Propuesta teórica de Webmining para Detección de Intrusos mediante Lógica Borrosa

David Campoy
Master de Software Libre
{dcampoym}@uoc.es

Resumen. Las aplicaciones técnicas basadas en Lógica Borrosa para la detección de intrusos (IDS), están demostrando ser un enfoque adecuado para solucionar muchos de los problemas de seguridad. Esta investigación propone el modelo teórico necesario para el desarrollo de un método de detección de intrusos web basado en lógica difusa, diseñado para identificar el ataque mediante reglas y ofrecer el grado del ataque mediante lógica difusa. La actividad que no cumpla con las reglas borrosas tras una parametrización de experto es considerada acceso malicioso.

De esta manera, el principal propósito de la presente investigación consistirá en elaborar un modelo de Sistema Experto Borroso aplicado a la Detección de Intrusos WEB, para ello la propuesta consiste en profundizar en los dos aspectos siguientes: el diseño de un SBRB y aplicar la adaptación de estos modelos en "minería de datos" relativos a la detección de intrusos. La presenten investigación muestra las reglas para identificar tráfico malicioso en web-logs y se presenta a nivel teórico el sistema de SBRB.

Términos Generales. Seguridad, web

Palabras clave. Minería de datos, Detección de Intrusos, Web Logs, Sistemas Basados en Reglas Borrosas (SBRBs)

Introducción

Con la proliferación de amenazas a la seguridad de Internet; como denegación de Servicio (DoS), Phising, SQL injection y Cross Site Scripting es de vital importancia desarrollar algoritmos que identifiquen los intentos de acceso maliciosos. La seguridad es fundamental para la infraestructura de la información digital. Los pilares en la detección de intrusos se basan en los complejos sistemas IDS, que definen una barrera de seguridad advirtiendo de que un sistema está siendo atacado o es objeto de actividades no permitidas. El IDS supervisa las actividades dentro de la red y alerta a los administradores de la actividad sospechosa que compromete la integridad, confidencialidad y disponibilidad. La integridad se puede comprometer cuando los intrusos son capaces de modificar los datos de una base de datos (SQL injection) e incluso el código de una página web (Cross Site Scripting). La confidencialidad puede ser violada cuando un usuario puede acceder a contenidos sin autorización (WebAdmin). Disponibilidad, cuando el atacante consigue que el sistema web deje de estar operativo, para ello se suelen utilizar denegación de servicio DoS o DdoS.

Los métodos actuales de detección de intrusos se basan en el análisis de tráfico de red (registros de auditoría), centrados en el acceso a nivel de sistema operativo. El análisis consiste en la detección de anomalías, cuando un usuario se desvía de lo que se considera habitual. En este trabajo, nos centraremos en la detección de anomalías a nivel de usuario web, mediante el análisis de tráfico que genera los servidores de aplicaciones. Los ficheros logs contienen información de la actividad web y es de gran valor para obtener un esquema del comportamiento de las usuarios por la web, y de este modo construir un modelo para su extracción de datos y detección. La minería de datos se aplica a grandes volúmenes de datos de tráfico para buscar patrones, y puede proporcionar información de gran valor, tanto para las seguridad como para el conocimiento del comportamiento de las visitas.

Es frecuente encontrarse con grandes volúmenes de datos de auditoría web, que incluyen varias variables y de distinta naturaleza. Interpretar los datos recae en la capacidad de los administradores para obtener reglas útiles y descubrir tendencias. Esta tarea se encuentra dentro de la Minería de Datos. En muchas ocasiones y dependiendo de la naturaleza de los datos, el aprendizaje es dirigido porque se conocer la variable a caracterizar.

Son pocas las investigaciones realizadas en la detección de intrusos orientada a web-logs, existe mucha literatura sobre la detección y previsión a nivel general pero no como sistemas especializados en web. Esta investigación pretende cubrir ese vacio en donde millones de servidores son atacados y se encuentra comprometida la seguridad de los usuarios.

El objetivo de esta investigación es la de proponer un método de detección de intrusos utilizando lógica difusa, tomando como variables de entrada los datos de los ficheros web-logs.

El presente artículo está estructurado en tres partes, en primer lugar se comenta el estado actual del datamining aplicado a la detección de intrusos, en segundo lugar se describe el sistema propuesto de lógica difusa mediante el método de Mamdani y por último líneas futuras.

Estado del arte

Internet se ha convertido en una tecnología imprescindible para nuestras vidas. Sorprende la capacidad de generar nuevos servicios y aplicaciones sin ningún tipo de relación, desde escaparates de comercio electrónico hasta funcionar como medio de transmisión de VozIP. Además, con los últimos avances en movilidad es más fácil acceder a Internet a través de los dispositivos móviles (tables, móviles, PC Pocket, ...), por lo que prácticamente la sociedad vive conectada a la información. La información de Internet está alojada en los llamados servidores dedicados cuyos datos deben mantenerse a salvo de cualquier intrusión de modo que la seguridad en este sector es crítica. El NIST (National Institute of Stans and Technology) [5] define la detección de intrusos como los procesos que monitorizan los eventos en los sistemas de red y procesan dichos eventos en busca de patrones que indiquen anomalías. El objetivo principal de un sistema para la detección de intrusos debe ser la detección oportuna del ataque, lo ideal sería que fuera en el mismo momento que se está llevando a cabo [2].

Los servidores dedicados, en especial los servidores WEB a través del puerto 80, generan gran cantidad de eventos registrando los impactos de las visitas. En la actualidad la mayor parte de las soluciones es mediante el filtrado de paquetes IP [9], se trata de un proceso de análisis básico, en dónde no se realiza un estudio de los eventos anteriores, simplemente se filtra de manera general un tipo de evento. Uno de los problemas que presentan los ficheros logs generados por los servidores web es la gran cantidad de

información no útil, esta información puede dificultar la detección de intrusos, la Universidad Aristotle ha realizado una investigación liderada por Nanopoulos [1] en donde se indentifica la información no útil.

El artículo de Joshila Grace [13] muestra la información que contienen los weblogs, indica que la información básica a analizar está formada por eventos compuestos por; *user name, visiting Path, Path Traversed, Time stamp, page last visited, success rate, user-agent, url y request type*. El análisis de los datos de la actividad de usuario web generará patrones del comportamiento de las visitas a través del sitio web, del mismo se puede aplicar para detectar patrones anormales de comportamiento (posibles ataques al sitio web).

K.R. Suneetha publicó en Internal Jornal of Computer Science and Network Security (2009) [14], la eficacia del datamining en los ficheros weblogs para la identificación de usuarios reales.

En la actualidad se está haciendo un esfuerzo a través de varias investigaciones para comprobar la eficacia de distintos modelos de detección de intrusos mediante data mining. Para conseguirlo se analizan grandes ficheros de eventos (logs de servidores, logs de seguridad Unix, ...), se han creado múltiples proyectos de investigación creando así diferentes tipos de sistemas.

