Desarrollo de un agente SNMP v3 para modelado de usuario en entornos LAN

N. Lasierra, M. Lopez, J. García, A. Alesanco
Grupo de Tecnologías de la Información (GTC) del Instituto en Investigación en Ingeniería de Aragón (I3A)
Universidad de Zaragoza
50018, Zaragoza
nelia.lasierra@unizar.es, marcs.lopz@gmail.com, jogarmo@unizar.es, alesanco@unizar.es

***Resumen*—En este artículo se presenta el desarrollo de un agente SNMP v3 (*Simple Network Management Protocol*) para el modelado de usuarios en entornos LAN (*Local Area Network*). Este agente establece comunicaciones SNMP tanto con los gestores de red encargados de configurar el proceso del modelado, como con los usuarios de los que recoge información contenida en las MIB (*Management Information Base*) con el objetivo de encontrar un patrón que caracterice su comportamiento. Esta información será procesada y analizada por una red neuronal tipo SOM (*Self Organizing Map*), que permitirá, tras el proceso de aprendizaje, la detección de anomalías respecto al comportamiento normal del usuario. Tanto los parámetros a configurar para definir el modelado de cada usuario como los resultados de la supervisión del agente quedan recogidos en la MIB del modelado contenida en el agente propuesto. De esta forma, el agente desarrollado facilita una herramienta única para modelar a todos los usuarios de la misma red LAN y constituye un sistema totalmente integrado en la arquitectura SNMP. Por último, se presenta un escenario de pruebas para la aplicación de la detección de intrusos del agente propuesto. Se ofrece la información seleccionada para modelar al usuario, así como parámetros de configuración y primeros resultados obtenidos.**

***Palabras Clave*—MIB, modelado de usuarios, SNMP, SOM**

I. Introducción

Conocer las preferencias, costumbres, utilización de recursos o de forma general, información que identifique o caracterice el comportamiento de un usuario, es actualmente un ámbito de gran interés, impulsado por el crecimiento de las redes y la expansión de Internet. Extraer esta información, que permite adecuar el diseño de páginas web al perfil de los usuarios o la personalización de búsquedas guiadas entre otras aplicaciones, es una necesidad que se extiende a redes de área local, donde conocer el comportamiento de sus usuarios proporciona información de gran utilidad para, por ejemplo, distribuir y ofrecer de forma diferenciada el acceso a los recursos disponibles en la red [1]. Especialmente, esta técnica es de gran utilidad en redes en las que existen limitaciones de ancho de banda o existe la necesidad de aplicar técnicas de QoS (*Quality of Service*). Este proceso de extracción y evaluación de la información que permite determinar el comportamiento de un usuario se conoce como modelado del usuario. Sin embargo, la distribucción de recursos no es la única aplicación del modelado en redes LAN. Puesto que la seguridad es uno de los problemas de mayor preocupación que se plantea hoy en día en el mundo de la informática y las redes de comunicaciones, la aplicación más popular del modelado de usuario en redes LAN es la detección de intrusos. A diferencia de otras soluciones tecnológicas desarrolladas para este fin, como la detección de patrones o firmas de ataques conocidos (fundamentalmente existen estas dos técnicas para la detección de intrusos), la idea que implica este proceso es, una vez identificado el comportamiento habitual de un usuario, detectar posibles anomalías en el mismo que se deban a la presencia de un intruso. Esta técnica abre una nueva vía para la detección de intrusos que trata de solventar las limitaciones encontradas en los sistemas anteriores, como por ejemplo, la detección de nuevos atacantes hasta el momento desconocidos por el sistema. Esta tarea del modelado de usuario, implica por tanto una intensa recogida de datos que debe hacerse de forma transparente al mismo y para lo que se hace necesario la búsqueda de técnicas de gestión de redes que permitan este intercambio de información de forma sencilla. Una solución atractiva para esta tarea es la utilización de la arquitectura SNMP [2]. Esta arquitectura de gestión proporciona un interfaz estándar que facilita las comunicaciones entre diferentes dispositivos de red y pone a disposición del gestor gran cantidad de información de forma estructurada a través de las MIB tanto de parámetros de tráfico como de la utilización de recursos o información de configuración del dispositivo

Si bien es cierto que la utilización de SNMP para el modelado de tráfico de usuarios es una apuesta ampliamente extendida [3] y algunos estudios proponen integrar el desarrollo de un detector de intrusos en esta arquitectura [4], hasta lo que nosotros sabemos, ninguna solución propuesta para el modelado de usuarios ofrece una solución global en términos de escalabilidad, seguridad, facilidad de comunicaciones, procedimiento de identificación de patrones y generalización de aplicaciones. Por ello, en este artículo se presenta el desarrollo de un agente SNMP v3 como solución global para el modelado de usuario. Este agente, que aprovecha las ventajas de esta arquitectura tanto para las comunicaciones como para el acceso y almacenamiento de la información analizada, va a permitir que uno o varios gestores integrados dentro de una arquitectura de gestión de red SNMP puedan modelar el comportamiento de todos los usuarios situados en la misma red de área local así como su continua supervisión y detección de posibles anomalías en su comportamiento habitual gracias al desarrollo de una red SOM que analizará la información extraída de cada usuario.

II. Modelado de Usuario

Podemos definir el modelado de usuario como un proceso a través del cual obtenemos una representación de las características relevantes de su comportamiento. De forma general,

este proceso se compone de tres fases: 1) selección de información 2) caracterización y aprendizaje del comportamiento y 3) supervisión y detección de anomalías.

