas genera un sistema que puede ser utilizado en cualquier sector del internet de las cosas.

II. INTERNET DE LAS COSAS

Se refiere a la interconexión de objetos de uso diario tales como refrigeradores, termostatos, bombillos, automóviles, etc. cuentan con un identificador único, los dispositivos usan APIs para conectarse a internet para intercambio de datos. La aplicación del internet de las cosas puede darse en casi cualquier campo de acción como el hogar, ciudades inteligentes, en la agricultura, wereables, automatización industrial, infraestructura, transporte, redes eléctricas, seguimiento de inventarios, medicina, etc. Con la capacidad de obtener, analizar y distribuir datos que se convertirán en información, y luego en conocimiento que será aprovechado.

Para que esta tecnología alcance su potencial es necesario tener en cuenta las fuentes de energía autosustentables para cada dispositivo, ya que sería una tarea desgastante tener que cambiar las diversas fuentes de energía cada vez que estas lleguen a su fin.

El protocolo IPv6 que fue diseñado para reemplazar IPv4 debido a que a principios del 2010 la cantidad máxima de direcciones IP asignadas estaban llegando a su límite, por lo tanto, se aumentó el tamaño de las direcciones de 32 bits a 128 bits para aumentar más niveles de jerarquías de direcciones y las subredes tenían como tamaño máximo 2^{64}. Gracias a esta actualización del protocolo hoy en día es una realidad poder conectar la cantidad de objetos o sensores que deseemos.

La capa de tecnologías usadas en el internet de las cosas está dividida en tres: la capa del dispositivo hace referencia al hardware, sensores, actuadores o procesadores, la capa de conectividad para establecer comunicación con la capa de la nube por medio del protocolo machine-to-machine (M2M) MQTT fue diseñado para transportar mensajería de publicar y subscribirse de una manera extremadamente ligera. En la capa de la nube donde se gestionan los dispositivos conectados a la red y se manejan los datos extraídos de los dispositivos para ofrecer servicios útiles.

III. SEGURIDAD EN EL INTERNET DE LAS COSAS

En cuanto a seguridad de la información, el internet de las cosas tiene grandes desafíos debido a las restricciones que tienen los dispositivos en cuanto a poder de procesamiento, memoria y la baja velocidad de transmisión de datos. Las restricciones no permiten portar algoritmos criptográficos de vanguardia ya que requieren más capacidad de computo.

En respuesta a las restricciones que tienen los dispositivos se puede usar criptografía de curvas elípticas (CCE), es una variante de la criptografía asimétrica que cuenta con una llave publica y otra privada con la que se descifra un mensaje, CCE usa matemáticas de curvas elípticas. La ventaja que trae con respecto a los algoritmos de cifrado asimétrico es ser más rápida y usa claves más cortas con un tamaño mínimo recomendado por la NIST de 160 bits para CCE en comparación con 1024 bits para los algoritmos RSA y DSA.

Otro algoritmo usado en el entorno IoT implementando un cifrado simétrico por bloques, se divide el mensaje en bloques del mismo tamaño para aplicar transformaciones a cada bloque. PRESENT cuenta con un nivel de seguridad moderado con llaves de 80 bits y 128 bits, es una implementación de cifrado en respuesta a que no era posible usar AES en entornos con restricciones de recursos de computo. El diseño que tiene hace que sea un algoritmo ultraligero, igualmente pensando en la eficiencia y seguridad. Debido a que tiene un nivel de seguridad moderado, en una evaluación de riesgos no supondría que un ataque tenga un factor de riesgo significativo.

Además que existen algoritmos y protocolos pensados en la seguridad también se pueden implementar placas a los dispositivos IoT con módulos criptográficos como el Arduino MKR1000 que tiene integrado el chip ATEC508A, se caracteriza por brindar el algoritmo Elliptic Curve Digital Signature Algorithm (ECDSA) que es una modificación del algoritmo DSA para firmas digitales. Se diferencian por que DSA emplea exponenciaciones y ECDSA operaciones sobre

puntos de curvas elípticas. Además de encontrar soluciones para firmas digitales como ECC, Atmel una compañía fabricante de semiconductores también ofrece dispositivos basados en SHA y AES.

