SIDI v. 1.: Una propuesta de Sistema inteligente para la Detección de Intrusos [1]

Enrique López González (*), Jesús Calabozo Morán (**), Cristina Mendaña Cuervo (*), Angela Díez Díez (**) y Francisco J. Rodríguez Sedano (**)

(*) Departamento de Dirección y Economía de la Empresa
(**) Departamento de Ingeniería Eléctrica y Electrónica
Universidad de León,
Campus de Vegazana, s/n. 24071. León (España).
e-mail para correspondencia: ddeelg@unileon.es

Resumen. Con el incremento en el empleo de los ordenadores y la emergencia del comercio electrónico, la detección de intrusos se ha convertido en una prioridad importante, pues no resulta técnicamente factible construir un sistema totalmente invulnerable. La hipótesis de la presente investigación es que la Lógica Borrosa es capaz de producir "mejores" reglas que incrementen la flexibilidad y robusted de los sistemas de auditoria informática. De hecho, los Sistemas Inteligentes (SI) han demostrado ser una herramienta eficaz en problemas de control, clasificación, modelado etc., ante contextos donde la información y/o los datos están afectados de imprecisión no probabilística. De esta forma, el principal propósito de esta contribución consistirá en elaborar un modelo de Sistema Inteligente aplicado a la Detección de Intrusos (SIDI), para lo cual se precisa profundizar en los dos aspectos siguientes: el diseño de un SI y su adaptación en la extracción de conocimiento relativo a la detección de intrusos.

1 Introducción

Nuestra sociedad depende cada vez más del acceso y procesamiento rápido de información, lo que ha supuesto la proliferación del empleo de los ordenadores y de las redes de comunicaciones y con ello la existencia de problemas de acceso desautorizado y manipulación de datos: la cantidad de intentos de accesos no autorizados a la información que existe en Internet ha crecido durante estos últimos años.

Según el Computer Security Institute, más de un 70% de las organizaciones anunciaron al menos un incidente de seguridad durante el último año, frente a un 42% anunciado en 1996. La mayoría de los expertos considera que estos números están muy por debajo de la tasa real, ya que muchas organizaciones evitan dar a conocer sus incidentes y muchas otras ni siquiera los detectan. Además, los intrusos se han vuelto

[1] Este trabajo está soportado por el proyecto de investigación DPI 2001-0105 del MCT.

expertos en determinar las debilidades, empleando diversos niveles de engaño antes de irrumpir en un sistema determinado, intentando cubrir sus huellas para que su actividad en el sistema no se descubra fácilmente. De ahí que, la detección de intrusos se haya convertido en una prioridad importante, pues no resulta técnicamente factible construir un sistema totalmente invulnerable, ya que el propio concepto de seguridad es en si mismo “borroso”.

La hipótesis principal que orienta nuestro trabajo consiste en que las denominadas “técnicas inteligentes” son capaces de producir "mejores" reglas que incrementen la flexibilidad y robusted de los sistemas de auditoría informática. De hecho, los Sistemas Inteligentes (SI) han demostrado ser una herramienta efectiva en problemas de control, clasificación, modelado, etc., en aquellos contextos donde la información y/o los datos están afectados de imprecisión no probabilística. El sustrato teórico de manejo de estos sistemas es la Lógica Borrosa, cuyas reglas de inferencia permiten construir algoritmos eficientes para la gestión de los SI [Bardossy, Duckstein, 1995].

El diseño de un SI contempla dos aspectos principales. Por un lado, y principal centro de atención del presente trabajo, la selección del modelo de inferencia en un SI es un aspecto que ha sido ampliamente estudiado tanto para problemas de modelado y control como para problemas de clasificación. A este respecto, en [Cordón, del Jesús, Herrera y López, 1997] se puede encontrar un estudio de las diferentes propuestas existentes en la literatura especializada, así como nuevas propuestas que permiten inferir a partir de la base de reglas de forma cooperativa. Por otro lado, la construcción de la base de reglas Borrosas, donde inicialmente se carecía de procedimientos sistemáticos generales para obtener una base de reglas lo que hizo que muchas de las aplicaciones que se desarrollaron siguiesen un enfoque de ensayo-error, mientras que en la actualidad, se estudian diferentes técnicas para el aprendizaje de bases de reglas Borrosas, tanto a partir de ejemplos como a partir del modelo del sistema.