De manera general los métodos data mining [5] necesitan de un contenedor de información estructurada, que tras un método de conocimiento y un entrenamiento el sistema aprenderá y detectará los eventos sospechosos. Los servidores webs, generan ficheros logs que pueden llegar a almacenar gigas de información de los movimientos de las visitas webs, entre estos eventos se encuentran trazas de ataques y la detección de estos patrones no es fácil. Por ello las investigaciones realizadas en el campo de detección de intrusos mediante técnicas datamining resultan de gran valor en la aplicación de weblogs.

El uso de técnicas para la detección de intrusos, por medio de logs [2] y sus correspondientes técnicas de minería de datos, es relativamente frecuente, se han encontrado patrones de asociación descubiertos en la red, tales como: exploración web (web scan); ataques específicos a una máquina; alto número de conexiones TCP anómalas sobre el puerto 80.

Es importante recordar que las investigaciones que se están realizando en el campo de data mining aplicado a la detección de intrusos se basa en la detección de patrones de grandes ficheros cuyo contenido está formado por eventos generador por los sistemas informáticos, estos grandes ficheros pueden ser de distinta naturaleza: logs de servidores web, logs de servicios UNIX, … Las investigaciones presentadas pueden ser trasladables a cualquier tipo de fichero log mediante la selección correcta de criterios.

En la actualidad, la automatización de patrones se basa en el análisis de datos a través de modelos o también conocido como data mining. Es importante darse cuenta de que se trata más de una metodología y un proceso continuo que una técnica o un conjunto de técnicas [6].

El primer modelo de detección de intrusos fue propuesto por Dorothy Denning, llamado NIDES (Intrusion Detection Expert System), aplicando algoritmos de desviaciones de uso [4]. Se puede considerar este incipiente sistema como el inicio del análisis de datos orientado a la detección de anomalías.

Las distintas estrategias de análisis que en la actualidad se están investigando [2] se dividen en dos grandes áreas, *Detección de Anomalías* y *Detección de Formas de Uso*

Indebidas. La primera categoría puede ser *No supervisada* o *Supervisada*. La detección de anomalías supervisada mediante data mining puede realizarse con las aplicaciones IDES, NIDES, EMERALD, SPACE, Computer Watch y Wisdom & sense. La segunda categoría mediante detección de formas de uso indebidas (Misuse detection) presenta mayor número de métodos: minería de datos (Shadow), Transición de estados (Network Flight Recorder,NetStat UCSB, NetRanger de Cisco), Sistemas expertos (P-Best SRI) y Patrones (Snort).

Desce hace años varias universidades de Norteamérica, principalmente a nivel de investigación han estado desarrollando algoritmos para la detección de intrusos aplicada a la informática, algunos de los más conocidos en la actualidad [5]:

- MINDS (Minnesota Intrusion Detection System) de la Universidad de Minnesota.
- MADAM ID (Mining Audit Data for Automated Models for Intrusion Detection) de la Universidad de Columbia, Georgia tech y Florida Tech.
- ADAM (Audit Data Analisis and Mining) de la Universidad de George Mason.
- IIDS (Intelligent Intrusion Detection) de la Universidad de Mississipi.
- Data Mining for Network Intrusion detection de la Corporación MITRE.
- Agent based data mining system de la Universidad de Iowa
- IDDM del departamento de defensa de Australia.

La linea de investigación que se está siguiendo es la clasificar los IDS según el paradigma de aprendizaje [4] que se utiliza, las actuales lineas son: reglas de asociación, inducción de reglas, árboles de decisión, clasificadores de Bayesianos, redes neuronales, support Vector Machines, modelos de Markov, algoritmos de genéticos, lógica Fuzzy (o lógica borrosa) [10].

Las *reglas de asociación* se basan en derivar correlaciones de varias características a partir de datos de ejemplo, ya que los datos obtenidos de la actividad de los usuarios muestran correlaciones. Wenke Lee y Salvarore Stolfo [18] presentan sus algoritmos de reglas de asociación para averiguar los eventos en una ventana temporal para calcular patrones a partir de una base de datos logs de la actividad del sistema. Utilizaron los archivos logs de la actividad generada de los servidores del MIT Lincoln Labs, con contenido de 7 semanas, el método detectó el 70% de los ataques.

La *inducción de reglas* es un método para inducir patrones secuenciales temporales en una secuencia de eventos. Lee y Stolfo [18] propusieron el uso de la herramienta RIPPER para la detección de intrusiones. RIPPER [4] es una herramienta que se utiliza para el aprendizaje de reglas sobre un conjunto de datos previamente etiquetados.

Árboles de decisión [6], Nong Ye de Arizona State University, realizó una propuesta de árboles de decisión como método de aprendizaje de patrones de ataques [17]. Hay estudios que hacen uso de árboles de decisión como método de optimización de IDS basados en patrones, construyendo una versión del algoritmo ID3 a partir de las reglas de Snort 2.0 y todo indica que los resultados en ella detección mejoran. [12]

Clasificadores Bayesianos [6], funcionan en forma de relaciones entre probabilidades condicionales en lugar de reglas. Las redes Bayesianas son modelos muy potentes. S.Axelsson fue el primero en investigar sobre la eficacia de este método en la detección de intrusos [7]. El sistema ADAM (Audit Data Analysis and Mining) [8] hace uso del método bayesiano, este sistema ideado por Barbará propone el uso estimaciones pseudo-Bayes para mejorar la detección de anomalías, el resultado fue que se reducía el número de fallos positivos. Lo interesante de esta investigación fue que la técnica era capaz de clasificar los registros sospechosos como normales, ataques conocidos y ataques nuevos.

Redes neuronales [6], posiblemente sea el área donde más se esté investigando gracias a la flexibilidad y adaptación a los cambios naturales que se pueden dar en el entorno y,

sobre todo, a la capacidad de detectar instancias de los ataques desconocidos. Aunque los resultados muestran la gran potencia en la detección de intrusos también evidencian una gran limitación que aún no ha sido solucionada [3], ya que se no es posible determinar la razón de la decisión tomada.

Iren Lorenzo y Francisco Maciá han publicado un artículo cuya propuesta se basa en un IDS basado en redes neuronales en la *Reducción de Características* [3], consiste en técnicas PCA y ANN que pretende resolver los problemas de los falsos positivos de los detectores de intrusos. La aplicación de PCA se dirige a dispar la relación disyuntiva entre la necesidad de capturar datos suficientes para la clasificación pero asegurando que sea el número mínimo para que el rendimiento del clasificador no se vea comprometido. El método de aplicación en el que se han basado es en paquetes TCP/IP, pero fácilmente trasladable a logs de servidores dedicados.