A. Selección de información relevante

La primera tarea a desarrollar para encontrar un patrón que caracterice el comportamiento de un usuario es seleccionar e identificar la información que queremos extraer del mismo. La elección de estos parámetros vendrá condicionada por la aplicación concreta para la que se quiera conocer su comportamiento. Por ejemplo, en experiencias como [1], en las que el objetivo del modelado es clasificar a los usuarios en diferentes categorías o perfiles para la posterior distribución de recursos en el acceso a Internet, se propone la utilización de parámetros de tráfico HTTP, SMTP y FTP. Entre otros, se analiza el ancho de banda utilizado, la duración de las conexiones, tamaño de los paquetes enviados y los puertos utilizados con el objetivo de identificar el uso de aplicaciones *peer to peer* en concreto. Sin embargo, si la aplicación es la detección de intrusos, otros estudios como [5] y [6] proponen para modelar al usuario utilizar parámetros de tráfico ICMP, TCP y UDP (número de paquetes enviados y recibidos, paquetes erróneos, análisis de cabeceras o número de conexiones, etc) así como otros parámetros internos como el uso de memoria y CPU o los comandos tecleados por el usuario. El compromiso que plantea mantener la eficiencia de red y obtener la cantidad de información necesaria para el modelado hace aumentar la importancia de este proceso de selección de parámetros.

B. Caracterización del comportamiento

Una vez seleccionada la información con la que se va a modelar al usuario comienza la fase de adquisición de datos y aprendizaje del comportamiento. Para la recogida de esta información, se ha seleccionado la utilización de la arquitectura de gestión SNMP. Esta es una solución muy popular utilizada por sistemas de modelado de usuario dada la flexibilidad y facilidad de uso que esta arquitectura ofrece, permitiendo de una forma estructurada y efectiva el intercambio de información. Por ejemplo, en [3] y [7] se analiza información contenida en las MIB 2 para la detección de ataques de DoS (Denial of Service). Otros métodos o técnicas ampliamente adoptados para la adquisición de datos de usuario son la utilización de archivos de logs del sistema operativo o sistemas de filtrado de tráfico [8].

Durante esta fase de caracterización del comportamiento, la información extraída del usuario es procesada y clasificada como normal o habitual para el usuario modelado. La duración de este periodo de aprendizaje debe permitir observar y definir de forma representativa los distintos patrones que puede presentar el usuario. Para esta tarea de caracterización del comportamiento de los usuarios existen múltiples propuestas. Algunas de las técnicas más populares tanto para el modelado de usuarios como de tráfico de los mismos, apoyan su funcionamiento en la utilización de redes neuronales, modelos de markov, estudios estadísticos o lógica borrosa , entre otros [6] [9] [10]. En la solución propuesta, el elemento utilizado para el aprendizaje del comportamiento del usuario y decisión de las posibles anomalías observadas, es una red SOM.

Por último, en la fase de test o supervisión, se establecen periodos de encuesta para recoger información del usuario y comparar el comportamiento observado con el comportamiento clasificado como normal con el objetivo de detectar así anomalías en el comportamiento monitorizado.

III. Arquitectura del Agente SNMP para el modelado de usuario

La arquitectura general del agente SNMP v3 propuesto para el modelado de usuario está compuesto por tres bloques: 1) el bloque de comunicaciones, 2) el bloque de la MIB del modelado, y 3) el bloque de las redes SOM (una por cada usuario a monitorizar). Como se muestra en la figura 1, este agente establece comunicaciones SNMP tanto con los usuarios a modelar como con los gestores que configuran y supervisan el modelado de cada uno de ellos. A través de la MIB del modelado (contenida en el agente desarrollado) el gestor puede configurar los parámetros involucrados en el modelado de cada usuario (individualizando así el proceso para cada uno de ellos) así como acceder a los resultados obtenidos de la supervisión del agente. Estos resultados se consiguen tras el procesado de la información extraída del usuario y posterior análisis mediante una red neuronal SOM. El agente desarrollado constituye una herramienta única para modelar a todos los usuarios de la misma red que evita la utilización de nuevas herramientas software en los equipos a modelar y posibilita su uso en los equipos que ya posean una gestión de redes vía SNMP.

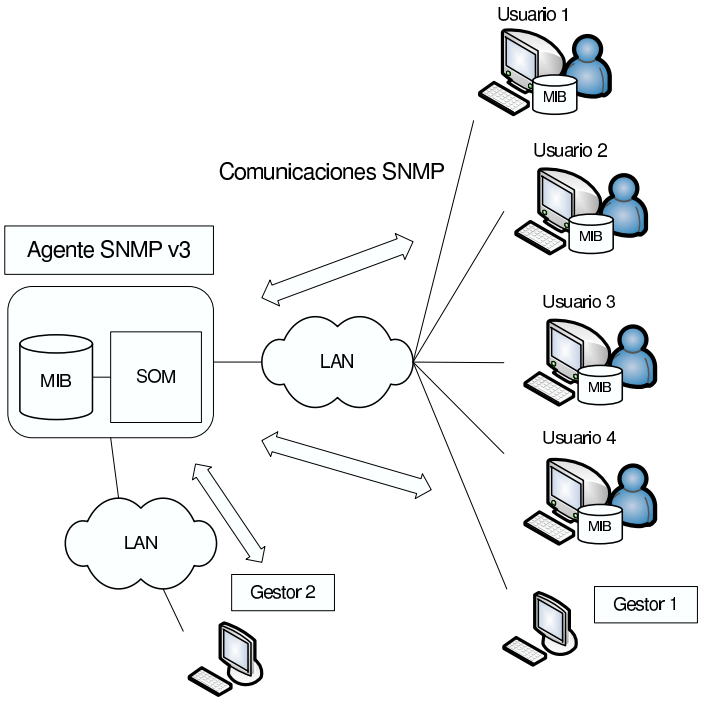

Fig. 1. Funcionamiento general del Sistema.