Con este trabajo se trata de propugnar un diseño de sistema de detección de intrusos con un sistema experto embebido el cual interprete información de forma adecuada, filtrando el exceso de datos que pueden hacer a un administrador disminuir la atención que debe de prestar al sistema, para lo cual en el siguiente apartado se efectuará una descripción del tópico de estudio y a continuación presentar el modelo propugnado, analizando las fases de su diseño e implementación práctica, concluyendo con la presentación del interface desarrollado.

2 Detección de intrusos: Descubrimiento de amenazas en auditoria informática

La mayoría de sistemas de computación proporcionan un mecanismo de control de acceso como su primera línea de defensa. Sin embargo, esto sólo limita si el acceso a un objeto en el sistema se permite, pero no restringe lo que un sujeto puede hacer con el propio objeto si tiene el acceso para manipularlo. A mayor abundamiento, en sistemas dónde el control de acceso es discrecional, la responsabilidad de la protección de los datos recae sobre el usuario final. Esto requiere a menudo que los

usuarios entiendan el mecanismo de protección ofrecido por el sistema y cómo lograr la seguridad deseada usando estos mecanismos.

La cantidad de mensajes publicados en listas de vulnerabilidades como BUGTRAQ ha aumentado de forma exagerada durante los últimos años. Las vulnerabilidades no solo afectan a sistemas tradicionalmente seguros, sino que afectan incluso a sistemas de seguridad: cortafuegos. Por otra parte, aunque muchos escaneos de red y técnicas de ataques son conocidos desde hace varias décadas, no ha sido hasta hace poco tiempo que las herramientas para producir análisis sofisticados a redes han llegado a estar disponibles en el ámbito comercial, estando la mayoría de estas basadas en algún tipo de Sistema de Detección de Intrusos (SDI), entendido como una herramienta de seguridad encargada de monitorizar los eventos que ocurren en un sistema informático en busca de intentos de intrusión, esto es, cualquier intento de comprometer la confidencialidad, integridad, disponibilidad o evitar los mecanismos de seguridad de una computadora o red.

Las intrusiones se pueden producir de varias formas: atacantes que acceden a los sistemas desde Internet, usuarios autorizados del sistema que intentan ganar privilegios adicionales para los cuales no están autorizados y usuarios autorizados que hacen un mal uso de los privilegios que se les han asignado. También, se puede entender por intrusión a una violación de la política de seguridad del sistema. Pero, en todo caso, conviene poner de manifiesto que cualquier definición de intrusión es necesariamente imprecisa, al igual que los requisitos de política de seguridad no siempre se traducen en un conjunto totalmente definido de acciones. De esta forma, y aún cuando la política define las metas que deben satisfacerse en un sistema, los detectores de brechas de esta política enfocan toda su atención en el conocimiento de pasos o acciones que pueden producir su violación.

La detección de intrusos puede ser dividida en dos categorías principales: la detección de intrusión de anomalías y la detección de intrusión de mal uso (abuso). La primera se refiere a intrusiones que pueden descubrirse basadas en la conducta anómala y uso de recursos de computación. Por ejemplo, si el X usuario sólo usa la computadora de su oficina entre las 9:00 a.m. y las 5 p.m., una actividad en su cuenta fuera de ese horario es anómalo y, por lo tanto, puede ser una intrusión. Posteriormente, considérese a otro usuario Y, que siempre pueda conectarse fuera de las horas de trabajo a través del servidor de la compañía. Una sesión del "login" remota nocturna a su cuenta podría ser considerada extraña, anómala o simplemente "rara". La detección de la anomalía intenta cuantificar la conducta usual o aceptable y señala cualquier conducta irregular como un intruso potencial.

En el contraste, la detección de intrusión de mal uso se refiere a intrusiones que siguen modelos bien definidos de ataque que explotan las debilidades en el sistema y en el software de aplicación. Precisamente, tales modelos pueden escribirse por adelantado, tales como por ejemplo la explotación de los virus del envío de correo electrónico utilizados en ataques por Internet. Esta técnica representa el conocimiento sobre la conducta mala o inaceptable y busca descubrirlo directamente, en contraposición a la detección de intrusión de anomalías que busca descubrir el complemento de la conducta normal.