Support Vector Machine (SVM), es una técnica menos investigada que se utiliza en el aprendizaje supervisado. Mukkamala publicó una comparación entre la eficacia de los métodos SVM con el método basado en redes neuronales y concluyó que los métodos SVM mostraban mejores resultados. [15]

Otro interesante modelo en que se está investigando mediante *Cadenas de Markov*, estos modelos son un tipo de aprendizaje basado en secuencias. Una cadena de Markov es una secuencia de eventos, donde la probabilidad del resultado de un evento depende sólo del resultado del evento anterior. Los HMM son técnicas probabilísticas para el estudio de series en tiempo en función de las cadenas de Markov. Yi Hu presentó en 2004 [18] su sistema basado en dependencias para eventos como el modelo subyacente para la detección utilizando intrusos.

Uno de los métodos de investigación más recientes son los *Algoritmos Genéticos*, Ludovic Mé fue el primero en plantear este tipo de algoritmos para la detección de intrusos. Estos algoritmos son métodos sistemáticos para la resolución de problemas de búsqueda y optimización que aplican los métodos biológicos: selección basada en la población, reproducción sexual y mutación. La utilización de estos algoritmos para la detección de intrusos se ha llevado a cabo para mejorar la eficiencia en la selección de subconjuntos de características para reducir el número de características observadas mantiendo la precisión del aprendizaje. En 1998 se presenta el proyecto GASSATA (Genetic Algorithm as an Alternative Tool for Security Audit Trail Analysis) que utiliza un algoritmo genético para buscar la combinación de los ataques conocidos que mejor se correspondan con el registro auditado.

El útimo trabajo conocido con este método corresponde a la universidad de Mississipi, donde el algoritmo genético se inicia con un pequeño conjunto de reglas generadas aleatoriamente, y dichas reglas evolucionan hasta generar un conjunto de datos mayor que contiene reglas del IDS.

En 1994 se presentó un sistema basado en *Inmunes Artificiales*, un grupo de investigación liderado por Forrest [11] en Nuevo México están trabajando desde entonces en desarrollar un sistema inmune artificial basado en eventos de ordenadores. En 1996 presentaron su primer experimento con objetivo de detectar intrusiones en sistemas UNIX, la idea principal se basaba en recoger secuencias de comandos del servidor de email sendmail de UNIX y tras un periodo de entrenamiento decidir que eventos eran SPAM y cuales no.

T.Lane realizó un IDS basado en host para la detección de anomalías basándose en la técnica *Vecino más Cercano* (k-NN) [19], consiste en utilizar el aprendizaje basado en instancias, se consigue almacenando los ejemplos de entrenamiento y cuando se desea clasificar un nuevo evento, se extraen los objetos más parecidos y se usa su clasificación para catalogar el nuevo evento. Lo que principal diferencia a unos y otros algoritmos es la

medida utilizada para definir la similitud entre eventos. El experimento de Lane se basó en tomar como eventos las entradas de los comandos Shell de Unix con el fin de mapear datos temporales sobre el espacio, y basó la medida de similaridad en la regla de clasificación 1-NN. Esta técnica es interesante para ser aplicada en weblogs y tomar los resultados con una clasificación 1-NN.

Los métodos de *Clustering* [6] (agregación) son utilizados cuando existe un desconocimiento de los eventos capturados, se trata de obtener una descripción inicial que separe en grupos de objetos con características parecidas. Esta primera separación debe permitir reflexionar acerca de las características comunes de los objetos que pertenecen a cada grupo. Dentro este método de datamining los investigadores Portnoy, Eskin y Solfo [16] crearon la técnica K-means, que consiste en construir k particiones de los datos donde cada partición representa un cluster. Estos permite crear una partición inicial e iteran hasta satisfacer el criterio de parada.

Por último, cabe destacar el esfuerzo de Enrique López y Angela Díez de aplicar el modelo Borroso [10] para la detección de intrusos. La lógica borrosa o *fuzzy* es adecuada para la detección de intrusos por dos razones, por un lado están involucradas una gran cantidad de características cuantitativas, y por otro lado, la seguridad en sí misma incluye la confusión, es un hecho borroso. Dada una característica cuantitativa, se puede usar un intervalo para indicar un valor normal. La investigación de Enrique López y Angela Díez [10] se basa en la determinación del nivel de intrusión que se puede producir en un sistema informático a partir del análisis de una serie de variables utilizadas en la detección de intrusos, para ello utilizaron las siguientes variables de entrada: horario, directorios de sistema, uso de comandos peligrosos, tiempo CPU, comandos usados y tiempo de Uso I/O. Como variable de salida sugirieron el nivel de intrusión, previa calificación del mismo como informativo, sospechoso, serio o crítico.

Este mismo método ha sido utilizado para evitar los falsos positivos de un IDS. Hay que recordar que el principal problema de los IDS es el alto porcentage de falsos positivos, investigación realizada por la universidad Carlos III de Madrid.

Las investigaciones analizadas evidencian el potencial del data mining como método para detección de intrusos. Los servidores de sitios web no son la excepción de sufrir ataques y determinar los criterios de filtrado es de gran importancia. Distinguir entre un registro de actividad permitida de un registro sospechoso puede ser prácticamente imposible a simple vista, por lo que se requiere de algoritmos que analicen la actividad capturada en los ficheros logs. Por ello la importancia de las diferentes investigaciones presentadas en este documento son muy interesantes ya que resuelven el principal problema en la detección de intrusos en servidores web en busca de patrones sospechosos. El método borroso [10] comentado anteriormente abre las posibilidades de ser aplicado en los logs de servidores web por su sencillez y por su propiedad de previa calificación.

Para concluir, en cuanto al uso de data mining para la detección de ataques es importante seleccionar adecuadamente las características que intervienen en el proceso de aprendizaje, más que en el propio algoritmo que se vaya a utilizar, por lo que no se puede destacar un método más que otro. Es necesario disponer de una gran base de datos de eventos (del orden de varios megas, e incluso gigas) para un eficiente aprendizaje. Los weblogs de servidores de alto tráfico ofrecen gran cantidad de información por lo que se considera base suficiente para que se le aplique cualquier técnica de data mining.

En la actualidad no existen trabajos de investigación que propongan aspectos teóricos centrados en la utilización de técnicas de datamining sobre archivos weblogs. Como se ha expuesto en el estado del arte, la carencia de este es tipo de investigaciones justifica la necesidad de realizar el trabajo de investigación que se presentará en el artículo.

Sistema propuesto para la detección de intrusos WEB

Este trabajo propone a nivel teórico un sistema data-mining basado en lógica borrosa, representado gráficamente en la figura 1. Basado en la aplicación de lógica difusa que va encaminado a determinar el nivel de intrusión. Con este método se busca eficiencia y escalabilidad en la aplicación Fuzzy para la detección de intrusos [20].

