MODELO PARA LA DETECCION DE INTRUSOS EN LA SEGURIDAD DE SISTEMAS INFORMATICOS APLICANDO LOGICA DIFUSA

2011

UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA

TESIS DE GRADO

MODELO PARA LA DETECCION DE INTRUSOS EN LA SEGURIDAD DE SISTEMAS INFORMATICOS APLICANDO LOGICA DIFUSA

PARA OPTAR AL GRADO DE LICENCIATURA EN INFORMÁTICA
MENCION INGENIERIA DE SISTEMAS INFORMATICOS

POSTULANTE: Univ. Ines Margarita Ramos

TUTOR: Lic. Eufren Llanque Quispe

REVISOR: Lic. Carlos Mullisaca Choque

LA PAZ – BOLIVIA
2011

Dedicatoria

Con todo mi amor a mi mamá, Esperanza…

A mi familia y a mis tres pequeñas adoraciones…

Agradecimientos

A Dios, por haberme guiado en todos los pasos que di durante todos los años de mi vida…

Quiero agradecer al Lic. Llanque y al M.Sc. Mullisaca, tutor y revisor del presente trabajo, quienes con sus observaciones y sugerencias contribuyeron a perfeccionar y culminar el tema de investigación.

También, agradezco a todos los docentes de la Carrera de Informática que día a día transmiten sus conocimientos a los estudiantes para formar profesionales capaces de enfrentar los retos que nos esperan al culminar nuestros estudios en la universidad. Al personal administrativo, en especial a los encargados de la biblioteca por la colaboración brindada en todo momento.

Finalmente a mi mamá, a quien debo todo lo que soy. Gracias por tu apoyo incondicional… a mis amigos en general, quienes de alguna manera me ayudaron con su apoyo moral para culminar con la presente tesis.

…. gracias a todos.

Resumen

En los últimos años, la seguridad informática se ha vuelto en una prioridad importante debido al incremento en el uso de las computadoras, el surgimiento del comercio electrónico y el rápido crecimiento de las redes de computadoras, desde entonces las intrusiones toman ventaja de las vulnerabilidades del sistema..

La detección de intrusos es un componente esencial y crucial para la seguridad de los sistemas informáticos. Una de las razones es que no resulta técnicamente posible construir un sistema invulnerable. De hecho es muy difícil probar la capacidad de seguridad de un sistema ya que es casi imposible incorporar todos los patrones de intrusión conocidos. Además, los intrusos podrían usar patrones completamente desconocidos los cuales son difíciles de detectar. Por otro lado la mayoría de las intrusiones se originan dentro la red por usuarios que abusan de los derechos de acceso que les han sido otorgados, los cuales no causarían alarma sin el uso de un método de detección de intrusos.

El presente trabajo plantea un modelo para la detección de intrusos aplicando lógica difusa, el cual determina la posible intrusión de un usuario a un sistema, ya que se observó la vulnerabilidad a la que están expuestos los sistemas informáticos, sí estos no cuentan con las herramientas necesarias para su debida protección.

Abstract

In recent years, computer security has become increasingly important and an international priority. This is due to the wide use of computers, the emergence of electronic commerce, and the rapid growth of computer networks. Since intrusions will take advantage of vulnerabilities in computer systems, intrusion detection methods are usually developed to enforce the security policy of computer hosts and computer networks.

In a modern computer system, intrusion detection has become an essential and critical component. One of the reasons is that it is not technically feasible to build a system without any vulnerability. As a matter of fact, it is also very difficult to test the security capabilities of a system since it is almost impossible to incorporate all intrusion patterns. In addition, future attackers may use completely unknown patterns which are unexpected and difficult to detect. On the other hand, intrusions originating from authorized system users who choose to abuse their access rights will not cause an alarm without the use of intrusion detection methods.

This research proposes an intrusion detection model applying fuzzy logic. The model determines the possibility of intrusion in a system, due to systems are exposed to many risks, if them do not have tools to their protection.

INDICE

Pág.

Lista de Figuras

Pág.

Lista de Cuadros

Pág.

Capítulo I

Marco Referencial

Resumen

Se presenta la introducción y antecedentes referidos a la Detección de Intrusos en la Seguridad Informática. También se describe el planteamiento del problema, la hipótesis, los objetivos, la justificación, los alcances y aportes, además de la metodología que se empleara para realizar el presente trabajo.

1.1 Introducción

En los últimos años, debido al incremento en el uso de las computadoras y el espectacular crecimiento del internet y de los servicios que este ofrece, la detección de intrusos se ha convertido en una prioridad importante, ya que no resulta técnicamente factible construir un sistema invulnerable, pues a pesar de la existencia de numerosas medidas de seguridad para proteger los recursos informáticos de cualquier organización económica y aún cuando se respeten escrupulosamente todas las políticas de seguridad y las recomendaciones de los expertos, no se puede suponer la ausencia de posibles ataques con éxito.

En multitud de ocasiones los intrusos de la red han superado los mecanismos de autentificación diseñados para proteger los sistemas, pues con el aumento en el entendimiento sobre cómo funcionan los sistemas, los intrusos se han vuelto 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.

La seguridad de los sistemas informáticos es uno de los principales problemas con los que podemos encontrarnos hoy en día. Resulta de vital importancia la conservación, almacenamiento e integridad de la información en formato electrónico, ya que esta ha

pasado de ser un elemento abstracto de baja importancia, a ser considerada incluso como un activo dentro del capital de las empresas.

El presente trabajo plantea un modelo para la detección de intrusos aplicando lógica difusa, el cual determina la posible intrusión de un usuario al sistema, ya que se observó la vulnerabilidad a la que están expuestos los sistemas informáticos, sí estos no cuentan con las herramientas necesarias para su debida protección.

1.2 Antecedentes

La mayoría de los sistemas informáticos 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. La mayor cantidad de 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.

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 computadoras y 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[1] (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.

[1] Para mayor referencia consultar Anexo A

La cantidad de intentos de accesos no autorizados a la información que existe en Internet ha crecido de una manera alarmante. Según el Instituto de Seguridad Computacional *(Computer Security Institute)* [CSI09] un 70% de las organizaciones anunciaron al menos un incidente de seguridad durante los últimos tres años, frente a un 42% anunciado en 2009. La mayoría de los expertos piensa que estos números están por debajo de la tasa real, puesto que muchas organizaciones evitan dar a conocer sus incidentes y muchas otras ni siquiera los detectan.