De acuerdo con lo anterior, la principal hipótesis de este trabajo es que la Lógica Borrosa es capaz de producir "mejores" reglas que incrementen la flexibilidad y

robusted de los sistemas de auditoria informática. De hecho, los Sistemas Inteligentes (SI) han demostrado ser una herramienta eficaz en problemas de control, clasificación, modelado etc., ante contextos donde la información y/o los datos están afectados de imprecisión no probabilística, lo que justifica el interés por presentar en el siguiente apartado una propuesta original de Sistema Inteligente de Detección de Intrusos.

3 Propuesta Sistema Inteligente de Detección de Intrusos

3.1 Fase1: Determinación de las variables del sistema y esquema asociativo de variables

El sistema elaborado esta encaminado a 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, se utilizan como variables de entrada las siguientes:

- Horario: indica si la intrusión se produce en horario laboral o fuera de dicho horario. Será calificado como horario laboral o no laboral.
- Directorios de sistema y/o datos: indica si el intruso accede a directorios de sistema y/o directorios de datos. Será calificado como accede (a dichos directorios) o no accede (a los directorios).
- Uso de comandos peligrosos: indica si el intruso utiliza determinados comandos considerados como peligrosos (ej. copiar, mover, modificar, borrar, crear,....). Será calificado como utiliza (dichos comandos) o no utiliza (los comandos).
- Tiempo CPU: tiempo uso de la CPU. Permitirá su clasificación como bajo, medio o alto.
- Comandos usados: número de comandos diferentes utilizados. Será calificado como bajo, medio o alto.
- Tiempo de Uso de I/O: tiempo de uso de los distintos dispositivos de I/O. En función del mismo se procederá a su clasificación como bajo, medio o alto.

Las tres primeras las podemos considerar como variables binarias, ya que solo miden si la variable se utiliza o no. Las tres siguientes se pueden considerar como variables ordinales, pues miden algunos comportamientos cuantificables numéricamente.

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

Una vez establecidas las variables del sistema se procede a determinar las relaciones existentes entre las diferentes variables del modelo, es decir, establecer el expertizaje que permite dar una salida a partir de los diferentes valores de las entradas. En este caso, dichas relaciones se pueden observar en la siguiente Figura 1, donde cabe resaltar los siguientes aspectos:

- Las tres primeras variables (horario, directorios y comandos peligrosos) se agrupan para dar lugar a otra variable intermedia 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.

- Las tres últimas variables (tiempo C.P.U., número de comandos usados y tiempo uso I/O) se agrupan a su vez en otra variable intermedia llamada frecuencia que determinará la periodicidad con que se produce la posible intrusión. Esta variable se califica como muy baja, baja, media, alta y muy alta.
- Para determinar el nivel de intrusión se tendrán en cuenta estas dos variables intermedias.

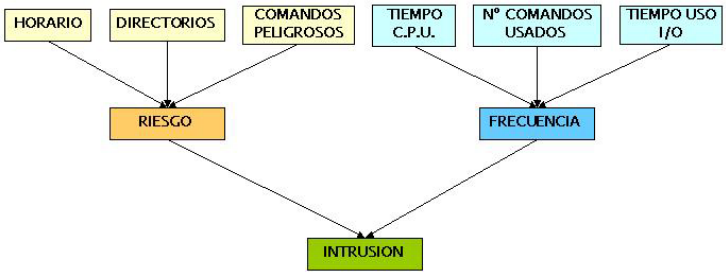

Figura 1. Esquema asociativo entre variables

3.2 Fase 2: Borrosificación de las variables

Una vez que establecidas las variables que utilizará el sistema y las relaciones existentes entre las mismas, el siguiente paso a seguir consiste en determinar el dominio de cada uno de los números borrosos representativos de las diferentes etiquetas lingüísticas en las que se divide el universo de discurso de cada variable, salvo para las tres primeras variables que, al ser variables binarias, solo pueden tener los valores 1 ó 0, tal como sigue:

3.2.1 Variable Tiempo C.P.U.

Se han diferenciado dentro de esta variable los siguientes estados: tiempo bajo, tiempo medio y tiempo alto. Los dominios asignados a los diferentes números borrosos relacionados con esta variable son:

- Tiempo C.P.U. Bajo = (0, 5, 10, 15)
- Tiempo C.P.U. Medio = (10, 25, 30, 50)
- Tiempo C.P.U. Alto = (40, 60,100, 100)

Por tanto, los valores asignados a esta variable se recogen en la Figura 2.