En la figura 2 se muestra un esquema del funcionamiento e interacción del agente con el gestor y los usuarios, así como de la relación entre los bloques del mismo (comunicaciones, MIB y SOM) que intervienen en cada una de las fases del proceso de modelado del usuario. Mediante el envío de paquetes tipo SnmpGet, el agente extraerá información

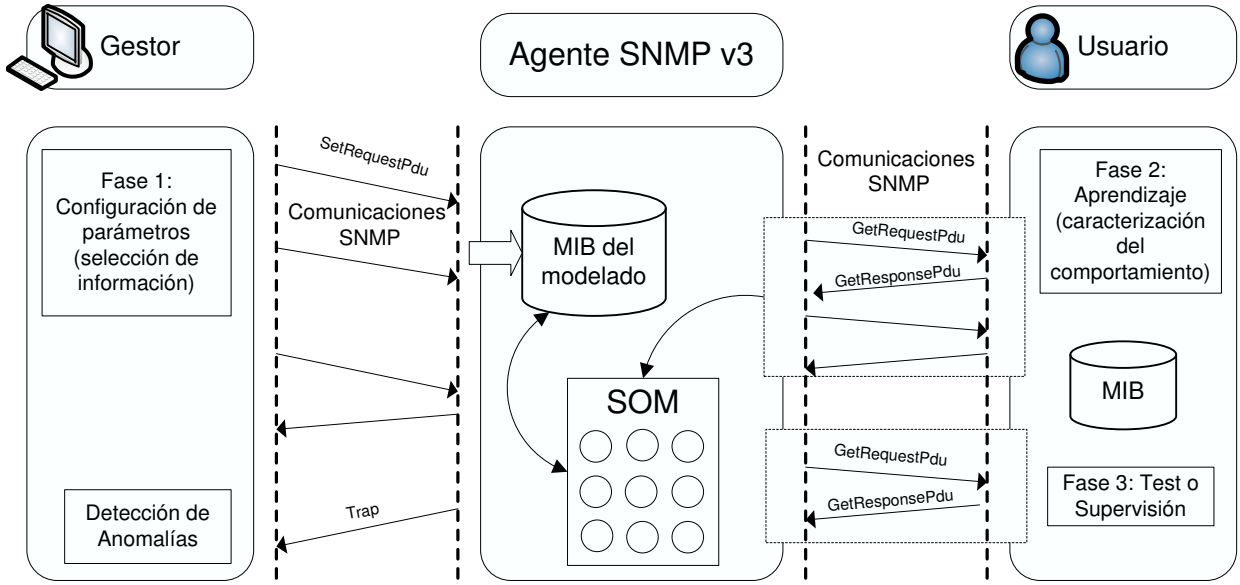

Fig. 2. Comunicacion entre bloques del Sistema.

contenida en las MIB de los usuarios. Esta información será analizada por la SOM asociada al mismo y los resultados obtenidos, almacenados en la MIB del modelado. De esta forma el gestor, a través del envío de paquetes tipo SnmpSet (para la modificación de datos) y tipo SnmpGet (para la lectura de resultados) dispondrá de control total sobre la definición y supervisión del comportamiento de los usuarios. Para el desarrollo de los módulos que componen el sistema se ha escogido el lenguaje de programación JAVA, así como el motor de base de datos MySQL como soporte para la implementación de la MIB propuesta.

A. Bloque 1: Comunicaciones SNMP

Las comunicaciones entre el agente y los participantes involucrados en el proceso del modelado (gestores y usuarios) se realizan mediante el protocolo SNMP v3. Esta versión del protocolo incorpora aspectos de seguridad con lo que el agente diseñado dispone de funciones de autenticación, cifrado y control de acceso [11]. De esta forma queda garantizada que tanto las funciones de gestión como el acceso a la información queda limitado a personas autorizadas. Aunque es recomendable utilizar esta versión, el agente desarrollado soporta además las anteriores versiones del protocolo.

Como se observa en la figura 2, el agente desarrollado se comunica tanto con los gestores que definen y configuran los parámetros del modelado a través de la MIB que implementa, como con los usuarios de los que recoge información para caracterizar y posteriormente supervisar su comportamiento. De esta forma, el agente incorpora dos funcionalidades: por una parte actúa como un agente normal de la arquitectura SNMP capaz de responder a órdenes y peticiones de una estación gestora así como enviar de forma asíncrona información importante no solicitada (traps) y por otra, actúa como un gestor ya que establece comunicaciones SNMP con los agentes de los usuarios para pedir información de forma periódica (lanzamiento de encuestas) acerca de variables contenidas en las MIB (está información puede estar localizada tanto en las MIB2 como en las MIB privadas que implemente el dispositivo).

B. Bloque 2: MIB UserModelling

La MIB del modelado, contenida en el agente desarrollado, constituye una estructura de almacenamiento de todos aquellos valores que intervienen en este proceso. Su implementación ofrece flexibilidad en la definición de parámetros (lo que permite individualizar y particularizar el modelado para cada uno de los usuarios de la red de acuerdo a los criterios del gestor) y permite disponer de un módulo de fácil acceso para el gestor en el que se exponen de forma estructurada los resultados de la monitorización del usuario. Esta MIB que ha sido denominada *UserModelling* y está situada dentro de las MIBs privadas (puesto que el esquema propuesto de modelado sólo se aplica a terminales personales), aporta escalabilidad al diseño propuesto, ya que sirve como base para futuras aplicaciones que quieran extender las funcionalidades del agente.