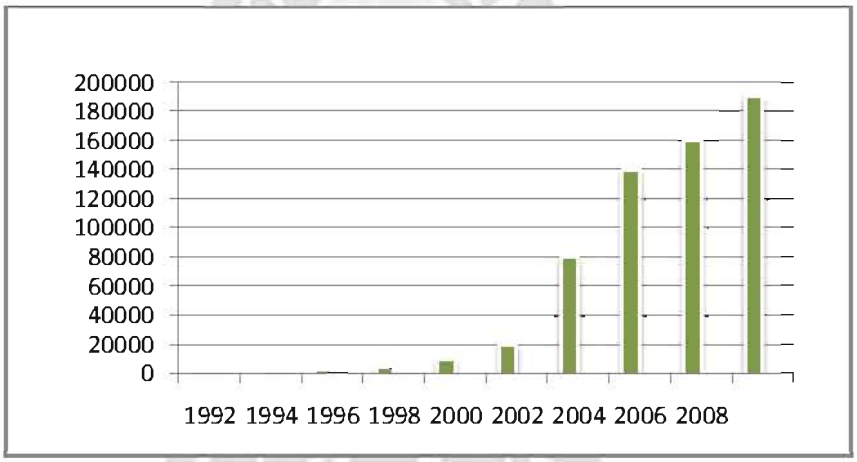


Figura 1.1 *Evolución de los incidentes remitidos al CSI.*

Fuente. *Instituto de Seguridad Computacional [CSI09]*

En el ámbito local los datos de la Fuerza Especial de Lucha Contra el Crimen (FELCC) revelan que en Santa Cruz, La Paz y Cochabamba se producen más delitos informáticos desde 2003. Desde ese año hasta 2009, la Policía registró 185 fraudes electrónicos en todo el país, de éstos, 177 corresponden a manipulación informática y ocho a alteración, acceso y uso indebido de información [LAPRENSA09].

Actualmente en la Universidad Mayor de San Andres en la Carrera de Informática se realizaron los siguientes trabajos relacionados al tema del presente trabajo:

- Tesis de Grado *"Seguridad en Internet utilizando redes neuronales artificiales para la detección de Intrusos"*, fue realizado por Orozco Orlando en el año 2001.

En el trabajo se hace la construcción de un modelo neuronal y su correspondiente implementación en un software neuronal para la detección de intrusos.

- Tesis de Grado *"Agentes Autónomos para la Detección de Intrusos en Redes"*, fue realizado por Miranda Quisbert Edgar en el año 2004. En el trabajo se realizo el diseño de un agente para la detección de intrusos en redes en tiempo real, basado en Sistemas de Detección de Intrusos (SDI).

1.3 Planteamiento del Problema

La seguridad informática es un aspecto al que se le presta mucha atención en la actualidad, como consecuencia del valor que se le ha dado a la información almacenada en las computadoras y al desarrollo de grandes redes como el internet; las empresas e instituciones aceptan los riesgos de este entorno de intercomunicaciones distribuido y se preocupan cada vez más por la seguridad de sus sistemas.

El escaso conocimiento que se tiene sobre seguridad informática hace que los usuarios de computadoras conectadas a una red resulten un blanco fácil para aquellos que buscan un rédito económico, a partir del fraude o el robo de información confidencial. En un nuevo informe sobre las amenazas a la seguridad en la Red, la empresa desarrolladora de software antivirus Symantec [SYMANTEC10] reveló que el 86% de los ataques informáticos están dirigidos a ordenadores de hogar, puesto que son los que menos posibilidades tienen de implementar medidas de protección efectivas. Según dicha compañía, la tendencia marca ataques pequeños y dirigidos, que se centran en el fraude, el robo de información y la actividad delictiva.

Una de las metas de los intrusos informáticos es sacar provecho de las carencias de seguridad no solo electrónicas, sino también humanas, pasando por alto los mecanismos de detección, durante el periodo analizado, fueron identificadas más de 4,6 millones de

computadoras pertenecientes a redes Bot[2], con un promedio diario de 57,717 equipos activos. Asimismo, se observo un promedio de 6,110 ataques diarios de negación de servicio (DOS). Se estima un aumento de los ataques que aprovechan los conceptos de Web 2.0, como la publicidad basada en el usuario y tecnologías como AJAX[3], así como un aumento de las vulnerabilidades reportadas debido al uso de fuzzers[4].

Una intrusión es cualquier conjunto de acciones que intente comprometer la integridad, confidencialidad o disponibilidad de un recurso. Las técnicas de prevención de la intrusión como la autenticación de usuario, evitar los errores de programación y proteger la información almacenada y en tránsito se han utilizado como primera línea de defensa para proteger sistemas de computación. La prevención de intrusiones por sí sola no es suficiente ya que cuando los sistemas se hacen más complejos, siempre existen vulnerabilidades explotables en los sistemas debido a errores de diseño, programación, gestión o a diferentes técnicas de penetración (por ejemplo, ataques sistemáticos en paralelo desde múltiples puntos y paralelización de scripts intrusivos).

Las políticas que establecen equilibrio entre el control estricto de un sistema y el acceso a la información también hacen imposible que un sistema en funcionamiento sea completamente seguro. Por tanto, es necesaria la detección de intrusiones como una segunda barrera para proteger los sistemas informáticos. Los elementos centrales para detectar las intrusiones son los recursos a proteger en un sistema destino, por ejemplo las cuentas de usuario, los sistemas de ficheros, los kernels del sistema, etc., modelos que caracterizan el comportamiento normal o legítimo de estos recursos, técnicas para comparar las actividades del sistema real con los modelos establecidos e identificar los intrusivos o anormales.

---

[2] ***"Bot"*** computadora Zombi cuyo propósito es generar ataques.

[3] ***"Ajax"*** nueva tecnología para crear páginas Web, acrónimo de Asíncrono Java Script y XML

[4] ***"Fuzzer"*** script diseñado para encontrar vulnerabilidades en el código de software.

La detección de intrusos es un área clave, conforme las redes se van haciendo más complejas. El administrador de red debe tener un conocimiento detallado del tráfico y de los posibles ataques que se intenten. Puesto que los sistemas informáticos desempeñan un papel vital de creciente importancia en la sociedad moderna, se han convertido en los objetivos de explotación de intrusos y delincuentes. Por consiguiente se necesita disponer de los mejores mecanismos para proteger nuestros sistemas. La seguridad de un sistema de computación se ve comprometida cuando tiene lugar una intrusión.

1.3.1 Formulación del problema

De acuerdo con lo anterior, el problema planteado es:

"La vulnerabilidad a la que están expuestos los sistemas informáticos, ya que está puede ser aprovechada por alguien con malas intenciones, por lo que existe la posibilidad de intrusión, si el sistema no cuenta con las herramientas necesarias para la detección de intrusos."

1.4 Objetivos

Los objetivos planteados para el trabajo de investigación son:

1.4.1 Objetivo General

Desarrollar un modelo, que determine la posibilidad de intrusión de un usuario a un sistema informático, a partir de variables utilizadas en la detección de intrusos.

1.4.2 Objetivos Específicos

- Realizar un estudio profundo de los sistemas de detección de intrusos, para la elaboración del modelo.

- Identificar la estructura superficial del modelo a desarrollar.

- Formalizar el conocimiento obtenido haciendo uso del modelado difuso.

- Implementar el modelo desarrollado en un software especializado para la aplicación de lógica difusa.

- Generar resultados satisfactorios con la construcción e implementación del modelo difuso.