	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

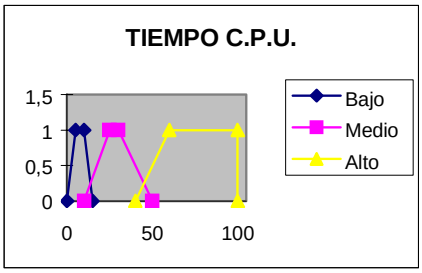

Figura 2. Número borroso de "Tiempo CPU"

3.2.2 Variable Número de comandos utilizados

La variable "número de comandos utilizados" se ha modelizado a través de tres etiquetas lingüísticas: bajo, medio y alto. Al igual que en el caso anterior, se determinaron los dominios de cada uno de los subconjuntos borrosos representativos de tales estados; siendo los valores establecidos y su representación gráfica la que se muestra en la Figura 3.

	Bajo	Medio	Alto
No menor que	0	2	5
Igual que	0	3	7
Igual que	1	4	10
No mayor que	3	6	10

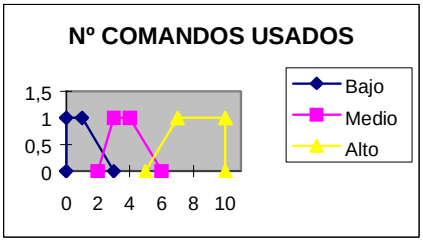

Figura 3. Número borroso de "Número de comandos utilizados"

3.2.3 Variable Tiempo uso I/O

La tercera de las variables consideradas como entradas al sistema es el tiempo de uso del sistema de I/O. En este sentido, se consideraron tres posibles estados de esta variable: bajo, medio y alto. Los números borrosos asociados a estas tres etiquetas y la representación gráfica de los mismos son los recogidos en la Figura 4.

	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

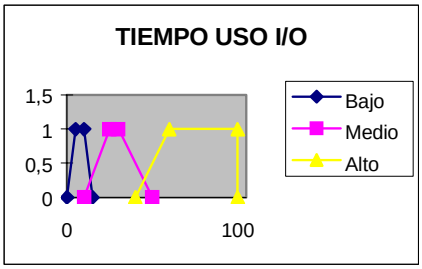

Figura 4. Número borroso de "Tiempo uso I/O"

3.2.4 Variable Intrusión

Además de las variables de entrada, se ha establecido como única variable de salida el nivel de intrusión que indicará en qué grado podemos considerar que la conexión al sistema la ha establecido un intruso. Dicha variable de salida la calificamos según cuatro etiquetas lingüísticas: informativo, sospechoso, serio y critico, siendo la representación gráfica y los números borrosos asociados a las mismas los que se recogen en la Figura 5.

	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

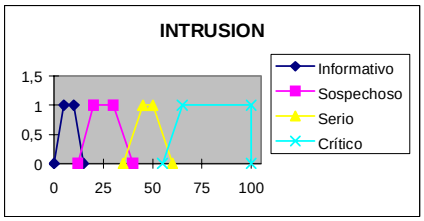

Figura 5. Número borroso de "Intrusión"

3.3 Fase 3: Establecimiento de las reglas borrosas

Una vez establecidas las definiciones de cada una de las etiquetas lingüísticas en las que se dividen las variables borrosas establecidas (tanto de entrada como de salida), y conocidas las relaciones entre las mismas, la siguiente fase radica en el establecimiento de reglas borrosas del tipo *SI* ..... *ENTONCES* .....

3.3.1 Variable intermedia "Riesgo"

Esta variable es el resultado de la combinación de las tres variables binarias (horario, directorios y comandos peligrosos), de ahí que sea necesario establecer las reglas de funcionamiento operativas para su formación, las cuales se muestran en la Tabla 1.

Tabla 1. Determinación de la variable riesgo

Horario	Directorios	Comandos Peligrosos	**Riesgo**
Laboral	Accede	Utiliza	***MEDIO***
Laboral	Accede	No utiliza	*BAJO*
Laboral	No accede	Utiliza	*MEDIO*
Laboral	No accede	No utiliza	*MUY BAJO*
No laboral	Accede	Utiliza	*MUY ALTO*
No laboral	Accede	No utiliza	*ALTO*
No laboral	No accede	Utiliza	*ALTO*
No laboral	No accede	No utiliza	*MEDIO*

3.3.2 Variable intermedia "Frecuencia"

De forma análoga a la variable anterior, esta variable es resultado de la combinación de las tres variables ordinales (tiempo CPU, nº comandos usados y tiempo uso I/O), siendo preciso, por tanto, establecer las reglas de funcionamiento operativas para su formación. Dichas reglas se recogen en el Cuadro 6.