Por definición, un sistema de inferencia borrosa (FIS, Fuzzy Inference System en inglés) es una forma de transformar un espacio de entrada en un espacio de salida utilizando lógica borrosa [21]. Los FIS tratan de formalizar, mediante lógica borrosa (construyendo reglas IF-THEN borrosas) razonamientos del lenguaje humano, como por ejemplo:

"Si la visita procede de un país no objetivo, aunque no esté utilizando comandos sospechosos, filtraré el acceso"

Se utilizan para resolver un problema de decisión, esto es, tomar una decisión y actuar en consecuencia.

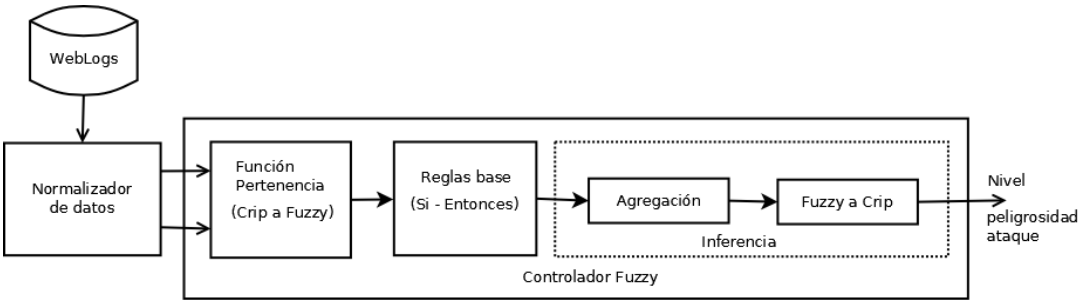

Figura1. Sistema Fuzzy de detección de intrusos

El método inicialmente propone el análisis de los registros de actividad web con el objetivo de identificar los datos útiles y generar las variables derivadas. Los datos que responden al patrón son pasados al controlador Fuzzy, la primera tarea es la conversión del valor real a su correspondiente grado de pertenencia [22]. Una vez realizada la conversión de dominio se les aplica las reglas establecidas por un experto, reglas lógicas y sencillas de patrón IF – THEN. El sistema Fuzzy tiene varias reglas y por lo tanto puede considerar varias entradas y varias salidas.

El bloque "*Inferencia*" establece la contribución de conjunto de reglas para lograr la salida deseada. La contribución está determinada por el conocimiento de un experto, sobre los valores de las variables de entrada, de modo que se consideran reglas de detección de ataques, por ejemplo, DoS y Web Scripting. El módulo "*Agregación*" determina el significado de todas las reglas Fuzzy en base al uso de operadores Fuzzy. Por último el módulo "*Defuzzification*" extrae el valor generado por el sistema y lo convierte al dominio real, es decir, una salida entendible por el ser humano y con grado de pertenencia de 100%.

Justificación de uso de Sistemas de Inferencia Borrosa

Son varios lo motivos que justifican el diseño de un sistema borroso en el análisis de web-logs para la detección de intrusos, principalmente [22]:

- La lógica borrosa no resuelve problemas nuevos, sino que utiliza nuevos métodos para resolver los problemas de siempre.
- Los conceptos matemáticos dentro del razonamiento borroso son muy simples.
- La lógica borrosa es flexible: es fácil transformar un FIS añadiendo o eliminando reglas sin tener que empezar desde cero.

- La lógica borrosa admite datos imprecisos (NO estudia la incertidumbre): maneja elementos de un conjunto borroso, es decir, valores de una función de pertenencia. Por ejemplo, en lugar de manejar el dato "Se ha producido un ataque", maneja "Ataque crítico con grado de 0.8".
- La lógica borrosa se construye sobre la experiencia de los expertos: confía en la experiencia de quien ya conoce el sistema.
- La lógica borrosa puede mezclarse con otras técnicas clásicas de control.

Elección del algoritmo borroso

Los métodos de inferencia borrosa se clasifican en *métodos directos* y *métodos indirectos*. Los directos son los más utilizados, como por ejemplo los de Mamdani y Sugeno (estos dos métodos se diferencian en la forma de obtener las salidas). Los métodos indirectos son más complejos. Para la detección de intrusos se ha elegido el método de Mamdani [24], por su sencillez de diseño.

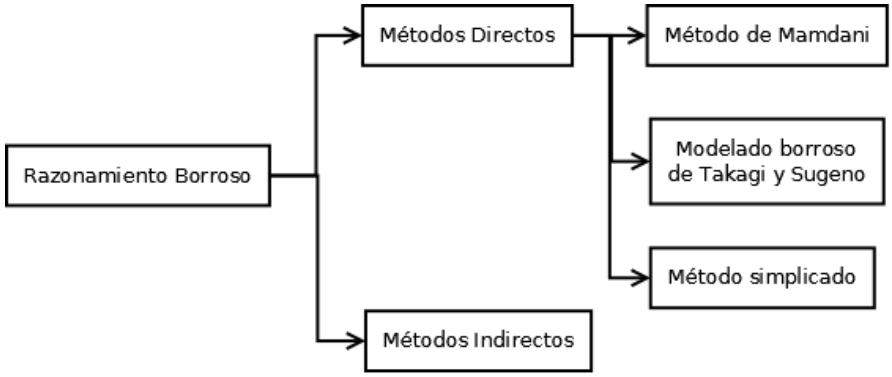

Figura2. Métodos de inferencia borrosa

Preprocesado de datos web-logs

El sistema borroso tiene varias variables de entrada obtenidas de los web-logs, el diagrama 1 establece un bloque de *preprocesado* de datos que consiste en obtener las variables de los logs. Los logs son grandes volúmenes de datos que llegan a ocupar desde megas a gigas, registrando la actividad web de los usuarios. El sistema propuesto pretende identificar la actividad sospechosa y para ello será necesario preparar los datos para ser introducidos en el sistema borroso.