La MIB implementada está compuesta por 4 tablas: *userModelingControlTable*, *modelingParametersTable*, *userModelingDataTable* y *logUserModelingTable*. Las tablas *userModelingControl* y *modelingParameters* permiten al gestor configurar todas las variables que intervienen en el modelado del usuario. Estas tablas funcionan de forma lógica como una única tabla de ”Control”, a pesar de estar separadas en su defición en dos para facilitar el proceso de configuración de parámetros, cuyo funcionamiento sigue las reglas de las tablas de control RMON [12] . La tabla *userModelingData* es una tabla de lectura para el gestor, en la que se muestran para varias observaciones realizadas por el agente, los resultados obtenidos del modelado del usuario, lo que permite al gestor un seguimiento temporal de la evolución del comportamiento del mismo. Por último, la tabla *logUserModeling*, recoge información relativa a observaciones del comportamiento del usuario identificadas como comportamiento anómalo. A continuación se describen

con más detalle los objetos contenidos en estas tablas así como las funcionalidades de los mismos:

- ***userModelingControlTable***: La tabla de Control es una tabla de lectura/escritura, en la que se definen: los parámetros para identificar a cada uno de los usuarios a modelar (dirección IP del usuario a modelar e índice del modelado, ambos necesarios ya que cada usuario puede ser modelado por varios gestores y con distintos parámetros), aspectos relacionados con la configuración de la red neuronal (dimensiones, factor de aprendizaje, tiempo de entrenamiento o radio de vecindad entre otros), información relativa para la detección de anomalías (umbrales y eventos asociados), información acerca del gestor que controla cada proceso de modelado configurado así como los intervalos de tiempo involucrados en los periodos de encuesta para el aprendizaje y supervisión del comportamiento. Como se observa en la figura 3, para la recogida de datos del usuario, se realizan encuestas al usuario de forma periódica. La duración de estas encuestas, el tiempo de activación entre ellas, así como el tiempo entre envío de paquetes de petición de datos por parte del usuario al agente, permitirán la configuración de los periodos de test y aprendizaje del comportamiento.

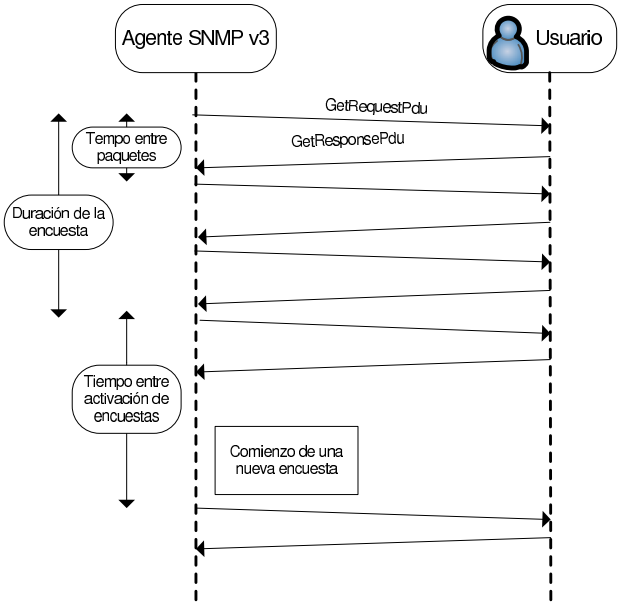

Fig. 3. Esquema de encuestas

- ***modelingParametersTable***: Esta tabla permite al gestor especificar la información contenida en las MIB que va a servir para modelar el comportamiento del usuario. Puesto que muchas veces la información que el gestor considera más interesante para caracterizar el comportamiento del usuario no se encuentra de forma directa en las MIB, sino que es consecuencia de la combinación de dos que si lo están, por cada variable a utilizar en el modelado, el gestor deberá especificar dos OIDs, así como la relación que existe entre ambas para conseguir el resultado deseado.
- ***userModelingData*** : En esta tabla se recogen los resultados de las encuestas realizadas a cada usuario. Es decir, en ella se muestran los valores observados para cada una de las variables definidas, el instante temporal de la observación proporcionada así como la salida de la red neuronal para dicho comportamiento observado. El contenido de esta tabla va a permitir al gestor observar el comportamiento del usuario durante un intervalo de tiempo, ya que asociado a cada uno de ellos se define un buffer circular que permite mostrar la evolución de los valores observados.
- ***logUserModelingTable*** : En función del evento definido en la tabla de control se registrará o no una entrada en esta tabla cada vez que la red neuronal detecte un comportamiento anómalo en el usuario. En esta tabla se muestra la información relativa a todas las observaciones realizadas al usuario que indican claramente un comportamiento distinto al habitual. Su funcionamiento es similar a la anterior, ya que para cada usuario modelado se define un buffer en el que se muestran todas las anomalías detectadas en el periodo de monitorización.

Para notificar al gestor cambios en la evaluación del comportamiento del usuario e informarle acerca de anomalías detectadas, la MIB *UserModelling* define el envío de dos tipos distintos de Traps: *BehaviourAlarm* y *WarningAlarm*. La definición y posterior envío de este tipo de paquetes permite al gestor aumentar los intervalos de encuesta (al agente propuesto) para revisar el comportamiento de los usuarios modelados. De esta forma, se reduce el tráfico de la red y se permite al gestor el conocimiento total sobre cambios en esos parámetros.

C. Bloque 3: SOM.

Para analizar la información extraída del usuario y posteriormente facilitar la toma de decisiones (comportamiento normal o anómalo), se utilizan redes SOM [13]. Los mapas autoorganizados (SOM) son un modelo de red neuronal muy utilizado en la práctica para el análisis y visualización de grandes cantidades de datos. Este tipo de redes se basan en el aprendizaje no supervisado, de forma que la red es entrenada con un conjunto de patrones de entrada sin asociar una salida deseada para cada una de ellas durante esta fase. En función de las similitudes observadas entre los datos de entrada se van formando, de forma adaptativa, grupos de neuronas a la salida con características similares, de forma que se representa en las agrupaciones de las neuronas de la red una estimación de la función de densidad de probabilidad de las entradas. Así, el funcionamiento de estas redes se basa en establecer una correspondencia entre los patrones de entrada y un espacio bidimensional de salida, de forma que los representantes de cada agrupación obtenida queden espacialmente correlados y alejados de aquellos respecto a los que presentan mayores diferencias.