Los protocolos inalámbricos más usados son RFID, WiFi, Zigbee y Z-wave. Esto dificulta encontrar una solución heterogénea que se acomode a la diversidad de dispositivos existentes en cuanto a seguridad.

Es necesario contar con buenas practicas a la hora de gestionar servicios IoT, teniendo una visión holística del sistema y todos los elementos usados en una implementación:

- Dispositivos IoT
- La nube
- Aplicaciones móviles
- Interfaces de red
- Software usado
- Uso de cifrado
- Uso de autenticación
- Seguridad física y puertos USB

Debido a que personas o organizaciones maliciosas pueden aprovechar cualquier vulnerabilidad para obtener privilegios en el sistema, se recomienda cambiar las contraseñas por defecto de acceso a los dispositivos IoT porque diferentes malware como Mirai, Bashlite, Linux/Hydra se aprovechan de esta vulnerabilidad para tomar el control de los dispositivos, una vez el hacker ha ingresado al dispositivo lo primero que hace es descargar el malware para ser añadido a una botnet con la finalidad de realizar ataques DDoS a gran escala, con la capacidad de infectar 4000 dispositivos por hora, en su mayoría cámaras CCTV, DVRs y home routers. Aproximadamente se estima que alrededor de medio millón de dispositivos se encuentran infectados, es capaz de generar tráfico desde 200 Gbps hasta 1.2Tbps.

Un dispositivo infectado no solo servirá a sus atacantes para hacer ataques DDoS, entre otros usos esta enviar emails spam, password cracking, key loggers, y minería de cripto monedas. En octubre de 2016 la US Computer Emergency Readiness TEAM (US-CERT) emite las siguientes recomendaciones con respecto al malware Mirai:

- Se recomienda monitorear el puerto ip 2323/TCP de intentos no autorizados para controlar el dispositivo usando la terminal telnet.
- Cambiar las contraseñas predeterminadas por claves fuertes.
- Actualizar el firmware de los dispositivos IoT.
- Deshabilitar el conjunto de protocolos de comunicación Universal Plug and Play (UPnP) a no ser que sea absolutamente necesario.
- Monitorear tráfico anómalo en el puerto 48101, ya que los dispositivos infectados tratan de propagarse usando este puerto.

IV. Sistemas multi agentes

Los sistemas multi agentes consiste en una plataforma distribuida donde interactúan múltiples agentes para la resolución de problemas que no pueden ser resueltos en un sistema monolítico. Un agente es una entidad capaz de percibir su entorno por medio de sensores y actuar de acuerdo con comportamientos establecidos para la resolución de problemas, entre las características más importantes de un agente se encuentran la autonomía, ser proactivos, poder comunicarse con otros agentes y viajar por los diferentes nodos de la red.

La arquitectura de un sistema multi agentes es de suma importancia para su funcionamiento puesto que son los mecanismos con los cuales se tendrá comunicación y coordinación entre los agentes. Entre las arquitecturas existentes se encuentran las basadas en lógica, reactivas, belief desire intention(BDI) y arquitecturas en capas.

BDI es la arquitectura mas popular, sus origines tienen raíces en la filosofía para ofrecer una teoría lógica que define las actitudes mentales de las creencias, el deseo y la intención usando lógica modal. Esta basada en cuatro estructuras de datos: creencias, deseos, intenciones, planes y un intérprete.

Las creencias están compuestas por la información que tiene el agente del entorno, que podrían estar incorrectas o incompletas. Los deseos corresponden a las tareas asignadas a un agente y por lo tanto son sus objetivos, las intenciones son los deseos que un agente se compromete a lograr. En cuanto a los planes especifican procedimientos que se siguen para lograr las intenciones del agente.

En la arquitectura basada en la lógica el entorno es representado simbólicamente y se manipula usando mecanismos de razonamiento. La ventaja de esta arquitectura es que el conocimiento humano es simbólico y por lo tanto fácil de entender, pero es difícil traducir el mundo real a un lenguaje lógico además que los resultados esperados podrían tardar tanto como para dejar de ser una solución útil.