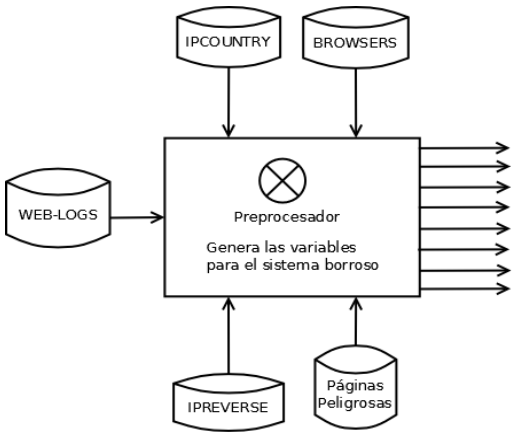

Figura 3. Preprocesado de datos

El diagrama de bloques muestra que las variables de salida se obtienen cruzando varias bases de datos en función de los web-logs, las bases de datos son:

- *Ip Country*: Para averiguar la procedencia geográfica de la visita
- *IP Reverse*: Para conocer si una IP tiene un dominio alojado
- *Browsers*: Huellas de los principales navegadores de Internet (msi, firefox, …)
- *Páginas peligrosas*: Cadenas sospechosas en las solicitudes (phpadmin , admin,, …)

Determinación de las variables

El sistema propuesto está encaminado a la detección del nivel de ataque que se produce en un servidor web a partir del análisis de una serie de variables de los web-logs obtenidas del módulo *preprocesado.* Para ello, se utilizan como variables de entrada al sistema borroso las siguientes:

- *País*: indica si la intrusión se produce desde la zona geográfica permitida o no. Será calificado como país objetivo o no objetivo. Por ejemplo, una tienda online buscará clientes de la eurozona, cualquier otra visita puede considerarse como sospechosa.
- *IP reverse*: indica si la IP de la visita tiene asociada algún dominio alojado. Será calificado como servidor web o no web. Por ejemplo, es sospechoso que un usuario esté navegando desde un servidor web con dominios alojados.
- *Tipo conexión*: indica el tipo de conexión del usuario al servidor web. Por ejemplo, un tipo de conexión diferente a POST o GET puede considerarse sospechoso ya que los navegadores de Internet utilizan uno de estos dos métodos. Será calificado como válido o no válido.
- *Navegador usado*: indica si el navegador utilizado por la visita es conocido. También incluye los user-agents de los spiders de los principales buscadores. Será calificado como conocido o no conocido.
- *Páginas vistas sin referido*: indica el número de accesos directos. Será calificado como bajo, medio o alto.
- *Páginas vistas por minuto*: indica el número de accesos por minuto. Sirve para detectar ataques de denegación de servicio. Será calificado como bajo, medio o alto.
- *Solicitud de páginas peligrosas*: indica si el intruso solicita paginas consideradas como peligrosas (ej: admin, phpmyadmin, …). Será calificado como solicita (dichos recursos) o no solicita (recursos).
- *Páginas peligrosas*: número de páginas diferentes peligrosas. Será calificado como bajo, medio o alto.

Las variables *país, IpReverse, tipo conexión, navegador usado y páginas solicitadas* las podemos considerar como variables binarias ya que solo miden si la variable se utiliza o no. El resto de variables se pueden considerar ordinales ya que miden algunos comportamientos cuantificables numéricamente.

Como variable de salida se utilizará el nivel de ataque, previa calificación del mismo como informativo, sospechoso, serio o crítico.

Determinación de la relación entre las variables

Para el correcto funcionamiento de sistema, la primera tarea es determinar las relaciones entre las variables del modelo, es decir, establecer el aprendizaje relacional que permita dar una salida en función de los diferentes valores de las entradas. Se puede observar las relaciones en la siguiente figura.

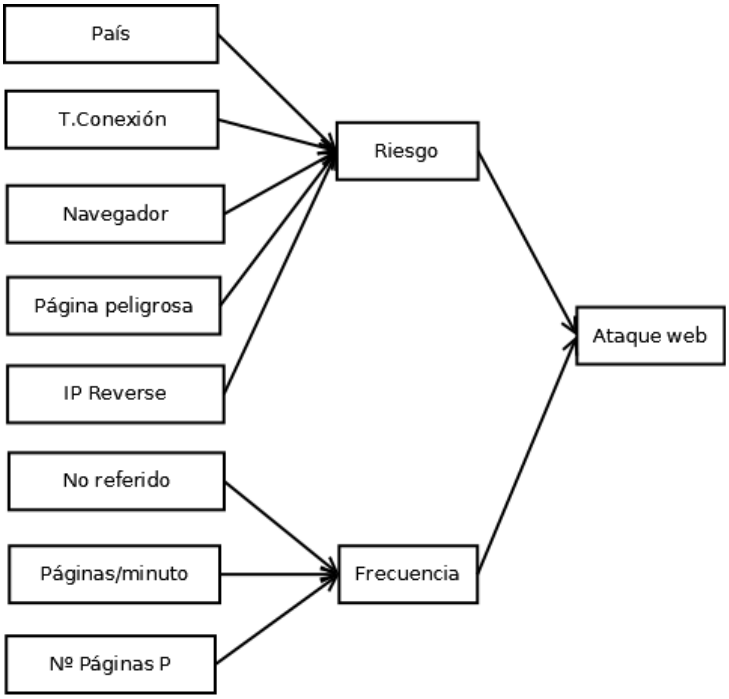

Figura 4. Esquema relacional entre variables

Del diagrama anterior se puede observar las siguientes reglas asociativas:

- De las variables superiores (*país, tipo de conexión, navegador, página peligrosa e IP Reverse*) se agrupan para dar valor a una nueva variable deducida llamada riesgo que determinará el riesgo que supone para el sistema la posible intrusión. Esta variable se califica como muy bajo, bajo, medio, alto o muy alto.
- De las variables inferiores (*No referido, páginas/minuto y nº páginas sospechosas*) se agrupan en otra variable deducida llamada frecuencia que determinará la periocidad con que se produce la posible actividad sospechosa. Esta variable se califica como muy baja, baja, media, alta y muy alta.
- Para determinar el nivel de ataque se tendrán en cuenta estas dos variables intermedias.

Fuzzificación de las variables

Una vez decididas las variables a utilizar bajo criterios de expertos y las relaciones existentes entre las mismas, se procede a determinar el dominio de cada uno de los números borrosos en las que se divide cada variable. Nos centraremos en determinar el dominio de las variables multidimensionales de *No referido, Páginas minuto y Número de páginas solicitadas sospechosas*, ya que es evidente que el resto de variables solo pueden tener dos valores 1 ó 0, siendo variables binarias.

Variable No Referido

Se han diferenciado dentro de esta variable los siguientes estados: *densidad baja, densidad media y densidad alta*. Las funciones de transferencia según los dominios asignados a la variable son:

$$\mu_{NRF.BAJO}(x)=\begin{cases} 0 & si \quad (x\le 0)\,\acute{o}\,(x\ge 15) \\ \frac{x}{5} & si \quad 0<x\le 5 \\ 1 & si \quad 5<x<10 \\ \frac{15-x}{5} & si \quad 10\le x<15 \end{cases}$$

$$\mu_{NRF.MEDIO}(x)=\begin{cases} 0 & si \quad (x\le 10)\,\acute{o}\,(x\ge 50) \\ \frac{x-10}{15} & si \quad 10<x\le 25 \\ 1 & si \quad 25<x<30 \\ \frac{50-x}{20} & si \quad 30\le x<50 \end{cases}$$

$$\mu_{NRF.ALTO}(x)=\begin{cases} 0 & si \quad (x\le 40)\,\acute{o}\,(x\ge 100) \\ \frac{x-40}{20} & si \quad 40<x\le 60 \\ 1 & si \quad 60<x<100 \end{cases}$$