- Demostrar la aplicabilidad de la Lógica Difusa para la Detección de Intrusos en la Seguridad de Sistemas Informáticos.

1.5 Hipótesis

La aplicación de la Lógica Difusa determina la posibilidad de intrusión de un usuario a un sistema informático.

1.6 Conceptualización de Variables

En la figura se muestra la conceptualización de variables de acuerdo al problema planteado.

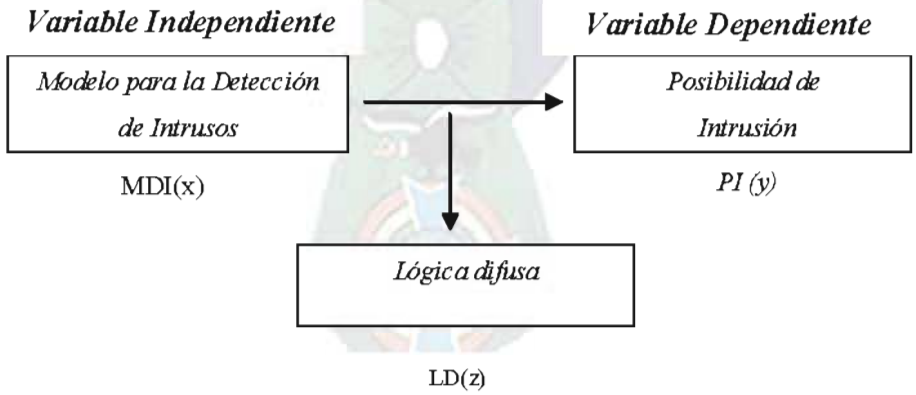


Figura 1.2 *Conceptualización de Variables*
Fuente: *Elaboración Propia*

1.7 Justificación

Se requiere llevar a cabo la presente tesis atendiendo la necesidad imperante de poseer mecanismos más dinámicos relacionados a la detección de intrusos en la seguridad de los sistemas informáticos.

1.7.1 Justificación Científica

El grado de desarrollo tecnológico en lo que se refiere a las comunicaciones o el avance tecnológico, el acceso a Internet obliga a utilizar los nuevos adelantos tecnológicos propuestos por la inteligencia artificial como lo es la lógica difusa en cuanto a la detección de intrusos en un medio que está empezando a dirigir estos adelantos y que necesita de respuesta a problemas que involucran modernización, renovación e investigación en el área de la ciencia de la informática.

1.7.2 Justificación Económica

En la actualidad se busca la utilización eficiente de recursos económicos ya sean estos estatales o privados, el presente trabajo proporciona una nueva herramienta para la seguridad en los sistemas informáticos permitiendo tener un sistema más seguro, lo cual provocaría un impacto directo e indirecto en la economía de las empresas que hacen uso de los mismos y esto beneficiara a la economía en general haciéndose un bien social importante.

1.7.3 Justificación Técnica

El uso de la lógica difusa, para el diseño y formalización del modelo planteado nos brinda una nueva perspectiva frente a las tradicionales herramientas que se usan para la detección de intrusos. En este sentido, el presente trabajo es un aporte conceptual y metodológico para la evaluación de la vulnerabilidad en la seguridad de sistemas informáticos.

1.7.4 Justificación Social

El beneficio que se procura lograr con la construcción del modelo es dotar a los profesionales en el área de informática de una herramienta útil que proponga un enfoque diferente además de nuevos esquemas que permiten una representación más cercana a la realidad, ya que es importante buscar otras y nuevas estrategias en las cuales, se busque la eficiencia en cuanto a contar con información precisa para la seguridad de los sistemas informáticos.

1.8 Metodología y Herramientas

El desarrollo de la presente tesis se apoya dentro del método científico, que sirve de guía para la organización del proceso de investigación, ya que cubrirá los requerimientos necesarios para llegar al cumplimiento de los objetivos planteados. La investigación científica es muy importante para poder resolver diferentes problemas aplicando diferentes principios y conceptos que clasifican de la teoría a la práctica.

Para el diseño del modelo se hará uso del modelado difuso *(Fuzzy Modeling)*. El sistema de reglas difusas hace que el incorporar conocimiento experto al sistema sea muy fácil. Por lo tanto, el modelado difuso tiene como ventaja que aprovecha conocimiento del dominio. Se pueden usar técnicas convencionales para el aprendizaje estructural y para métrico de las reglas.

1.9 Alcances y Límites

El modelo desarrollado determina la posibilidad de intrusión que se pueda producir en un sistema informático a partir de variables utilizadas en la detección de intrusos haciendo uso de la lógica difusa como herramienta, ya que las posibilidades que esta ofrece para abordar los problemas de decisión en este campo son amplias.

Esta herramienta constituye una opción dentro de las herramientas de protección de los sistemas informáticos, no existe una solución integral o única, deben utilizarse conjuntamente un grupo de elementos de protección y prevención para obtener un sistema informático fiable. Además al ser una herramienta pasiva, es necesaria la intervención humana para las tareas de configuración, diseño de políticas de seguridad y determinación de procedimientos o estrategias a seguir cuando reporta una posible intrusión.

La clasificación de detección de intrusos que se toma en cuenta para el presente trabajo es de acuerdo a la estrategia del análisis tomando en cuenta solo la detección de intrusión por anomalías.

1.10 Aportes

El modelo planteado ira en beneficio de los profesionales informáticos a fin de que permita determinar la posibilidad de intrusión de un usuario a un sistema informático y de esta manera obtener un optimo funcionamiento o integridad de la información almacenada, además de manifestar la capacidad de análisis que ofrece la lógica difusa para la detección de intrusos.

Capítulo II

Marco Teórico

Resumen

Se presenta una visión general de los aspectos teóricos en cuanto a la Seguridad Informática y la Detección de intrusos, definición, la clasificación de ataques informáticos e intrusiones, técnicas de detección de intrusos, políticas de seguridad. La lógica difusa, su definición, características, elementos y modelos además de la metodología para la construcción del modelo difuso.

2.1 Seguridad Informática

2.1.1 Introducción

La seguridad de datos digitales estaba muy ligada en su inicio a diseñadores de sistemas operativos. Los primeros sistemas de tiempo compartido (varias personas comparten una computadora, pero cada una de ellas trabaja como si estuviera sola) a principios de los 60 ya tenían esquemas de contraseña como parte del sistema de accesos, hardware de protección de memoria o listas de control de acceso sobre ficheros. Para 1970, el significado de garantizar la seguridad y la protección ya se consideraba como fundamental en sistemas operativos a la hora de diseñar sus núcleos.

2.1.2 Definición

La primera definición general del término seguridad de sistemas informáticos la proporcionan Garfinkel y Spafford [GARFINKEL99] como aquello de lo que se puede depender para que se comporte como se espera. Un sistema es seguro desde el punto vista de un usuario si la serie de acciones que cada actor puede realizar están limitados por lo que el usuario cree que el actor puede hacer [KUMAR00].