Las arquitecturas reactivas se diferencian de las basadas en la lógica en que no cuentan con un complejo modelo simbólico. Esta implementa toma de decisiones definiendo capas con máquinas de estado finito conectadas a sensores que transmiten información en tiempo real y son activadas por los sensores. La ventaja es que los agentes basados en una arquitectura reactiva responderán más rápido al entorno sin embargo su razonamiento no será tan bueno.

V. FIPA

FIPA (Foundation for Intelligent Physical Agents) es una organización de estándares de la IEEE Computer Society para desarrollar estándares de software para la construcción de sistemas basados en agentes, fue una organización de origen suizo activa desde 1996 hasta el año 2005 con el objetivo de crear especificaciones para agentes heterogéneos, que interactúan y sistemas basados en agentes, entre los logros más importantes de la organización están:

- FIPA-ACL: Un lenguaje de comunicación entre los agentes, un conjunto de lenguajes de contenido (FIPA -SL) y una serie de protocolos de interacción clave desde un simple intercambio de mensajes hasta complejas transacciones
- Frameworks open source para la programación orientada a agentes siguiendo los estándares FIPA tales como JADE (Java Agent Development Framework)
- Una extensión específica para agentes de UML conocida como AUML

Tabla I
DEFINICIONES DE LOS ACTOS COMUNICATIVOS

Communicative act	Base CA
accept-proposal	inform
agree	inform
cancel	disconfirm
cfp	query-ref
confirm	confirm
disconfirm	disconfirm
failure	inform
inform	inform
inform-if	inform
inform-ref	inform
not-understood	inform
propagate	inform
propose	inform
proxy	inform
query-if	request
query-ref	request
refuse	disconfirm; inform
reject-proposal	inform
request	request
request-when	inform
request-whenever	inform
subscribe	request-whenever

- La existencia del proyecto Agentcities que ha creado una red global de ciudades siguiendo los estándares FIPA sistemas multi agentes

Los actos comunicativos (CA) son el corazón del modelo FIPA-ACL y define la comunicación como un conjunto de diferentes CA basados en la teoría del acto del habla. La semántica de FIPA-ACL esta centrada en transferir la actitud mental (modelo BDI) del transmisor a uno o mas receptores, sin considerar modelos sociales o terceros.

VI. APLICACIONES DE SEGURIDAD DE LA INFORMACIÓN CON SISTEMAS MULTI AGENTES

Diversas aplicaciones en el área de la seguridad se han dado a los sistemas multi agentes (MAS). ***Agent-based Intrusion Detectaron Mechanism*** (AIDeM) cuenta con métodos avanzados de detección para confrontar técnicas que producen ataques de denegación de servicio en entornos web con arquitectura SOAP. Se basa en un grupo de agentes diseñados para trabajar inteligentemente entre ellos y tener la capacidad de adaptarse para resolver el problema de la confiabilidad de los mensajes SOAP, la estrategia empleada tiene dos fases para clasificar los mensajes. La primera fase aplica filtros para la detección de ataques sin requerir una cantidad excesiva de recursos usando un clasificador bayesiano ingenuo, mientras que la segunda fase tiene un proceso más complejo con una red neuronal que termina requiriendo una cantidad más grande de recursos.

VII. JADE

JADE (Java Agent DEvelopment Framework) Es un framework open source con licencia LGPL (Lesser General Public Licence Version 2) para el desarrollo de sistemas multi agentes en Java. Ayuda a los desarrolladores en la implementación de este tipo de sistemas brindando herramientas siguiendo las especificaciones FIPA permitiendo al desarrollador preocuparse por la lógica de los agentes que resolverá los problemas para los cuales fueron creados.
Una de las partes más importantes de JADE es su "middleware" que se refiere a una colección de bibliotecas de alto nivel que proveen servicios como comunicación, acceso a los datos, codificadores, control de recursos. Estos servicios también los ofrecen los sistemas operativos, pero para hacer de JADE un sistema multiplataforma se decide implementar estos servicios desde el middleware.