- % No referido Bajo = (0,5,10,15)
- % No referido Medio = (10,25,30,50)
- % No referido Alto = (40,60,100,100)

	Bajo	**Medio**	**Alto**
No menor que	0	10	40
Igual que	5	25	60
Igual que	10	30	100
No mayor que	15	50	100

Variable Páginas/segundo

Se han diferenciado dentro de esta variable los siguientes estados: bajo, medio y alto. Al igual que la variable anterior, se determinan los rangos borrosos representativos de tales estados, cuyos valores se representan en la siguienta tabla 3. Las funciones de transferencia según los dominios asignados a la variable son:

$$\mu_{PM.BAJO}(x)=\begin{cases} 0 & si \quad (x<0)\,\acute{o}\,(x>3) \\ 1 & si \quad 0\le x\le 1 \\ \frac{3-x}{2} & si \quad 1\le x\le 3 \end{cases}$$

$$\mu_{PM.MEDIO}(x)=\begin{cases} 0 & si \quad (x<2)\,ó\,(x>6) \\ x-2 & si \quad 2\le x\le 3 \\ 1 & si \quad 3\le x\le 4 \\ \dfrac{6-x}{2} & si \quad 4\le x\le 5 \end{cases}$$

$$\mu_{PM.ALTO}(x)=\begin{cases} 0 & si \quad (x<5)\,ó\,(x>10) \\ \dfrac{x-5}{2} & si \quad 5\le x\le 7 \\ 1 & si \quad 7\le x\le 10 \end{cases}$$

- Páginas minuto bajo = (0,0,1,3)
- Páginas minuto medio = (2,3,4,5)
- Páginas minuto alto = (5,7,10,10)

	Bajo	**Medio**	**Alto**
No menor que	0	2	5
Igual que	0	3	7
Igual que	1	4	10
No mayor que	3	5	10

Variable Páginas peligrosas

La última variable de frecuencia como entrada es el número de solicitudes de páginas peligrosas. Del mismo modo que las anteriores variables los estados posibles son: bajo, medio y alto. Los números borrosos asociados se presentan en la tabla 4. Las funciones de transferencia según los dominios asignados a la variable son:

$$\mu_{PP.BAJO}(x)=\begin{cases} 0 & si \quad (x<0)\,ó\,(x>3) \\ 1 & si \quad 0\le x\le 1 \\ \dfrac{3-x}{2} & si \quad 1\le x\le 3 \end{cases}$$

$$\mu_{PP.MEDIO}(x)=\begin{cases} 0 & si \quad (x<2)\,ó\,(x>6) \\ x-2 & si \quad 2\le x\le 3 \\ 1 & si \quad 3\le x\le 4 \\ \dfrac{6-x}{2} & si \quad 4\le x\le 5 \end{cases}$$

$$\mu_{PP.ALTO}(x)=\begin{cases} 0 & si \quad (x<5)\,ó\,(x>10) \\ \dfrac{x-5}{2} & si \quad 4\le x\le 6 \\ 1 & si \quad 7\le x\le 10 \end{cases}$$

- Páginas peligrosas bajo = (0,0,1,3)
- Páginas peligrosas medio = (2,3,4,5)
- Páginas peligrosas alto = (5,7,10,10)

	Bajo	**Medio**	**Alto**
No menor que	0	2	5
Igual que	0	3	7
Igual que	1	4	10
No mayor que	3	5	10

Variable AtaqueWeb

La variable de salida determinará el grado de ataque en función de las variables de entrada anteriores y aplicando las reglas establecidas sobre estas. La variable de salida es calificada como: informativo, sospechoso, serio y crítico. Los números borrosos asociados se presentan en la tabla 5. Las funciones de transferencia según los dominios asignados a la variable son:

$$\mu_{INFORMATIVO}(x)=\begin{cases} 0 & si \quad (x<0)\,ó\,(x>15) \\ \frac{x}{5} & si \quad 0\le x\le 5 \\ 1 & si \quad 5\le x\le 10 \\ \frac{15-x}{5} & si \quad 10\le x\le 15 \end{cases}$$

$$\mu_{SOSPECHOSO}(x)=\begin{cases} 0 & si \quad (x<12)\,ó\,(x>40) \\ \frac{x-12}{8} & si \quad 12\le x\le 20 \\ 1 & si \quad 20\le x\le 30 \\ \frac{40-x}{10} & si \quad 30\le x\le 40 \end{cases}$$

$$\mu_{SERIO}(x)=\begin{cases} 0 & si \quad (x<35)\,ó\,(x>60) \\ \frac{x-35}{10} & si \quad 35\le x\le 45 \\ 1 & si \quad 45\le x\le 50 \\ \frac{60-x}{10} & si \quad 50\le x\le 60 \end{cases}$$

	Inform.	**Sosp.**	**Serio**	**Crítico**
No menor que	0	12	35	55
Igual que	5	20	45	65
Igual que	10	30	50	100
No mayor que	15	40	60	100

Establecimiento de las reglas

Se basa en reglas heurísticas de la forma *SI (antecedente) ENTONCES (consecuente),* donde el antecedente y el consecuente son también conjuntos difusos. La entrada de la regla es un valor numérico (*conjunto crisp*) que se le da a la variable de entrada del antecedente. La salida de la regla es un conjunto borroso que se asigna a la variable de salida y del consecuente. La ejecución de la regla se realizará aplicando un operador borroso de implicación cuyos argumentos son el valor del antecedente y los valores del conjunto borroso del consecuente. La salida de cada regla será el conjunto borroso resultante de la implicación.

Variable derivada "Riesgo"

Es una variable derivada de 5 variables binarias (país, tipo conexión, navegador usado, página peligrosa e IP Reverse), las reglas de funcionamiento para la formación de la veriable derivada se muestran en la tabla 6:

País	Tipo Conexión	Navegador	Página P.	IP Reverse	Riesgo
Objetivo	Válido	Conocido	Peligrosa	Servidor web	Medio
Objetivo	Válido	Conocido	Peligrosa	No servidor	Bajo
Objetivo	Válido	Conocido	No peligrosa	Servidor web	Bajo
Objetivo	Válido	Conocido	No peligrosa	No servidor	Muy bajo
Objetivo	Válido	No conocido	Peligrosa	Servidor web	Alto
Objetivo	Válido	No conocido	Peligrosa	No servidor	Medio
Objetivo	Válido	No conocido	No peligrosa	Servidor web	Medio
Objetivo	Válido	No conocido	No peligrosa	No servidor	Bajo
Objetivo	No válido	Conocido	Peligrosa	Servidor web	Alto
Objetivo	No válido	Conocido	Peligrosa	No servidor	Medio
Objetivo	No válido	Conocido	No peligrosa	Servidor web	Medio
Objetivo	No válido	Conocido	No peligrosa	No servidor	Bajo
Objetivo	No válido	No conocido	Peligrosa	Servidor web	Alto
Objetivo	No válido	No conocido	Peligrosa	No servidor	Alto
Objetivo	No válido	No conocido	No peligrosa	Servidor web	Alto
Objetivo	No válido	No conocido	No peligrosa	No servidor	Medio
No objetivo	Válido	Conocido	Peligrosa	Servidor web	Alto
No objetivo	Válido	Conocido	Peligrosa	No servidor	Medio
No objetivo	Válido	Conocido	No peligrosa	Servidor web	Alto
No objetivo	Válido	Conocido	No peligrosa	No servidor	Bajo
No objetivo	Válido	No conocido	Peligrosa	Servidor	Alto

				web	
No objetivo	Válido	No conocido	Peligrosa	No servidor	Alto
No objetivo	Válido	No conocido	No peligrosa	Servidor web	Alto
No objetivo	Válido	No conocido	No peligrosa	No servidor	Medio
No objetivo	No válido	Conocido	Peligrosa	Servidor web	Muy alto
No objetivo	No válido	Conocido	Peligrosa	No servidor	Alto
No objetivo	No válido	Conocido	No peligrosa	Servidor web	Alto
No objetivo	No válido	Conocido	No peligrosa	No servidor	Medio
No objetivo	No válido	No conocido	Peligrosa	Servidor web	Muy Alto
No Objetivo	No válido	No conocido	Peligrosa	No servidor	Muy Alto
No Objetivo	No válido	No conocido	No peligrosa	Servidor web	Muy Alto
No Objetivo	No válido	No conocido	No peligrosa	No servidor	Alto

Variable derivada "Frecuencia"

De forma similar a la variable Riesgo, esta variable es derivada de Nº referidos, Página/minutos y nº de páginas sospechosas. Donde las reglas de funcionamiento para formación se recogen en la tabla 7.

No referido	**Páginas/minuto**	**Nº páginas P**	**Frecuencia**
Bajo	Bajo	Bajo	Muy bajo
Bajo	Bajo	Medio	Bajo
Bajo	Bajo	Alto	Medio
Bajo	Medio	Bajo	Bajo
Bajo	Medio	Medio	Medio
Bajo	Medio	Alto	Medio
Bajo	Alto	Bajo	Bajo
Bajo	Alto	Medio	Medio
Bajo	Alto	Alto	Medio
Medio	Bajo	Bajo	Bajo
Medio	Bajo	Medio	Medio
Medio	Bajo	Alto	Medio
Medio	Medio	Bajo	Medio
Medio	Medio	Medio	Medio
Medio	Medio	Alto	Medio
Medio	Alto	Bajo	Medio
Medio	Alto	Medio	Medio
Medio	Alto	Alto	Alto
Alto	Bajo	Bajo	Bajo

Alto	Bajo	Medio	Medio
Alto	Bajo	Alto	Medio
Alto	Medio	Bajo	Medio
Alto	Medio	Medio	Medio
Alto	Medio	Alto	Alto
Alto	Alto	Bajo	Alto
Alto	Alto	Medio	Alto
Alto	Alto	Alto	Muy alto

Variable derivada "Ataque web"

Por último, la salida es el resultado de combinar las dos variables intermedias (riesgo y frecuencia), del mismo se define las reglas operativas para cuantificar el grado de ataque web, tabla 8

Riesgo	**Frecuencia**	**Grado de ataque web**
Muy Bajo	Muy Baja	Informativo
Muy Bajo	Baja	Informativo
Muy Bajo	Media	Informativo
Muy Bajo	Alta	Sospechoso
Muy Bajo	Muy Alta	Sospechoso
Bajo	Muy Baja	Informativo
Bajo	Baja	Informativo
Bajo	Media	Sospechoso
Bajo	Alta	Sospechoso
Bajo	Muy Alta	Sospechoso
Medio	Muy Baja	Sospechoso
Medio	Baja	Sospechoso
Medio	Media	Serio
Medio	Alta	Serio
Medio	Muy Alta	Serio
Alto	Muy Baja	Serio
Alto	Baja	Serio
Alto	Media	Serio
Alto	Alta	Crítico
Alto	Muy Alta	Crítico
Muy alto	Muy Baja	Crítico
Muy alto	Baja	Crítico
Muy alto	Media	Crítico
Muy alto	Alta	Crítico
Muy alto	Muy Alta	Crítico

Funcionamiento del sistema

Variables de entrada

Los datos de entrada se extraen de los logs generados por el servidor web y son introducidos en el sistema borroso. Con los datos de entrada ("*crisp*") se deberá determinar el grado de verdad de cada una de las etiquetas de las diferentes variables de entrada país, tipo conexión, navegador, página peligrosa, ip reverse, no referido, páginas minuto y nº páginas peligrosas. Por ejemplo, el grado de activación para un "*crisp*" de valor 55 de la variable Nº de páginas sospechosas, sería:

$$\mu_{SERIO}(x) = \begin{cases} \frac{60-x}{10} & si \quad 50 \le x \le 60 \end{cases}$$

$$\mu_{SERIO}(x) = \begin{cases} \frac{60-55}{10} = 0.5 \end{cases}$$

El nivel de peligrosidad sería Serio con un grado de activación de 0.5.

Lógicamente, se realiza lo mismo con el resto de variables.

Variables intermedias "Riesgo" y "Frecuencia"

Para determinar el grado de pertenencia de estas variables, es necesario conocer el grado de verdad de la regla utilizada.

Al ser reglas derivadas, el grado de verdad determinará a partir del grado de verdad de cada uno de los antecedentes. Relacionados a través de una T-norma, las funciones de transferencia serán:

$$\mu_r(RIESGO) = \mu_x(pais) * \mu_y(T.Conexión) * \mu_z(Navegador) * \mu_s(P.Peligrosas)$$

$$\mu_r(RIESGO) = \mu_r(RIESGO) * \mu_v(IPReverse)$$

$$\mu_f(FRECUENCIA) = \mu_x(Nº\ Referidos) * \mu_y(Páginas/minuto)$$

$$\mu_f(FRECUENCIA) = \mu_f(FRECUENCIA) * \mu_z(NºPáginas\ Peligrosas)$$

Para determinar el grado de pertenencia total al subconjunto de las salidas lingüísticas se ha elegido la opción propuesta por Mamdani; es útil cuando el número de variables es reducido y muy simple a nivel computacional.

Variable final "Ataque Web"

Se obtiene de la misma manera que las variables intermedias, para determinar el grado de pertenencia de la variable de salida "Ataque Web" será necesario el grado averiguar el grado de verdad de la regla utilizada.