Una definición más exhaustiva de lo que se conoce como seguridad informática está basada en la obtención de confidencialidad, integridad y disponibilidad en un sistema informático [RUSELL97]. La confidencialidad requiere que la información sea accesible únicamente por aquellos que estén autorizados, la integridad que la información se mantenga inalterada ante accidentes o intentos maliciosos, y disponibilidad significa que el sistema informático se mantenga trabajando sin sufrir ninguna degradación en cuanto a accesos y provea los recursos que requieran los usuarios autorizados cuando éstos los necesiten.

2.1.3 Mecanismos de protección

El cifrado es el más antiguo de entre los mecanismos de protección. Un sistema secreto se define como un conjunto de transformaciones de un espacio a otro espacio, donde cada transformación en particular corresponde a un cifrado con una llave particular. La criptografía es el estudio de sistemas matemáticos que involucra a dos problemas de seguridad: Privacidad y autenticación. Por otro lado, la mayoría de los sistemas de ordenadores proveen mecanismos de control de accesos en su primera línea de defensa. Este mecanismo únicamente limita el acceso a un objeto en el sistema, pero no modela ni restringe qué es lo que un sujeto puede hacer con el objeto en el caso de que tenga acceso a su manipulación [DENNING82].

El flujo de información se puede controlar para incrementar la seguridad mediante la aplicación de modelos para proveer confidencialidad eintegridad. Existen modelos conservadores que restringen operaciones de lectura y escritura para asegurar que no se pueda comprometer la integridad y la confidencialidad de los datos de un sistema. Por ello, un sistema completamente seguro no sería de gran utilidad ya que sería demasiado restrictivo [KUMAR00].

Los controles de acceso y modelos de protección no son útiles ante amenazas internas. Si una contraseña es débil y se compromete, las medidas de control de acceso no pueden prevenir la pérdida o corrupción de la información a la que el usuario estaba autorizado a

acceder. En general, los métodos estáticos de aseguramiento de propiedades de seguridad en un sistema o son insuficientes, o pueden resultar demasiado restrictivos para los propios usuarios. Por otro lado también se tienen mecanismos de identificación y autenticación (I&A). Estos mecanismos posibilitan la identificación adecuada de los sujetos y objetos del sistema. La identificación es la declaración de quién es el usuario (conocido a nivel global), mientras que autenticación es la prueba o confirmación de esa identificación [NSA89].

Existen tres tipos de identificación: La declaración de identidad, identidad colectiva y habilidad. Asimismo, esas identidades de los usuarios se verifican mediante tres métodos genéricos: Lo que saben (contraseñas, PIN,…), lo que tienen (tarjetas magnéticas, claves electrónicas,…) y finalmente lo que son (autenticación biométrica como iris, huellas dactilares,…). Por último, hay otra serie de mecanismos con el objetivo de velar por la disponibilidad de un sistema. Algunos de ellos actúan a modo de filtros, dejando pasar aquella información que esté autorizada, en el caso de routers (con listas de acceso o ACL) y cortafuegos. Y por último los que de alguna manera detectan amenazas, como antivirus y sistemas de detección de intrusos. Éstos últimos forman la última línea de defensa en el esquema general de protección de un sistema informático, y no sólo son útiles para detectar incidentes de seguridad, sino también intentos de romper la seguridad.

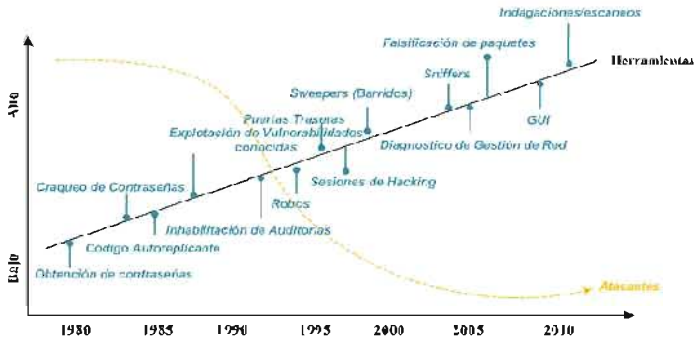


Figura 2.1 *Sofisticación de los ataques vs. Conocimiento técnico del intruso*

Fuente: *Computer Emergency Response Team/Coordination Center [CERT09]*

2.1.4 Amenazas

El uso creciente de los sistemas de ordenadores ha exacerbado el problema de accesos no autorizados y la manipulación de datos. El alto nivel de conectividad con el que se cuenta actualmente, no sólo proporciona acceso a gran cantidad y variedad de fuentes de datos más rápido que nunca antes, sino que lo provee desde cualquier lugar en la red. Desde el ataque del gusano Internet de 1988, ocurrieron una innumerable cantidad de intrusiones de red que se han saltado los mecanismos establecidos para la protección de los sistemas.

Otro aspecto a tener en cuenta es la dificultad que conlleva la realización de software, ya que éste es cada vez más complejo y el ciclo de vida del software se está reduciendo significativamente debido al aumento de la competitividad del mercado. Este hecho acarrea la consecuencia de realizar diseños pobres, testeo inadecuado, y por lo tanto errores en el software que se manifiestan como vulnerabilidades de seguridad.

Antes, los intrusos necesitaban de un conocimiento más profundo de las redes y las computadoras para poder lanzar sus ataques. Desgraciadamente, gracias al incremento del conocimiento sobre el funcionamiento de los sistemas, los intrusos están cada vez más preparados y lo que antes estaba accesible para sólo unos pocos (expertos), hoy en día cualquiera tiene herramientas accesibles con las que poder determinar las debilidades de los sistemas y explotarlas con el fin de obtener los privilegios necesarios para realizar cualquier acción dañina.

Además de la problemática del software y la facilidad de uso de herramientas, los problemas de configuración de muchos dispositivos son otro gran problema. Esto ocurre debido a la falta de conocimiento especializado que los administradores de redes tienen hoy en día, la falta de recursos para instalar los debidos parches de seguridad, etc. Se ve clara la necesidad de concienciación y enseñanza en torno a la seguridad informática.

2.1.5 Clasificación de ataques informáticos e intrusiones

Uno de los primeros trabajos dedicados a categorizar diferentes aspectos de la seguridad informática, se centraba en la debilidad de los sistemas informáticos y defectos de diseño en sistemas operativos así como en vulnerabilidades funcionales y métodos de abusos informáticos. Varias de las taxonomías desarrolladas más tarde se enfocaban principalmente en dos aspectos: Categorización de uso indebido de computadoras, y categorización de la gente que intentaba obtener acceso no autorizado a ordenadores [LINDQVIST00].

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 cuando su clasificación se basa en la estrategia del análisis:

- *Detección de intrusión por anomalías (Anomaly Detection)*

- *Detección de intrusión por mal uso (Misuse)*.

La primera se refiere a intrusiones que pueden descubrirse basadas en la conducta anómala y uso de recursos de computación. Por ejemplo, si el usuario X 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 anómala 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, 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.