La plataforma de JADE está compuesta por contenedores de agentes que pueden estar distribuidos en los diferentes nodos de la red, los agentes viven dentro de los contenedores los cuales son un proceso Java que proporciona el tiempo de ejecución de JADE. Existe un contenedor principal que sirve de punto de inicio de ejecución de la plataforma, cuenta con un registro que tiene las referencias de los objetos y direcciones de transporte de los nodos que hacen parte de la plataforma (Container table CT), Global agent descriptor table (GADT) es un registro que contiene todos los agentes existentes, su estado y localización, los agentes agent management system (AMS) y directory facilitator (DF). El primero se encarga de la gestión de agentes y el segundo presta el servicio de "paginas amarillas" en la plataforma para que los agentes registren sus servicios y puedan buscar otros servicios.

La "Agent Platform" (AP) cumple con las especificaciones FIPA97 para desplegar agentes, incluye un DF, AMS y "Message Transport Service" (MTS) que permite establecer comunicaciones con agentes que se encuentren en otra AP por medio de mensajes tipo FIPA-ACL

VIII.JADE LEAP

Es una versión modificada de JADE, con la principal característica de correr en dispositivos móviles con el requisito mínimo de contar con la versión 1.0 de MIDP, una versión de J2ME (Java 2 Micro Edition) o dispositivos corriendo el framework .Net de Microsoft. JADE LEAP provee la misma API que JADE, el único cambio se encuentra en los módulos usados para la comunicación, pero son mecanismos invisibles para el usuario y la aplicación.

JADE-LEAP puede correr en dos configuraciones, "***stand-alone***" que ejecuta completamente un container en el dispositivo donde esta corriendo JADE. "***Split***" es la otra configuración, que consiste en dividir el container en un FrontEnd ejecutado en el dispositivo donde se ejecuta el runtime JADE y un BackEnd que es ejecutado en un servidor remoto, están vinculados por una comunicación permanente. Esta configuración esta pensada para dispositivos con restricciones de recursos pues todas las comunicaciones con el container principal son mas rápidas pues no están saliendo y entrando a través de una red inalámbrica.

IX. IMPLEMENTACIÓN

El sistema multi agentes propuesto contará con diferentes agentes que van a estar distribuidos en la red interna que se desea proteger.

ListenerAgent: Se encarga de escuchar los paquetes que llegan a un punto de entrada de la red interna guardando muestras de los paquetes en una base de datos que será estudiada por otro agente que analiza los datos que entran a la red.

SmartAgent: Analiza los paquetes que han sido guardados en la base de datos con algoritmos de aprendizaje automático para determinar ataques DDoS, en caso de que un ataque ocurra el agente le informa a un centinela.

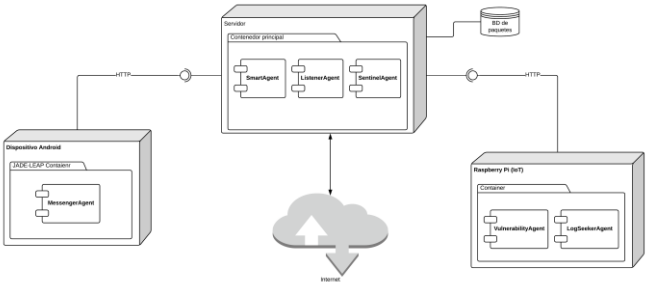
MessengerAgent: Desplegado en dispositivos móviles Android o capaces de soportar la plataforma J2ME. Su tarea es informar y notificar a los usuarios del sistema acerca de los diferentes eventos que pueden ocurrir en la red que usa el sistema SMIOT.

VulnerabilitySeekerAgent: Busca vulnerabilidades en el software y puertos de red abiertos que no estén en uso. En caso de encontrar alguna informa al centinela para que implemente una solución

LogSeekerAgent: Busca posibles accesos no autorizados a través de diferentes protocolos como SSH o TELNET en los logs del sistema. En caso de encontrar accesos no autorizados informa al agente mensajero y al centinela para que implemente contramedidas en el nodo comprometido.