A pesar de que existe una clara tendencia a la elección de redes MLP (*Multilayer Perceptron*) en la mayoría de sistemas en los que se implementa una red neuronal para la toma de decisiones, para el sistema de modelado propuesto la implementación de una red SOM es la más adecuada dado que este tipo de redes presentan propiedades como facilidad de implementación, flexibilidad y capacidad de generalización,

pero fundamentalmente su elección queda determinada por dos razones: 1) el número de comportamientos "normales" que puede presentar el usuario es en principio desconocido y 2) sólo es posible entrenar la red con patrones de comportamiento normal del usuario y no anómalos (ya que son desconocidos). Antes de comenzar el modelado de un usuario, no podemos asumir que el comportamiento observado sea estable en el tiempo, por ello, la estructura utilizada para caracterizar su comportamiento debe permitir identificar cada uno de estos comportamientos observados durante la fase de entrenamiento y aprenderlos como normales para el mismo. Por otra parte, puesto que para la fase de aprendizaje sólo conocemos la información relativa a lo que es normal o habitual en el usuario, el proceso de entrenamiento de la red debe realizarse únicamente a partir de patrones positivos. Dadas las condiciones del sistema diseñado para el modelado, la implementación de una red SOM es sin duda, para esta aplicación, la solución más adecuada.

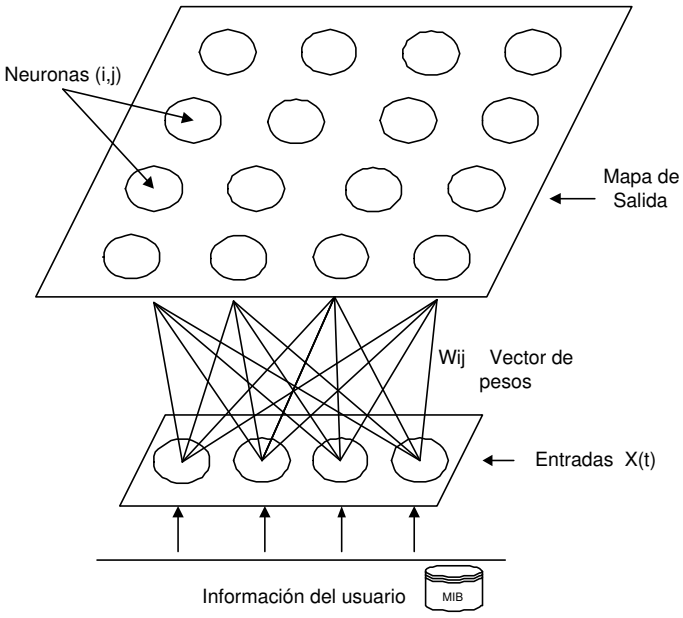

Fig. 4. Arquitectura general de una red SOM

En la figura 4 se muestra la arquitectura típica de una red SOM. La información extraída del usuario será procesada y presentada como entrada a la red neuronal, por lo que el número de elementos del vector de entrada (X(t)) queda determinado por el número de variables seleccionadas para modelar al usuario. El número de neuronas del mapa de salida, queda determinado por las dimensiones especificadas por el gestor. Asociado a cada una de ellas (denominadas en la figura como neuronas ij de acuerdo a su posición en el mapa), existe un vector de pesos wij(t) compuesto por un número de elementos igual al número de variables de entrada, de forma que su situación en el mapa y actualización del vector de pesos está directamente relacionada con las observaciones presentadas a la entrada de la red.

Para cada uno de los usuarios a modelar se asocia una red de este tipo, de forma que la implementación de este módulo va a permitir de forma sencilla, representar todos los posibles comportamientos normales que pueda presentar un usuario. Notar que tanto el número de comportamientos normales que puede presentar un usuario como el perfil de estos comportamientos es diferente de los que pueda presentar otro, ya que lo que para un usuario es normal para otro puede no serlo, por ello es necesario disponer de una red SOM para supervisar de forma individual el comportamiento de cada uno de ellos. Durante la fase se aprendizaje se procesará la información extraída de las encuestas a los usuarios y presentará a la red como datos de entrada normalizados a valores comprendidos en el rango [0,1]. Así, de forma adaptativa se irán actualizando los pesos de las neuronas (inicializados de forma aleatoria) hasta quedar organizada de acuerdo a las observaciones presentadas. Para la actualización de los pesos se ha programado una función gaussiana, ya que dadas sus características, ofrece rapidez de convergencia (lo que asegura una ordenación más rápida del mapa). Tanto para el factor de aprendizaje como para el radio de vecindad de la función gaussiana (parámetros configurables por el gestor) se ha programado una función exponencial, de forma que el ajuste entre la entrada y el vector de pesos de la neurona va disminuyendo a medida que se desarrolla el proceso de aprendizaje así como el radio de vecindad va tendiendo a valores más pequeños para poder incurrir en un error de desajuste mínimo al final del aprendizaje.

Durante la fase de supervisión o test, se presentará el patrón de observación a la SOM (obtenido de la información del usuario) y se indicará la clasificación del comportamiento observado. El agente implementado identifica en base a los resultados de la SOM los posibles comportamientos normales que presenta el usuario (diferenciación de zonas en el mapa de salida de la red) y notifica al gestor tras el procesado de la información extraida del usuario, si la observación del comportamiento es normal (se corresponde con alguna de las zonas detectadas como comportamiento habitual) o es anómala (no se corresponde con ninguno de los comportamientos obtenidos en la fase de aprendizaje). Esta decisión se toma en base a la combinación de dos criterios de distancia: 1) el vector de pesos de la neurona ganadora y 2) la distancia en posiciones a la neurona seleccionada como representativa del comportamiento normal del usuario.