2.1.6 Detección de intrusión por anomalías

Se puede definir *anomalía* como la existencia de una discrepancia de una regla o de un uso. Así pues, para poder detectar dicha discrepancia se tiene que definir primero lo que se considera como un comportamiento normal de un sistema. Una vez definido esto, se realiza una clasificación como sospechosa o intrusiva a aquellos comportamientos que se desvíen de lo que se considera como normal.

La detección de anomalías se basa en la suposición de que el comportamiento que tienen los usuarios y la red es lo suficientemente regular como para sacar un patrón, de forma que cualquier desviación significante pueda ser considerada como una evidencia de una intrusión en el sistema.

Este tipo de enfoque fue propuesto por Dorothy Denning en su artículo: *An intrusión detection model*. En él proponía la creación de perfiles de uso de los sistemas durante un determinado período de tiempo. La detección de anomalías es una aproximación que se basa en el aprendizaje de actividades normales y legítimas del sistema con el sistema de detección de intrusos entrenado para detectar actividades que se desvían del comportamiento normal.

La detección de anomalías tiene la desventaja que depende de la calidad del proceso de aprendizaje: Un entrenamiento demasiado restringido podría causar un sistema con un alto porcentaje de falsos positivos, mientras que un entrenamiento demasiado general, puede producir un alto porcentaje de falsos negativos. Adicionalmente, estos tipos de sistemas de detección de intrusión pueden ser engañados por alguien que conozca bien la red, ya que podrían usar protocolos usados en la propia red y no activarían este tipo de sistemas.

2.1.7 Técnicas de detección de anomalías

Existen diferentes técnicas para realizar la detección de anomalías en un sistema. Para ello, se hace uso de mecanismos heurísticos y estadísticos para realizar adaptaciones al comportamiento del objeto que se está estudiando, además de detectar cambios imprevistos.

- *Sistemas basados en conocimiento*

Los sistemas basados en conocimiento o sistemas expertos fueron los que en un principio se usaron más en la elaboración de los sistemas de detección de intrusos. El primer sistema experto desarrollado para este fin fue el IDES (Sistema Integrado de Desarrollo) [DENNING86] . Otro modelo, el de Denning se basa en la siguiente hipótesis: las violaciones de seguridad se pueden detectar mediante la monitorización de los registros de auditoría del sistema para buscar patrones anormales de uso, es decir, se utilizan reglas para adquirir conocimiento a partir de dichos registros.

- Sistemas basados en métodos estadísticos

Muchas técnicas se apoyan en métodos estadísticos para representar el comportamiento de los sujetos en base a métricas y modelos, y reglas para la obtención de conocimiento de los registros de auditorías. Pues bien, un componente básico de dichos sistemas es el de los perfiles de actividad, que son los encargados de caracterizar el comportamiento de un sujeto (normalmente usuarios), con respecto a un objeto (ficheros, programas, registros,…).

- Sistemas basados en aprendizaje automático

Los sistemas de aprendizaje automático son los más estudiados para el modelado de comportamientos normales. Existe una gran diversidad de modelos de aprendizaje automático [ZUR04], por ello se ha tratado de localizar el modelo que mejores resultados presente en cuanto a detección, reducción de falsos positivos y tiempo de computación.

2.1.8 Detección de Intrusión por Mal Uso o Uso Indebido

Los sistemas de detección basados en uso indebido monitorizan las actividades que ocurren en un sistema y las compara con una serie de firmas de ataques previamente almacenadas en una base de datos. Cuando se monitorizan actividades que coinciden con las firmas se genera una alarma. Este tipo de análisis se atiene al conocimiento previo de las secuencias y actividades que forman un ataque. Se detectan las tentativas de explotación de vulnerabilidades conocidas o patrones de ataques típicos. Es la estrategia más utilizada por los sistemas de detección de intrusos comerciales.

Un sistema de detección de uso indebido contiene dos componentes principales:

- Un lenguaje o modelo para describir y representar las técnicas utilizadas por los atacantes.

- Programas de monitorización para detectar la presencia de un ataque basado en las representaciones o descripciones dadas.

Constituye el enfoque más tradicional y actualmente sigue siendo el más extendido. Estos sistemas tienen conocimientos específicos sobre determinados ataques y se basan en ellos para analizar los datos de entrada del sistema. Si alguno de los datos coincide con alguno de los patrones conocidos se emite una alerta.

Una de las ventajas de los sistemas de detección basados en este tipo de análisis es la fidedigna detección de patrones de ataques conocidos. Por otra parte, presenta desventajas como tener que conocer a priori el patrón de ataque, lo que provoca que nuevas intrusiones pasen desapercibidas por el detector, o la facilidad con la que se podría engañar al sistema con pequeñas variaciones de los patrones de ataque conocidos.

2.1.9 Técnicas de Detección de Uso Indebido

Los métodos basados en conocimiento van comprobando los eventos que tienen lugar en los Host o redes para buscar reglas o patrones de ataques ya definidos. Se emplean representaciones de ataques ya conocidos para controlar su posible ocurrencia. Para representar los ataques se hace uso de los llamados sistemas expertos, firmas de ataques, transiciones de estados y redes de Petri.

- *Sistemas Expertos*

Los sistemas expertos tienen el conocimiento codificado mediante reglas de implicación (condición-acción) de tipo "if-then-else". Si se cumplen todas las condiciones se aplica la regla. El motor de inferencia se encarga de detectar si ha ocurrido una intrusión basándose en las reglas y los hechos. Una de las ventajas que presenta esta técnica es la separación de la lógica de control sobre el dominio del problema, pero presenta la desventaja de que las reglas no son secuenciales, lo que dificulta aislar pasos de intrusiones basadas en el tiempo [ZUR04].

- *Detección de firmas*

A este tipo de técnica también se la conoce como Sistema de Razonamiento Basado en Modelos, su funcionamiento consiste en observar la ocurrencia de patrones de cadenas que puedan ser sospechosas. Para ello realiza comparaciones de los eventos que ocurren en el sistema, con los patrones o firmas almacenadas en una base de datos de ataques, en busca de similitudes. Su principal desventaja es la necesidad de desarrollar e incorporar a la base de datos una firma nueva por cada nuevo tipo de ataque o vulnerabilidad descubierta.

- *Análisis de transacción de estados*

Se basan en una máquina de estados finitos. Los ataques se representan como una secuencia de transiciones que caracterizan el estado de seguridad de un sistema. Cuando se alcanza un estado considerado como intrusión se lanza una alarma.

2.1.10 Problemas de identificación

Como en todo sistema, puede presentar problemas durante su operación. Estos pueden ser clasificados en: Falsos positivos, falsos negativos y errores inducidos.

- **Falsos Positivos**

Ocurre cuando la herramienta clasifica una acción como una posible intrusión cuando se trata de un comportamiento legítimo, que no constituye una violación de seguridad.

- **Falso negativo**

Sucede cuando una intrusión real acontece, pero la herramienta permite que pase como si fuese una acción legítima.