SentinelAgent: Su tarea es implementar contramedidas a ciertos ataques dependiendo de la naturaleza de la amenaza, en caso de ocurrir un ataque DDoS puede bloquear las direcciones IP o redirigir el trafico maligno a otro nodo. Si recibe informes de vulnerabilidades existentes en algún nodo de la red su tarea es cerrar dichas brechas de seguridad. Tiene la característica de ser un agente con movilidad, es decir que puede viajar por los diferentes nodos de la red.

Los agentes van a estar distribuidos de la siguiente forma en la red donde van a ser desplegados:



X. CONCLUSIONES

Gracias a herramientas como la presentada en el proyecto se espera tener un entorno más seguro para los usuarios finales donde el impacto de un ataque puede tener consecuencias irreversibles y grandes daños físicos o cibernéticos. Los sistemas multi agentes tienen un papel muy importante a la hora de crear soluciones distribuidas debido a su capacidad de comunicación en tiempo real logrando incrementar la seguridad no solo en entornos de internet de las cosas, si no en cualquier entorno capaz de ejecutar el sistema creado. Sin embargo, para tener una herramienta ideal se debe tenef desarrollo continuo con herramientas de vanguardia ya que cada día existen personas interesadas en encontrar nuevas brechas de seguridad que se deben solventar para tener un internet más seguro para todos.

REFERENCIAS

[1] F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa, "Jade - a white paper," *Telecom Lab Ital.*, vol. September, no. 3, pp. 6--19, 2003.

[2] S. Poslad, "Specifying protocols for multi-agent systems interaction," *ACM Trans. Auton. Adapt. Syst.*, vol. 2, no. 4, p. 15–es, 2007.

[3] F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa, "JADE: A software framework for developing multi-agent applications. Lessons learned," *Inf. Softw. Technol.*, vol. 50, no. 1–2, pp. 10–21, Jan. 2008.

[4] S. Willmott, J. Dale, B. Burg, and P. Charlton, "Agentcities: a worldwide open agent network," *Agentlink News*, pp. 1–7, 2001.

[5] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," 1998.

[6] D. A. Stanford-Clark, "MQTT." [Online]. Available: http://mqtt.org/.

[7] C. Suchitra and C. P. Vandana, "Internet of Things and Security Issues," vol. 5, no. 1, pp. 133–139, 2016.

[8] R. G. Hernando, "La seguridad en el Internet Of Things," *PublicaTIC*, 2016. [Online]. Available: https://blogs.deusto.es/master-informatica/la-seguridad-en-el-internet-of-things/.

[9] A. Bogdanov *et al.*, "PRESENT: An Ultra-Lightweight Block Cipher," 2007.

[10] E. Paths, "Seguridad criptográfica en IoT," *Eleven Paths*, 2016. [Online]. Available: http://blog.elevenpaths.com/2016/05/seguridad-criptografica-en-iot-ii.html.

[11] C. Pinzón, M. Navarro, and J. Bajo, "AIDeM : Agent-Based Intrusion Detection Mechanism," pp. 347–354, 2010.

[12] C. Pinzon, "Unsupervised Visualization of SQL Attacks by Means of the SCMAS Architecture Unsupervised Visualization of SQL Attacks by Means of the SCMAS Architecture," no. July, 2015.

[13] K. Angrishi, "Turning Internet of Things(IoT) into Internet of Vulnerabilities (IoV) : IoT Botnets," pp. 1–17, 2017.

[14] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer (Long. Beach. Calif).*, vol. 50, no. 7, pp. 80–84, 2017.

[15] E. Bertino and N. Islam, "Botnets and Internet of Things Security," *Computer (Long. Beach. Calif).*, vol. 50, no. 2, pp. 76–79, 2017.

[16] H. Kopetz, “Real-Time Systems,” *Real-Time Syst.*, p. 338, 2011.
[17] M. Suresh and R. Anitha, “Evaluating Machine Learning Algorithms for Detecting DDoS Attacks *,” *CCIS*, vol. 196, pp. 441–452, 2011.
[18] M. Zamani and M. Movahedi, “Machine Learning Techniques for Intrusion Detection.”