3.3.3 Variable final "Intrusión"

La variable de salida es el resultado de la combinación de las dos variables intermedias (riesgo y frecuencia), lo que implica asimismo establecer las reglas de funcionamiento operativas para su formación, las cuales se muestran recogidas en las Tablas 2 y 3.

Tabla 2. Determinación de la variable “Frecuencia”

Tiempo Cpu	Nº Comandos Utilizados	Tiempo Uso I/O	Frecuencia
Bajo	Bajo	Bajo	MUY BAJA
Bajo	Bajo	Medio	MUY BAJA
Bajo	Bajo	Alto	BAJA
Bajo	Medio	Bajo	BAJA
Bajo	Medio	Medio	BAJA
Bajo	Medio	Alto	BAJA
Bajo	Alto	Bajo	MEDIA
Bajo	Alto	Medio	MEDIA
Bajo	Alto	Alto	MEDIA
Medio	Bajo	Bajo	BAJA
Medio	Bajo	Medio	MEDIA
Medio	Bajo	Alto	MEDIA
Medio	Medio	Bajo	MEDIA
Medio	Medio	Medio	MEDIA
Medio	Medio	Alto	MEDIA
Medio	Alto	Bajo	MEDIA
Medio	Alto	Medio	ALTA
Medio	Alto	Alto	ALTA
Alto	Bajo	Bajo	MEDIA
Alto	Bajo	Medio	ALTA
Alto	Bajo	Alto	ALTA
Alto	Medio	Bajo	ALTA
Alto	Medio	Medio	ALTA
Alto	Medio	Alto	ALTA
Alto	Alto	Bajo	ALTA
Alto	Alto	Medio	MUY ALTA
Alto	Alto	Alto	MUY ALTA

Tabla 3. Determinación de la variable "Intrusión"

Riesgo	Frecuencia	Intrusión
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	CRITICO
Alto	Muy Alta	CRITICO
Muy Alto	Muy Baja	CRITICO
Muy Alto	Baja	CRITICO
Muy Alto	Media	CRITICO
Muy Alto	Alta	CRITICO
Muy Alto	Muy Alta	CRITICO

3.4 Fase 4: Funcionamiento del sistema borroso

3.4.1. Variables de entrada

Una vez establecido el esquema general del sistema, es necesario determinar el funcionamiento del mismo. De esta forma, ante unos determinados datos de entrada ("crisp") en primer lugar será necesario determinar el grado de activación o grado de verdad de cada una de las etiquetas integrantes de las diferentes variables de entrada horario, directorios, comandos peligrosos, tiempo CPU, número de comandos usados y tiempo de uso I/O. Así, por ejemplo, para la variable tiempo CPU existen tres diferentes variables lingüísticas: Tiempo CPU medio y Tiempo CPU alto. El grado de activación de cada una ante una determinada entrada se determina como sigue para "Tiempo CPU bajo (0, 5, 10, 15)":

- Si el dato del ejemplo es menor o igual que 0, entonces el grado de pertenencia al subconjunto será 0.
- Si el dato es mayor o igual que 15, entonces el grado de pertenencia será 0.
- Si el dato es mayor o igual que 5 y menor o igual que 10, entonces el grado de pertenencia es 1.
- Si el dato es mayor que 0 y es menor que 5, entonces el grado de pertenencia es (dato-0)/(0-0).
- Si el dato es mayor que 10 y menor que 15, entonces el grado de pertenencia es (15-dato)/(15-10).

Lógicamente, el grado de pertenencia total al subconjunto se calcula como la suma de los grados de pertenencia resultantes.

Se procedería de la misma forma con todas las etiquetas lingüísticas de cada una de las variables de entrada.

3.4.2 Variable intermedia "Riesgo"

Para determinar el grado de pertenencia de esta variable, será necesario conocer el grado de verdad de la regla utilizada, o peso de la regla (que coincidirá con el grado de pertenencia buscado).

En la medida que se trata de reglas compuestas por más de un antecedente, el grado de verdad de la regla determinará a partir del grado de verdad de cada uno de los antecedentes, relacionados a través de una T-norma. En este caso la T-norma que va a utilizarse será el producto, de tal forma que:

$$\mu_t \text{ (RIESGO)} = \mu_x \text{ (HORARIO)} * \mu_y \text{ (DIRECTORIOS)} * \mu_z \text{ (COMANDOS)}$$

Si se observa la tabla, existen diferentes reglas con el mismo consecuente:

SI Tiempo Horario laboral y directorios accede y comandos peligrosos utiliza **ENTONCES** Riesgo medio.