IV. Aplicación a la detección de intrusos

Una de las aplicaciones más populares del modelado de usuarios es la detección de intrusos. Su funcionamiento se basa en encontrar anomalías o desviaciones respecto al comportamiento habitual del usuario que se deban a la presencia de un atacante. Actualmente, nuestro trabajo se centra en el estudio del modelado de usuarios para esta aplicación. A continuación se exponen los estudios realizados para la selección de información utilizada en el modelado del usuario, así como los parámetros seleccionados para la configuración del mismo, escenario de pruebas propuesto y los primeros resultados obtenidos.

A. Selección de Parámetros.

Para realizar una adecuada selección de parámetros se ha llevado a cabo estudio acerca de como la actividad de los diferentes tipos de ataques que existen afectan a la modificación del comportamiento de los usuarios atacados. Podemos

clasificar los diferentes ataques en 4 grandes grupos: ataques de denegación de servicio (DoS), Root Attacks, Remote to User Attacks y Probing [14].

- *DoS*: El objetivo final de estos ataques es la inutilización del sistema víctima. Un elevado consumo de CPU, la saturación de la memoria o la sobrecarga de tráfico IP pueden ser las consecuencias de la actividad de este tipo de ataques.
- *Root Attacks*: Intentan conseguir los privilegios de superusuario en el sistema víctima para robar información valiosa, borrar las huellas del ataque, atacar otros sistemas, etc. La detección inmediata debería ser más complicada, ya que no existen parámetros estándar SNMP para monitorizar los inicios de sesión de usuario en un sistema operativo. Sin embargo, a partir de las acciones posteriores del atacante y a su uso anómalo de los privilegios de usuario se podrá realizar una detección implícita o indirecta del ataque.
- *Remote to User*: Se asemejan al anterior grupo en que el éxito reside en la elevación de privilegios para el atacante. No obstante, el control del sistema es parcial, ya que depende de los privilegios del usuario comprometido en el sistema. El estudio de la posible detección es similar al anterior grupo.
- *Probing*: Consiste en obtener la mayor información posible del sistema atacado. Suele ser precedente de un ataque de otro grupo, es decir, con el probing se estudian las debilidades y el ataque en sí es el uso indebido de ellas. Al igual que ocurre para ataques *DoS*, el estudio de variaciones en parámetros de tráfico (paquetes UDP, ICMP o TCP enviados y recibidos, número de conexiones realizadas, paquetes erróneos, etc), permitirá una detección más sencilla de este tipo de ataques.

Cabe destacar que normalmente, un ataque no pertenece a un sólo grupo ya que está formado por una sucesión de ataques de distinto tipo. Por ejemplo, un atacante podría obtener información del sistema mediante probing, luego obtener privilegios de root usando esa información, para posteriormente utilizar la máquina víctima y realizar así un ataque DoS a un segundo sistema.

En total se han seleccionado 11 variables con las que modelar el comportamiento del usuario. Estos parámetros escogidos, extraídos de la MIB2 (estadísticas de red) y MIB25 (*Host-resources*), son los siguientes:

- (1) Datagramas UDP con puerto desconocido
- (2)(3) Ancho de banda de entrada y salida
- (4)(5) Segmentos TCP de entrada y salida
- (6)(7) Datagramas UDP de entrada y salida
- (8)(9) Número de conexiones TCP y UDP
- (10) Consumo de CPU
- (11) Uso de memoria RAM

La utilización de las 7 primeros parámetros (variables de tráfico) resulta de gran utilidad para la detección de ataques de *Probing*, así como la utilización de los 10 últimos parámetros especificados para el modelado del usuario (todos salvo el primero) proporciona información útil la detección de ataques de *DoS*. De forma general, cualquiera de estos parámetros podría quedar modificado como consecuencia de ataques de *Root* o *Remote to User*, ya que normalmente estos conllevan el lanzamiento de un ataque de DoS o Probing (esto dependerá de las acciones realizadas al adquirir los permisos del sistema).

B. Entorno de pruebas y primeros resultados

Para verificar la eficacia del sistema propuesto, hemos planteado un proceso de pruebas compuesto por dos fases: 1) simulaciones de comportamiento de usuario en entornos virtuales y 2) modelado de usuarios sobre escenarios reales. El objetivo de la primera fase de pruebas ha sido identificar los valores de configuración que garantizan el correcto funcionamiento de la SOM, así como determinar la capacidad de la red neuronal para identificar durante el aprendizaje distintos perfiles de comportamiento normal de un usuario y clasificar posteriormente durante el periodo de test las observaciones realizadas. Para ello, se ha definido un fichero compuesto por 2400 vectores de datos (utilizado para el aprendizaje) que simulan el comportamiento de un usuario cuyo perfil hemos determinado a través de observaciones realizadas a un usuario real con un software de gestión de redes. Para el periodo de test se ha definido igualmente un fichero de comportamiento compuesto por 1500 vectores de test que simulan tanto el comportamiento normal del usuario como comportamientos alterados por la presencia de intrusos. Estos últimos vectores de test se han conseguido a través de modificaciones del patrón original utilizado para el aprendizaje de la red de acuerdo a la forma de actuar de distintos ataques. En concreto se han simulado ataques de DoS que modifican el consumo de la CPU y el volumen de tráfico de entrada y salida, actuando tanto de forma individual como de forma combinada entre ellos.

El número de variables y funciones que se deben elegir en cuanto a la estructura y configuración de la red neuronal, es extenso. Para una convergencia óptima de una red SOM se recomiendan de 50 a 100 iteraciones por neurona aproximadamente [13], por lo tanto, determinar este parámetro es importante ya que constituye una referencia para definir el resto de variables. En base a las pruebas realizadas, el número de iteraciones por neurona óptimo para un correcto funcionamiento de la red se ha estimado en 40. Tras la realización de pruebas con diferentes configuraciones de las dimensiones de la red (número de filas y columnas), se optó por una red 6x4 ya que el mapa de salida de la red permite una representación clara de las zonas de actividad del usuario siendo el tiempo estimado para el aprendizaje del comportamiento(consecuencia de las variables definidas) de 16 horas. Este periodo permitirá identificar totalmente el comportamiento del usuario estudiando su actividad durante dos jornadas laborales típicas, en las que probablemente, el patrón observado se repita.