En este caso la T-norma que se utilizará será:

$$\mu_r(ATAQUE) = \mu_x(RIESGO) * \mu_y(FRECUENCIA)$$

y por último se aplicaría el método de Mamdani para determinar el grado de pertenencia

total al subconjunto debido a que existen diferentes reglas que dan lugar al mismo consecuente.

Desborrosificador

El último paso sería determinar el grado de pertenencia borroso de cada subconjunto de la variable final (informativo, sospechoso, serio y crítico), es decir, la cantidad "*crisp*" (no borrosa) a un valor numérico. Este último paso, los que nos dirá es el grado de importancia del ataque. Por ejemplo, el ataque puede "sospechoso" con una gravedad del 0.78%.

Para ello se procede a la defuzzificación de los valores de la variable Ataque. Uno de los métodos más utilizados es el del centroide [23], que calcula el centro del área definida por el conjunto borroso obtenido.

$$Centroide = \frac{\sum_{i=1}^{N} (x_i * u(x_i))}{\sum_{i=1}^{N} (u(x_i))}$$

Conclusiones

En este trabajo se ha propuesto la base teórica de un método para la detección de intrusos mediante lógica borrosa aplicada a weblogs, que se basa en reglas preestablecidas por un sistema experto. Se ha obtenido un modelo sencillo que detecta ataques webs y establece un valor de gravedad. En cuanto al uso de data mining para la detección de ataques weblogs, hay que destacar que muchas veces es más importante seleccionar adecuadamente las características que intervienen en las reglas de clasificación, que en el propio algoritmo o técnica que se vaya a utilizar. El sistema borroso es una propuesta acertada, ya que la generación de reglas puede ser establecida por la experiencia humana al conocerse los diferentes tipos de ataques y sus consecuencias. Ante diferentes ataques simplemente hay que añadir nuevas reglas que definan la nueva anomalía.

Este artículo se puede considerar como la base de aplicación de lógica difusa en la detección weblogs, se trata de una línea de investigación muy interesante ya que permite diferencias líneas de actuación futuras. La lógica borrosa permite que las reglas puedan ser más precisas mediante el autoaprendizaje, haciendo uso de redes neuronales para fortalecer las futuras tomas de decisión. Del mismo modo, puede ser utilizado para identificar el tipo de ataque (DoS, DdoS, SQL scripting, ...)

Referencias

1. ALEX Nanopoulus, Yannis Manolopoulos. "Finding Generalized Path Patterns for Web Log Data Mining". *Department of Informatics, Aristotle University*, 2000
2. CASTILLO, santos. "Detección de Intrusos mediante técnicas de mineria de datos". *Revista Clepsidra. Nº 2* (Enero 2006) p. 31-44
3. Iren Lorenzo Fonseca, Francisco Maciá Pérez, Rogelio Lau Fernández, Fco. José Mora Gimeno, Juan Antonio Gil Martínez-Abarca. "Método para la Detección de Intrusos mediante Redes Neuronales basado en la Reducción de Características". *Departamento de Tecnología Informática y Computación. Universidad de Alicante*. 2008.

4. Stefan Axelsson. “Intrusion Detection Systems: A Survey and Taxonomy”. *Chalmers University of Technology*. 14 March 2000.

5. MENA, Jesus. “Investigative data mining for security and criminal detection”. *Butterworth-Heinemann*, 2003. 272 p. ISBN: 0-7506-7613-2

6. MOR Enric, SANGÜESA Ramón, MOLINA Luis Carlos. “Data Mining”. *Fundació per a la Universitat Oberta de Catalunya*, 2004. ISBN: 84-9707-623-0

7. Axelsson S. “The Base-rate Fallacy and its Implications for the Difficulty of Intrusion Detection”. 6 *ACM Conference on Computer and Commincations* Security, 1999.

8. Barbará, D., Wu, N. “Detecting Novel Network Intrusions Using Bayes Estimators”. *First SIAM Conference on Data Mining*, 2001.

9. D.Brent Chapman. “Network (In) Security Through IP Packet Filtering”. *Proceedings of the Third UNIX Security Symposium*, 1992.

10. Enrique López González, Angela Díez Díez, Francisco J. Rodríguez Sedano, Cristina Medaña Cuervo. “Diseño de un sistema borroso para la detección de intrusos”. *Cruzando fronteras: Tendencias de Contabilidad Directiva para el siglo XXI*, Julio 2001. León.

11. Forrest, S., Hofmeyr, S., Somayaji, A. “A Sense of Self for Unix Processes”. *IEEE Symposium on Computer Security and Privacy*, 1996.

12. Kruegel, C., Toth, t., Kirda, E. “Service Specific Anomaly Detection for Network Intrusion Detection”. *Symposium on Applied Computing (SAC)*, 2002.

13. L.K. Joshila Grace, V.Maheswari, Dhinaharan Nagamalai. “Analysis of Web logs and web user in Web Mining” *IJCSNS International Journal of Network Security & Its Application*. 2011.

14. K. R. Suneetha, Dr. R. Krishnamoorthi. “Identifying User Behavior by Analyzing Web Server Access Log File” *IJCSNS International Journal of Computer Science and Network Security, Vishveshwaraya Technology University*. 2009.

15. Mukkamala S., Janoski G. “Intrusion Detection Using Neural Networks and Support Vector Machines”. *IEEE International Joint Conference on Neural Networks*, 2002.

16. Portnoy, L. Eskin, E, Stolfo, S. “Intrusion Detection with Unlabeled Data Mining Clustering”. *ACM Workshop on Data Mining Applied to Security*, 2001.

17. Ye, N., Li, X, Emram, S.M. “Decision Three for Signature Recognition and State Classification”. *IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop*, 2000.

18. Yi Hu, Brajena Panda. “A Data Mining Aproach for Database Intrusion Detection”. *ACM Symposium on Applied Computing*, March 14-17, 2004.

19. Wenke Lee. “A Data Mining Framework for Building Intrusion Detection Models”. *Computer Science Department, Columbia University*. 1999.

20. Zadeh, L.A. “Fuzzy Logic=Computing with Words, IEEE Tranactions on Fuzzy Systems”. *Vol.4, págs. 103-11* 1996.

21. Luo, J. “Integrating Fuzzy Logic with Data Mining Methods for Intrusion Detection”, *A Thesis Submitted to the Faculty of Mississipi State University in Partial Fulfillment, Mississippi*, August, 1999.

22. Zimmermann, H.J. “Fuzzy set theory”, *Kluwer Academic Publishers, Mississippi*, 2001.

23. Dae-Won Kim, Kwang H.Lee, Doheon Lee. “Fuzzy clustering of categorical data

using fuzzy centroids". *Pattern Recognition Letters*, 2004.

24. Emami, M.R, Rirksen, I.B, Goldenberg A.A. "Development of a systematic methodology of fuzzy logic modeling. Fuzzy System", *IEEE Transactions, págs 346-361* Aug, 1998.