SI Tiempo Horario laboral y directorios no accede y comandos peligrosos utiliza **ENTONCES** Riesgo medio.

Para determinar del grado de pertenencia total al subconjunto Riesgo medio, tenemos que elegir una de las opciones propuesta por Sugeno o Mamdani; en nuestro caso hemos elegido la opción propuesta por Sugeno, ya que mantiene la unicidad en el grado de pertenencia total a los diferentes subconjuntos borrosos de una variable.

$$\mu_{\text{MEDIO TOTAL}} \text{ (RIESGO)} = \mu_{\text{MEDIO1}} \text{ (RIESGO)} + \mu_{\text{MEDIO 2}} \text{ (RIESGO)} + \mu_{\text{MEDIO 3}} \text{ (RIESGO)}$$

3.4.3 Variable intermedia "Frecuencia"

La variable borrosa frecuencia se obtiene a partir de los datos relativos a tres variables iniciales, tiempo CPU, número de comandos usados y tiempo uso I/O; esta variable tiene cuatro estados diferenciados: Frecuencia muy baja, Frecuencia baja, Frecuencia media, Frecuencia alta y Frecuencia muy alta.

Lógicamente, para determinar cada uno de estos estados será preciso utilizar reglas lógicas del tipo:

Para determinar el grado de pertenencia de FRECUENCIA, será necesario conocer el grado de verdad de la regla utilizada, o peso de la regla (que coincidirá con el grado de pertenencia buscado).

Dado que se trata de reglas compuestas por más de un antecedente, el grado de verdad de la regla determinará a partir del grado de verdad de cada uno de los antecedentes, relacionados a través de una T-norma. En este caso la T-norma que va a utilizarse será el producto, de tal forma que:

$$\mu_t \text{ (FRECUENCIA)} = \mu_x \text{ (TIEMPO CPU)} * \mu_y \text{ (Nº COMANDOS)} * \mu_z \text{ (TIEMPO USO I/O)}$$

Si se analiza la tabla se puede observar que existen diferentes reglas con el mismo consecuente, como por ejemplo:

SI Tiempo CPU bajo **y** Nº comandos bajo **y** Tiempo uso I/O bajo
ENTONCES Frecuencia muy baja.

SI Tiempo CPU bajo **y** Nº comandos bajo **y** Tiempo uso I/O medio
ENTONCES Frecuencia muy baja.

El grado de pertenencia total de cada consecuente se determinará a partir de la regla de Sugeno, es decir, como suma de los grados de pertenencia parciales de cada una de las reglas que lo definen.

$$\mu_{\text{MUY BAJA TOTAL}}(\text{FRECUENCIA}) =$$
$$= \mu_{\text{MUY BAJA1}}(\text{FRECUENCIA}) + \mu_{\text{MUY BAJA 2}}(\text{FRECUENCIA})$$

3.4.4 Variable final "Intrusión"

Para determinar el grado de pertenencia de la variable de salida "Riesgo", será necesario conocer el grado de verdad de la regla utilizada, o peso de la regla (que coincidirá con el grado de pertenencia buscado).

Al tratarse de reglas compuestas por más de un antecedente, el grado de verdad de la regla determinará a partir del grado de verdad de cada uno de los antecedentes, relacionados a través de una T-norma. En este caso la T-norma que va a utilizarse será el producto, de tal forma que:

$$\mu z\,(\text{INTRUSION}) = \mu x\,(\text{RIESGO}) * \mu y\,(\text{FRECUENCIA})$$

Dado el conjunto de normas definido previamente, se puede observar que al igual que en el caso de las variables intermedias (riesgo y frecuencia), existen diferentes reglas que dan lugar al mismo consecuente, por tanto, el grado de pertenencia total de cada consecuente se determinará a partir de la regla de Sugeno, es decir, como suma de los grados de pertenencia parciales de cada una de las reglas que lo definen.

3.5. Fase 5: Desborrosificación

Una vez obtenido el grado de pertenencia borroso de cada ejemplo a cada subconjunto borroso de la variable final (en este caso, la evaluación del nivel de intrusión), el último paso consiste en determinar, a partir de dicha evaluación borrosa, la calificación del riesgo de intrusión.