En la tabla I se recogen los resultados de las pruebas realizadas en la primera fase utilizando la configuración de la red descrita en las líneas anteriores. En ella se detalla el número de comportamientos normales identificados para el usuario así como la tasa de falsos positivos (vectores de comportamiento normal clasificados como comportamiento

anómalo), el porcentaje comportamientos anómalos detectados debido a la presencia de un intruso (Inrusos Detectados) y el porcentaje de comportamientos anómalos no detectados (Intrusos No Detectados). Tras las pruebas realizadas, hemos fijado en 0.28 el umbral utilizado para la detección anomalías (para la distancia euclídea entre vectores de pesos) estableciendo así un compromiso entre el porcentaje de falsos positivos y la tasa de intrusiones no detectadas.

Tabla I
RESULTADOS OBTENIDOS DE LA FASE 1 DE PRUEBAS.

Nº Comportamientos Normales	% Falsos Positivos	% Intrusos Detectados	% Intrusos No Detectados
4	0.9%	86.6 %	13.4%

Actualmente estamos trabajando en segunda fase del proceso de pruebas modelando el comportamiento de diferentes usuarios reales. En las primeras pruebas realizadas, no se ha comprometido el sistema con agentes externos (como puedan ser infecciones víricas o ataques desde otros equipos), aunque el trabajo futuro incluirá la introducción de ataques con la finalidad de evaluar la capacidad del sistema para la detección de anomalías sobre diferentes escenarios. A partir de los resultados obtenidos en la fase anterior, hemos modelado a cada uno de los usuarios durante un periodo de aprendizaje de 16 horas no consecutivas (equivalente a 2 jornadas laborales típicas), eligiendo para ello una red SOM de 11 parámetros de entrada y un mapa de salida de dimensiones 4x6. Posteriormente, hemos supervisado durante 19 horas el comportamiento de los usuarios con el objetivo de determinar la tasa de falsos positivos obtenida con la red propuesta. Para esta fase se han establecido periodos de encuesta de 6 minutos, en las que el tiempo entre envío de paquetes ha quedado fijado a 1 minuto y el tiempo entre activación de encuestas en de 30 minutos. Es importante destacar, que tanto la configuración propuesta de la red como la elección de parámetros de tiempo, permiten de forma general caracterizar el comportamiento del usuario, y que por tanto, la solución propuesta puede ser utilizada en otra aplicación del modelado de usuarios distinta a la detección de intrusos.

Tras la fase de aprendizaje o entrenamiento del comportamiento del usuario 1, hemos diferenciado, en base a los resultados de la SOM, 5 zonas en su mapa de salida. Las zonas 1, 2 y 3 (marcadas en la figura 5) se corresponden con 3 comportamientos normales identificados en el usuario. La zona 4 o zona de transición ha quedado definida por su cercanía a las zonas de comportamiento normal del usuario y por tanto la decisión de comportamiento anómalo tras la activación de una neurona de esta zona en la fase de test vendrá condicionada fundamentalmente por la distancia de su vector de pesos. Por último, la última zona identificada (marcada en la figura 5 con neuronas en blanco) es una zona de neuronas que no han obtenido ninguna victoria (o un número insignificante) durante la fase de aprendizaje por lo que sus pesos no han sido apenas modificados respecto a su inicialización. Por lo tanto, la activación de una neurona de esta zona durante la fase de test, indicará la detección de un comportamiento anómalo en el usuario. Estas zonas quedan representadas en la SOM de la figura 5, en ella, el aumento de la frecuencia de victorias de cada neurona se representa con un color más oscuro en la escala de grises.

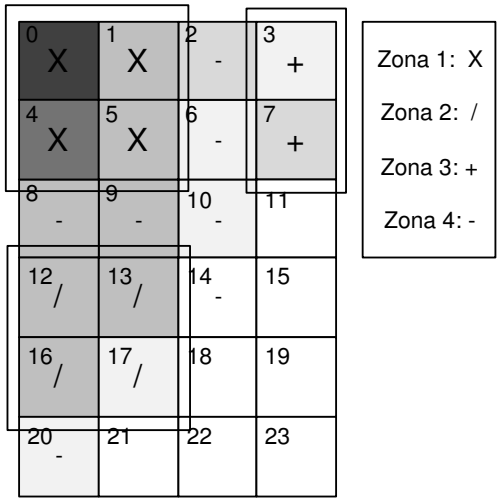

Fig. 5. SOM tras el aprendizaje.

A continuación se resumen los parámetros característicos de cada zona:

- Zona 1: No existe un consumo de ancho de banda apreciable, ni número elevado de conexiones UDP o TCP. El consumo de CPU es muy bajo, en torno al 1%.
- Zona 2: Es una modificación del perfil de la zona 1, en el cual se han establecido más conexiones TCP. El consumo de CPU es ligeramente mayor, aunque sigue siendo bajo, situándose entre el 4 y el 8%.
- Zona 3: Es la zona de mayor uso de procesador, entre el 20 y el 30%, además de una subida del uso de ancho de banda, tanto de subida como de bajada (aproximadamente 1 Mbps en ambos sentidos).
- Zona de transición o zona 4: Son neuronas cercanas a una o más de las zonas anteriores. Sus pesos son intermedios entre los de las zonas adyacentes a ellas o próximos a los de una, si sólo se encuentran cercanas a ella. Han llegado a este estado de pesos como consecuencia de estar situadas en el área de vecindad durante las victorias de las zonas normales, incluso llegando a obtener alguna victoria esporádica en transiciones de un perfil a otro.