- **Error inducido**

Ocurre cuando un intruso modifica la operación del SDI para forzar la ocurrencia de falsos negativos. Esto sucede generalmente cuando el atacante logra introducirse en el host en el que opera el detector de intrusos.

2.2 Lógica Difusa

2.2.1 Introducción

Una de las disciplinas matemáticas con mayor número de seguidores actualmente es la llamada lógica difusa o borrosa, que es la lógica que utiliza expresiones que no son ni totalmente ciertas ni completamente falsas, es decir es la lógica aplicada a conceptos que pueden tomar un valor cualquiera de veracidad dentro de un conjunto de valores que oscilan entre dos extremos, la verdad absoluta y la falsedad total.

Conviene recalcar que lo que es difuso, borroso, impreciso o vago no es la lógica en sí, sino el objeto que estudia, por que expresa la falta de definición del concepto al que se aplica. La lógica difusa permite tratar información imprecisa, como estatura media o temperatura baja, en términos de conjuntos difusos que se combinan en reglas para definir acciones: Si la temperatura es alta entonces enfriar mucho. De esta manera, los sistemas de control basados en lógica difusa combinan variables de entrada, definidas en términos de conjuntos difusos, por medio de grupos de reglas que producen uno o varios valores de salida. [CHOQUE02]

2.2.2 Historia

La lógica difusa fue investigada por primera vez alrededor de mediados de los años sesenta por el ingeniero Lotfy A. Zadeh en la Universidad de Berkeley (California). En un principio este ingeniero no denominó a esta lógica como lógica difusa sino que la llamó principio de incompatibilidad. A continuación se muestra como describió él este principio: "Conforme la complejidad de un sistema aumenta, nuestra capacidad para ser

precisos y construir instrucciones sobre su comportamiento disminuye hasta el umbral más allá del cual, la precisión y el significado son características excluyentes".

En este momento fue cuando introdujo el concepto de conjunto difuso (en inglés *Fuzzy Set*). Este nuevo concepto no es más que la idea de que los elementos sobre los que se basa el pensamiento humano no son números sino etiquetas lingüísticas. Esta idea es la que permite que se pueda representar el conocimiento, que es principalmente lingüístico de tipo cualitativo y no tanto cuantitativo, en un lenguaje matemático mediante los conjuntos difusos y funciones características asociadas a ello. Esto no quiere decir que exclusivamente se trabaje con números, este lenguaje nos permite trabajar con datos numéricos pero también con términos lingüísticos que aunque son más imprecisos que los números, muchas veces son más fáciles de entender para el razonamiento humano.

Figura 2.2 *Lotfy A. Zadeh*

Fuente: *Tutorial de Logica Difusa y sus Aplicaciones [ZADEH65]*

La lógica difusa es hoy en día más conocida gracias a Lotfy Zadeh, la idea que se esconde detrás de este término tiene sus orígenes hace unos 2500 años atrás puesto que los filósofos griegos ya trabajaban con la idea de que existían distintos grados de veracidad y de falsedad.

Volviendo a la idea originada por Zadeh, aunque se considera que él fue quien primero habló de la lógica difusa, su tesis se basa también en estudios y obras de otros pensadores de otras disciplinas que tenían una visión alejada de la lógica tradicional y muy similar a la de Zadeh. Entre las obras y personas que influyeron a Zadeh, podemos destacar: la paradoja del conjunto de Russell, el principio de incertidumbre de Heisenberg y a Jack Lukasiewicz creador de la lógica multivaluada.

En un principio, la comunidad científica no vio con buenos ojos la lógica difusa, sin embargo algunos de estos investigadores que en un principio habían mostrado su resistencia ante este concepto, terminaron siendo seguidores de Zadeh e incluso mientras él seguía asentando los conocimientos de la lógica difusa, estos científicos se dedicaron a explorar nuevas teorías referidas a este tipo de lógica. Entre estos nuevos seguidores de la lógica difusa podemos destacar a Bellman, Lakoff, Goguen, Smith…

Otro paso importante para el desarrollo de la lógica difusa fue que a principios de la década de los setenta se crearon varios grupos de investigación en diferentes universidades japonesas hicieron grandes contribuciones sobre las aplicaciones que podía tener este tipo de lógica. De esta forma se consiguió crear el primer controlador difuso para una máquina de vapor o crear un controlador de inyección química en depuradoras de agua.

En décadas posteriores esta teoría cada vez fue teniendo más éxito y se le iban encontrando nuevas aplicaciones. En la década de los ochenta, la investigación se orientó hacia las redes neuronales y su similitud con los sistemas fuzzy. Estos sistemas fuzzy lo que hacen es utilizar métodos de aprendizaje basados en redes neuronales para identificar y optimizar sus parámetros. En cuanto a la década de los noventa, a parte de la investigación de las redes neuronales y los sistemas fuzzy, surgen los algoritmos genéticos. Si combinamos estas tres técnicas computaciones, se puede conseguir una herramienta de trabajo muy potente de los sistemas de control.

Según lo expuesto hasta ahora, se puede ver que la lógica difusa ha provocado innumerables investigaciones y aplicaciones, la mayoría orientadas a sistemas de control pero actualmente se está yendo más allá y se empieza a investigar en áreas como el reconocimiento de patrones visuales o la identificación de segmentos de ADN. Por último, mencionar que muchos de los investigadores que actualmente investigan en los temas de la lógica difusa, comentan que el futuro de Internet (en cuanto a controlar la red, gestionarla o recuperar información), está en aplicar las tecnologías difusas en estas áreas.

Figura 2.3 *Lógica difusa Vs. Lógica clásica*

Fuente: *Fuzzy sets, Information & Control [ZADEH65]*

2.2.3 Teoría de Conjuntos Difusos

La lógica difusa permite tratar con información que no es exacta o con un alto grado de imprecisión a diferencia de la lógica convencional la cual trabaja con información precisa. El problema principal surge de la poca capacidad de expresión de la lógica clásica.

2.2.4 Conjuntos Clásicos

Los conjuntos clásicos surgen por la necesidad del ser humano de clasificar objetos y conceptos. Estos conjuntos pueden definirse como un conjunto bien definido de elementos o mediante una función de pertenencia μ que toma valores de 0 ó 1 de un universo en discurso para todos los elementos que pueden o no pertenecer al conjunto. Un conjunto clásico se puede definir con la siguiente función de pertenencia (1).

$$\mu_A(x) = \begin{cases} 0 & si\ x \notin A \\ & \\ I & si\ x \in A \end{cases} \qquad (1)$$

2.2.5 Conjuntos Difusos

En los conjuntos difusos relajamos la restricción de que la función de pertenencia valga 0 ó 1, y dejamos que tome valores en el intervalo [0,1].

La necesidad de trabajar con conjuntos difusos surge de un hecho: Hay conceptos que no tienen límites claros. Algunas definiciones útiles:

- *Variable lingüística* aquella noción o concepto que se califica de forma difusa. Por ejemplo: La altura, la edad, el error, la variación del error, etc. Se le aplica el adjetivo "lingüística" porque define sus características mediante el lenguaje hablado.