Para ello se precisa proceder a la desborrosificación de la “figura” borrosa obtenida, cuya representación gráfica se muestra en la Figura 6, en la que se puede observar que la misma responde al tipo de evaluación de reglas borrosas que se conoce con el nombre de producto aritmético, y que se utiliza para la elaboración de sistemas expertos en incertidumbre.

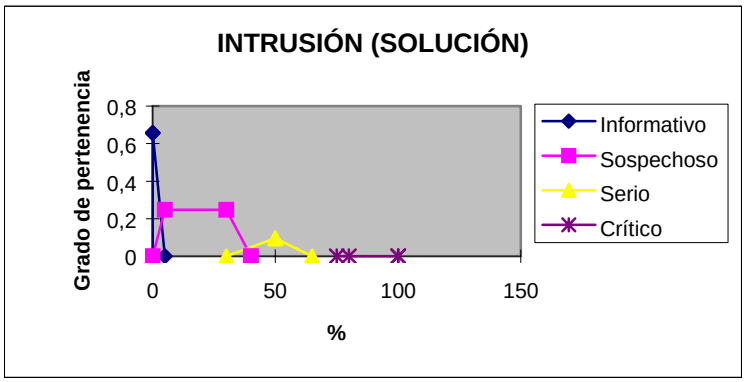

Figura 6. Representación de la variable de salida “Intrusión”

Para conocer el nivel de intrusión, es necesario proceder a la desborrosificación de la figura anterior. Aunque existen múltiples métodos de desborrosificación, a los efectos del presente trabajo se ha aplicado el método del centroide o centro de gravedad, que en el caso de trabajar con distintos subconjuntos borrosos de tipo trapezoidal se traduce en:

$$\text{Centroide} = \frac{\Sigma[a_1 + a_2 * (1 + \mu_A(y)) + a_3 * (1+\mu_A(y)) + a_4]_i}{\Sigma[1 + (1 + \mu_A(y)) + 1 + \mu_A(y)]_i}$$

Donde i hace referencia a los distintos subconjuntos borrosos existentes en la variable final (en este caso, la variable final es el nivel de intrusión con cuatro subconjuntos borrosos definidos: informativo, sospechoso, serio y critico).

De esta forma, puede llegarse a determinar el nivel de intrusión de una conexión realizada al sistema.

4 Interface de la aplicación: ejemplo práctico

El interface de la aplicación es muy sencillo, ya que en una sola pantalla se pueden visualizar todas las variables, tanto las de entrada, como las variables intermedias como la variable de salida. A modo ilustrativo se muestra la pantalla en la Figura 7.

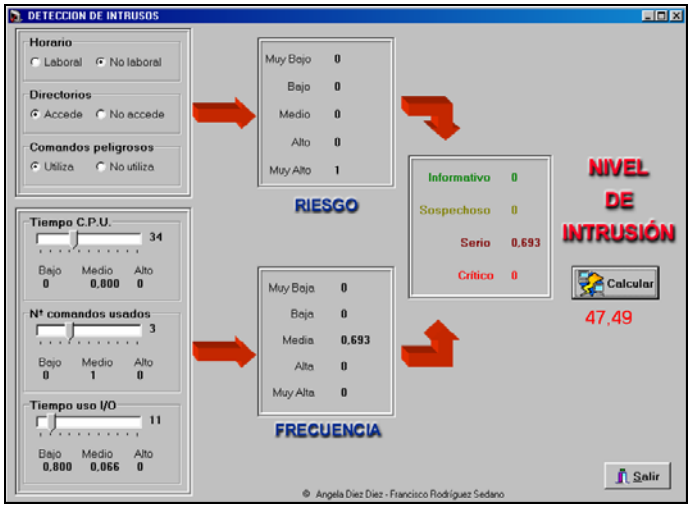

Figura 7. Interface gráfica del SIDI

En un primer bloque se encuentran las variables de entrada clasificadas como binarias: horario, directorios y comandos peligrosos. Solo hay que seleccionar el valor de dichas variables haciendo clic en la casilla correspondiente.

En el siguiente bloque se recogen las variables de entrada ordinales: tiempo CPU, nº de comandos peligrosos y tiempo de uso I/O. El valor de estas variables se puede elegir desplazando la barra con el ratón hasta visualizar a la derecha el valor correcto dentro del rango establecido para cada variable.