Tabla II
RESULTADOS DE LA FASE DE TEST.

Zonas	1	2	3	Transición	Anómala
Porcentaje en test	74,7%	3,2%	5,3%	15,8%	0,53%

Durante el periodo de test, en el se han introducido a la red un total de 190 vectores de entrada correspondientes a observaciones del comportamiento del usuario, se obtienen los datos reflejados en la tabla II. En resumen, un 99,47% de las entradas son catalogadas como normales frente a un

0,53%, que nos indicarían falsos positivos en la detección de comportamiento anómalo.

En base a estos resultados, se concluye que todavía es necesario trabajar en la mejora del procesado previo de los parámetros de entrada para optimizar el modelado de usuario ya que la influencia de variaciones en ciertos parámetros es superior a otros. Sin embargo, los resultados obtenidos en el aprendizaje y test de un comportamiento normal, que presentan una separación de diferentes perfiles en zonas, una sencilla caracterización de cada perfil con los pesos de las neuronas y, por último, una tasa baja de falsos positivos, parecen mostrar una buena aproximación del comportamiento de la red a lo que de ella se espera para nuestra aplicación.

V. Conclusiones y Líneas Futuras

En este artículo se ha presentado el desarrollo de un agente SNMP v3 orientado al control y supervisión de comportamientos de usuarios en entornos LAN. Para ello se han diseñado e implementado los bloques que integran el sistema completo: establecimiento y control de comunicaciones SNMP entre los participantes, redes SOM y diseño de la MIB del modelado. De esta forma, el agente propuesto permite la integración del modelado de usuario en un sistema de gestión de redes global, dejando a elección del gestor la configuración de parámetros necesarios para el proceso en función de la aplicación del mismo. Aunque existe libertad en la elección de estos parámetros, se ha ofrecido una propuesta tanto para la configuración de la red neuronal como para el establecimiento de los tiempos implicados en los periodos de encuesta, que de forma general, pueden ser utilizados para cualquier aplicación del modelado de usuario. Como ejemplo de aplicación, se ha presentado con mayor detalle la utilización del agente para la detección de intrusos. Tras un estudio de las consecuencias que la actividad de un intruso provoca en el comportamiento del usuario, hemos realizado una selección de parámetros, optimizado la configuración del SOM para el aprendizaje del comportamiento del usuario y realizado pruebas muy preliminares sobre un escenario real.

Actualmente estamos trabajando en la optimización del proceso de modelado para la aplicación de la detección de intrusos. Entre las tareas a desarrollar, se encuentra la mejora del sistema para la detección de comportamientos y definición de umbrales, así como el estudio de la correspondencia temporal entre el comportamiento observado y la activación de los comportamientos aprendidos del usuario. Además, se está llevando a cabo la segunda fase del proceso de pruebas. En esta fase se está modelando a usuarios reales con distintos comportamientos y utilizando ataques reales para el periodo de test. Este proceso permitirá generalizar la eficiencia del modelado y selección de parámetros de configuración.

Referencias

[1] K.Mizanian, R.Zakeri, S.Tasharrofi, M.Analoui "User Modeling Using Network Layer Information", *Proceedings of the International Conference on Networking,International Conference on Sytems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06)*,IEEE Publishing, 2006.

[2] J. Case, M. Fedor, M. Schoffstall, J.Davin "Simple Network Management Protocol", *Informe Técnico*, Internet Engineering Task Force(IETF), RFC 1157,1990

[3] J.Li, C.Manikopoulos, "Early Statistical Anomaly Intrusion Detection of DOS Attacks Using MIB Traffic Parameters", *Proceedings of the 2003 IEEE*.

[4] P.Astithas, G.Koutepas, A.Moralis, B.Maglaris "SIDS - A System for enterprise-wide Intrusion Detection", *International System Security Engineering Association Conference '01*, Orlando,USA, 2001

[5] S. Kompella, S. Singh, G. Varghese, "On Scalable Attack Detection in the Network", *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, 2007.

[6] L. Vokorokos, A. Balaz, M. Chovanec, "Intrusion Detection System using Self Organizing Map", Acta Electrotechnica et Informatica, 2006

[7] W. Lee, R. K. Prasanth, B. Ravichandran, R. K. Mehra, "Proactive Detection of Distributed Denial of Service Attacks using MIB Traffic Variables, A Feasibility Study", *Proceedings of the 7th IFIP/IEEE International Symposium on Integrate Network Management*, Seattle, WA, 2001.

[8] M. Barla, M. Bielikova, "Estimation of User Characteristics using Rule-based Analysis of User Logs", *11th International Conference on User Modeling*, Greece,2007.

[9] S. Hyun, W. Suk Lee, "An Anomaly Intrusion Detection Method by Clustering Normal User Behaviour", *Computers and Security*, vol. 22, no. 7, pp. 596-612, 2003.

[10] F. Astithas, G.Koutepas, A.Moralis, B.Maglaris, "Modelling Network Traffic as alfa -Stable Stochastic Processes. An Approach Towards Anomaly Detection", *VII Jornadas de Ingeniería Telemática,JITEL 2008*,2008

[11] J. Case, R. Mundy, D. Partain, B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", *The Internet Society*, RFC 2570, 1999

[12] J. Waldbusser, "Remote Network Monitoring Management Information Base", *Informe Técnico*, Internet Engineering Task Force(IETF), RFC 1757, 1999

[13] B. Martinez del Brio, A. Sanz Molina, "Redes Neuronales y Sistemas Borrosos", 3rd ed,Ed RA-MA ISBN 8478977430, 2006

[14] K. Kendall, "A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems", Master Thesis of Engineering in Electrical Engineering and Computer Science,1999