- *Universo de discurso* el rango de valores que pueden tomar los elementos que poseen la propiedad expresada por la variable lingüística. En el caso de la variable lingüística 'altura de una persona normal', sería el conjunto de valores comprendido entre 1.4 y 2.3 m.

- *Valor lingüístico* las diferentes clasificaciones que se efectúa sobre la variable lingüística. En el caso de la altura, se podría dividir el universo de discurso en diferentes valores lingüísticos, por ejemplo: *Bajo*, *mediano* y *alto*.

- *Conjunto difuso* un valor lingüístico junto a una función de pertenencia. El valor lingüístico es el "nombre" del conjunto, y la función de pertenencia se define como aquella aplicación que asocia a cada elemento del universo de discurso el grado con que pertenece al conjunto difuso. Se dice que un conjunto es *nítido* si su función de pertenencia toma valores en {0,1}, y *difuso* si toma valores en [0,1].

Un conjunto difuso en un universo en discurso puede definirse como:

$$A = \{(x, \mu_A(x)) \mid x \in U\} \qquad (2)$$

Donde $\mu_A(x)$ es la función de pertenencia de la variable x, y U es el universo en discurso. Cuando más cerca este la pertenencia del conjunto A al valor de 1, mayor será la pertenencia de la variable x al conjunto A, esto se puede ver en la figura 2.3.

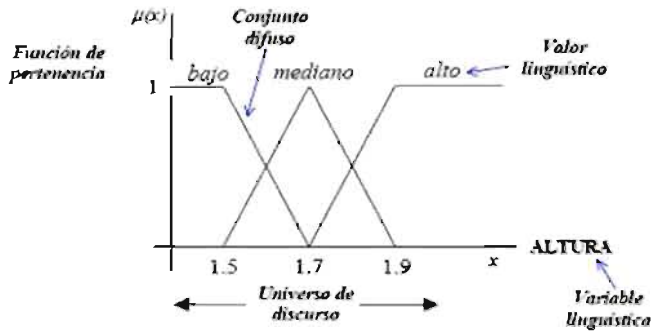


Figura 2.4 *Función de pertenencia*

Fuente: *Fuzzy sets, Information & Control [ZADEH65]*

2.2.6 Operaciones con conjuntos difusos

Se presentan algunos modelos matemáticos para realizar las operaciones de intersección, unión y complemento de los conjuntos difusos.

2.2.7 Extensión de las operaciones clásicas

De la misma manera que se realizan operaciones con los conjuntos clásicos, se definen operaciones para los conjuntos difusos.

- **Intersección de conjuntos**

El primer problema que se plantea es la obtención de la intersección de dos conjuntos difusos. Para ello, se observa que ocurre en el caso clásico. Dados dos subconjuntos

clásicos P y Q del universo X, un elemento x pertenece a la intersección $P \cap Q$, si y sólo si x pertenece a P y x pertenece a Q.

Se toman las respectivas funciones características como:

$$\varphi P(x) = \begin{cases} 1 & si\ x\ pertenece\ a\ P \\ 0 & si\ x\ no\ pertenece\ a\ P \end{cases}$$

$$\varphi Q(x) = \begin{cases} 1 & si\ x\ pertenece\ a\ Q \\ 0 & si\ x\ no\ pertenece\ a\ Q \end{cases}$$

La función característica de la intersección quedará:

$$\varphi_{P \cap Q}(x) = \{\ 1 \quad si \quad \varphi_P(x) = 1 \ \ y \ \ \varphi_Q(x) = 1\ \}$$

O lo que es lo mismo,

$$\varphi_{P \cap Q}(X) = Min\ (\varphi_P(x), \varphi_Q(x))$$

Sin embargo, en el caso de los conjuntos difusos, la definición del conjunto intersección no es tan trivial. Dadas las funciones de pertenencia $\mu_P: X \rightarrow [0,1]$ y $\mu_Q: X \rightarrow [0,1]$ la pregunta es: Si un elemento pertenece a P en un cierto grado entre 0 y 1 $(\mu_P(X))$ y a Q en otro grado entre 0 y 1 $(\mu_Q(X))$. Se toma como modelo el caso clásico, una primera forma de definir la intersección de dos conjuntos difusos es:

$$\mu_{P \cap Q}(x) = Min\ (\mu_P(x), \mu_Q(x))$$

- **Unión de conjuntos**

En el caso de los conjuntos clásicos, dados dos subconjuntos P y Q del universo X, un elemento x pertenece a la unión de $P \cup Q$, si y sólo si x pertenece a P ó x pertenece a Q. Dadas las respectivas funciones características:

$$\varphi_P(x) = \begin{cases} 1 & si \quad x \text{ pertenece a } P \\ 0 & si \quad x \text{ no pertenece a } P \end{cases}$$

$$\varphi_Q(x) = \begin{cases} 1 & si \quad x \text{ pertenece a } Q \\ 0 & si \quad x \text{ no pertenece a } Q \end{cases}$$

La función característica de la unión será:

$$\varphi_{P\cup Q}(x) = \begin{cases} 1 & si \quad \varphi_P(x) = 1 \text{ ó } \varphi_Q(x) = 1 \\ 0 & en \text{ otro caso} \end{cases}$$

O lo que es lo mismo,

$$\boldsymbol{\varphi_{P\cup Q}(x) = Max(\varphi_{P(x)}, \varphi_{Q(x)})}$$

Igual que en el caso de la intersección de conjuntos difusos la definición de la unión de dos conjuntos no es trivial. Teniendo las funciones de pertenencia $\boldsymbol{\mu_P: X \to [0,1]}$ y $\boldsymbol{\mu_Q: X \to [0,1]}$ y sabiendo que un elemento pertenece a P en un cierto grado 0 y 1 $\boldsymbol{(\mu_P(x))}$, y que pertenece a Q en otro grado $\boldsymbol{(\mu_Q(x))}$, Se observa el modelo del conjunto clásico para definir la unión de dos conjuntos difusos como:

$$\boldsymbol{\mu_{P\cup Q}(x) = Max(\mu_{P(x)}, \mu_{Q(x)})}$$

- **Complemento de un conjunto**

En el caso de los conjuntos clásicos cuando se realiza la operación del complemento de un conjunto ocurre que dado el subconjunto clásico P del universo X, un elemento **x** pertenece al complemento P^c, si y sólo si dicho elemento x no pertenece a P.