En un segundo bloque se pueden observar las dos variables intermedias: riesgo y frecuencia. Cada vez que se cambie un valor de las variables de entrada correspondientes, el programa de aplicación actualiza los valores de las variables intermedias.

Por último, se establece la variable de salida: nivel de intrusión, que también se actualiza automáticamente. Para calcular el valor *crisp* de esta variable de salida, solo hay que pulsar el botón inferior, con lo que se podrá visualizar el mismo debajo de dicho botón.

5 Conclusiones

En este trabajo se ha analizado la problemática que supone el descubrimiento de amenazas en el campo de la auditoria informática, especialmente la relacionada con la detección de intrusos. A este respecto, la hipótesis de trabajo se ha centrado en la consideración de la aplicación de la Lógica Borrosa como herramienta eficaz en este ámbito.

De esta forma, en el presente trabajo se ha presentado una propuesta original de sistema de detección de intrusos con un sistema experto embebido, detallándose las fases de su diseño e implementación práctica incluyendo una imagen gráfica del interface desarrollado.

6 Referencias

1. Agrawal, R., Imielinski, T. and Swami, A.: "Mining association rules between sets of items in large databases". Proceedings of the ACM SIGMOD Conference on Management of Data (1993) 207-216
2. Bardossy, A. and Duckstein, L.: "Fuzzy Rule-Based Modeling with Applications to Geophysical, Biological and Engineering Systems". CRC Press (1995)
3. Cios, K., Pedrycz, W. and Swiniarski, R.W.: "Data Mining. Methods for Knowledge Discovery". Kluwer (1998)
4. Cordón, O; Del Jesús, M; Herrera, F. and López, E.: “Selecting fuzzy rule-based classification systems with specific reasoning methods usin genetic algorithms”. 7th International Fuzzy Systems Association World Congress, Praga, junio, (1997) 424-429.
5. Denning, D.E.: "An intrusion-detection model". IEEE Transaction on Software Engineering (1987) 222-232.
6. Ilgun, K., Kemmerer, R. and Porras, P.: "State transition analysis: A rule-based intrusion detection approach". IEEE Transactions on Software Engineering (1995) 181-199.
7. Kumar, S. and Spafford, E.: "A software architecture to support misuse intrusion detection". Proceedings of the 18th National Information Security Conference (1995) 194-204.

8. Lam, L. and Suen, C.: "Application of Majority Voting to Pattern Recoginition and Analysis of its Behaviour and Performance". IEEE Transactions on Systms, Man, and Cybernetics (1997) 553-568.
9. Lane, T. and Brodley, C.: "Temporal sequence learning and data reduction for anomaly detection". 5th ACM Conference on Computer & Communications Security, San Fransico (1998) 150-158.
10. Liu, H. and Motoda, H.: "Feature Selection for Knowledge Discovery and Data mining". Kluwer (1998)
11. López González, E.: "A Methodology for Building Fuzzy Expert Systems (FES) with Spreadsheet to Quality Function Deployment (QFD) of the Target Costing". Incluido en Gil Aluja, J. (ed.): "Handbook of Management under Uncertainty", Kluwer Academic Publishers, Dorchecht (2001) 221-245
12. Luo, J.: "Integrating Fuzzy Logic with Data Mining Methods for Intrusion Detection", A Thesis Submitted to the Faculty of Mississippi State University in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science in the Department of Computer Science Mississippi State, Mississippi (1999)
13. Me, L. and Alanoun V.: "Detection d'intrusions dans un systeme informatique: methodes et outils". TSI (1996) 29-45.
14. Mukherjee, B., Heberlein, L. and Levitt, K.: "Network intrusion detection". IEEE Network (1994) 26-41.
15. Pedrycz, W.: "Fuzzy Set Technology in Knowledge Discovery". Fuzzy Sets and Systems (1998) 279-290.
16. Sugeno, M. and Yasukawa, T.: "A Fuzzy Logic-Based Approach to Linguistic Modeling". IEEE Transactions on Fuzzy Systems (1993) 7-31.
17. Warrender, C., Forrest, S. and Perlmutter, B.: "Detecting intrusions using system calls: Alternative data models". IEEE Symposium on Security and Privacy, Berkeley (1999) 133-145.
18. White, G. and Pooch, V.: "Cooperating security managers: Distributed intrusion detection systems". Computers & Security (1996) 441-450.
19. Zadeh, L.A.: "Fuzzy Logic=Computing with Words", IEEE Transactions on Fuzzy Systems (1996) 103-111.