La función característica está definida mediante:

$$\varphi\ P^c(x) = \begin{cases} 0 & si\ \varphi_P(x) = 1 \\ 1 & si\ \varphi_p(x) = 0 \end{cases}$$

El complemento de un conjunto borroso no es una operación tan claramente definida como en el caso clásico. Dada la función de pertenencia μ_P: $X \rightarrow [0,1]$, si un elemento pertenece a P en un cierto grado entre 0 y 1 (μ_P (x)), Se realiza una semejanza con los conjuntos clásicos para definir el complemento de un conjunto borroso P, mediante la función de pertenencia:

$$\varphi_{Pc}(x) = 1 - \varphi_p(x)$$

2.2.8 Función de Pertenencia

Aunque en principio cualquier función sería válida para definir conjuntos difusos, en la práctica hay ciertas funciones típicas que siempre se suelen usar, tanto por la facilidad de computación que su uso conlleva como por su estructura lógica para definir su valor lingüístico asociado. Las funciones más comunes son:

- Función Gaussiana

$$\mu_A(x) = e^{\frac{-(x-m)^2}{2\sigma^2}}$$

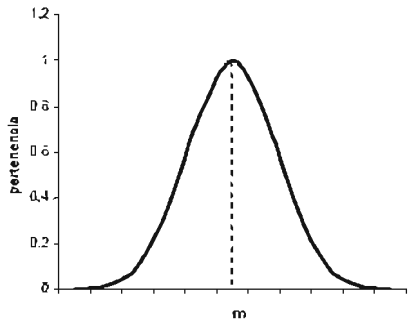
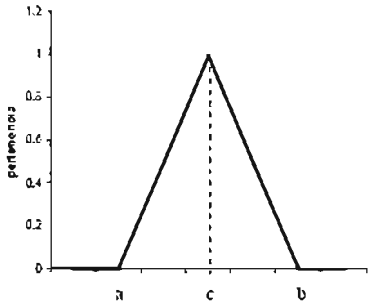


Figura 2.5 *Función de Pertenencia Gaussiana*

Fuente: *Inteligencia Artificial [CHOQUE02]*

- Función Lambda o Triangular

$$\mu_A(x) = \begin{cases} 0 & \text{si } x \le a \\ \dfrac{x-a}{c-a} & \text{si } x \in (a,c] \\ \dfrac{b-x}{b-c} & \text{si } x \in [c,b) \\ 0 & \text{si } x \ge b \end{cases}$$



-

Figura 2.6 *Función de Pertenencia Triangular*

Fuente: *Inteligencia Artificial [CHOQUE02]*

- Función PI o trapezoidal

$$\mu_A(x) = \begin{cases} 0 & \text{si } x \le a \\ \dfrac{x-a}{b-a} & \text{si } x \in (a,b) \\ 1 & \text{si } x \in [b,c] \\ \dfrac{d-x}{d-c} & \text{si } x \in (c,d) \\ 0 & \text{si } x \ge d \end{cases}$$

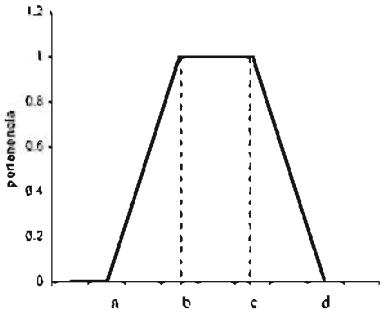


Figura 2.7 *Función de Pertenencia Trapezoidal*

Fuente: *Inteligencia Artificial [CHOQUE02]*

2.2.9 Características Esenciales

La lógica difusa es la lógica fundamental de los modos de razonamiento que son aproximados antes que exactos. Algunas de las características esenciales de la lógica difusa se relacionan con lo siguiente:

a) En la lógica difusa el razonamiento exacto es considerado como un caso limite del razonamiento aproximado.

b) En la lógica difusa todo es materia de grados.

c) Cualquier sistema lógico puede ser llevado a términos difusos.

d) En la lógica difusa el conocimiento está interpretado como una colección de restricciones elásticas.

e) La inferencia es considerada como un proceso de propagación de restricciones elásticas.

2.2.10 Diferencias Respecto a Sistemas Tradicionales

La lógica difusa difiere de los sistemas tradicionales de lógica tanto en espíritu como en detalle. Algunas de las principales diferencias son las siguientes:

- **Valores de Verdad**

En los sistemas de lógica bivalente, los valores de verdad solamente pueden tener dos valores: verdadero o falso. En los sistemas multivaluados, los valores de verdad de una proposición pueden ser elementos de un conjunto finito, un intervalo tal como [0,1] o como un algebra booleana.

En la lógica difusa los valores de verdad de una proposición pueden ser subconjuntos difusos de cualquier conjunto parcialmente ordenado pero usualmente este es un subconjunto difuso del intervalo [0,1] o simplemente un punto en ese intervalo. Los denominados valores de verdad lingüísticos expresados como: verdadero, demasiado verdadero, completamente verdadero, etc. Son interpretados como etiquetas de subconjuntos difusos del intervalo unitario.

- **Predicados**

En los sistemas bivalentes, los predicados son precisos por decir, mortal, aun, mas grande que. En la lógica difusa los predicados son difusos, por decir: alto, malo, gordo, veloz, mucho grande que. Se puede decir que la mayoría de los predicados en un lenguaje natural son difusos antes que precisos.

- **Modificadores de Predicados**

En los sistemas clásicos, el único modificador de predicados ampliamente utilizado es la negación. En la lógica difusa existe una variedad de modificadores de predicados tales como muy, más o menos, completamente antes, extremadamente. Tales modificadores de predicados juegan un rol esencial en la generación de los valores de una variable lingüística tales como: Muy joven, no muy joven, más o menos joven, etc.

- **Cuantificadores**

En los sistemas lógicos clásicos existen solamente dos cuantificadores: universal y existencial. La lógica difusa admite, en adición, una variedad amplia de cuantificadores tales como: Pocos, varios, usualmente, mas que, casi siempre, frecuentemente, etc. En la lógica difusa un cuantificador difuso en interpretado como un numero difuso o una proporción difusa.

- **Probabilidades**

En los sistemas de lógica clásica, la probabilidad es numérica o de valor en intervalos. En la lógica difusa se tiene la opción adicional de emplear lingüísticos, o de manera general, las probabilidades difusas ejemplificadas por probable, improbable, muy probable, alrededor de 0.7, grandemente probable. Tales probabilidades pueden ser interpretadas como números difusos que pueden ser manipulados mediante el uso de la aritmética difusa.

- **Posibilidades**

El concepto de posibilidad en la lógica difusa es graduada, antes que bivalente. Además, como en el caso de las probabilidades, las posibilidades pueden ser tratadas como variables lingüísticas con valores tales como: Posible, completamente posible, casi posible, etc. Tales valores pueden ser interpretados como etiquetas de subconjuntos difusos de la línea real.

2.2.11 Modelo Difuso

El modelado consiste en obtener una representación matemática de un fenómeno o un proceso, con la cual se pueda analizar sus propiedades dinámicas y predecir su comportamiento futuro y de la cual se pueda sintetizar un controlador o regulador para conseguir llevarlo a diferentes puntos de operación de forma segura, con un comportamiento dinámico deseado según requerimientos de operación.

La obtención de un modelo permite, mediante simulaciones, realizar y repetir experimentos que con el proceso real resultarían caros, lentos o peligrosos, generando conocimiento útil para el rediseño del proceso mismo o de nuevas estrategias de control; posibilidades que a su vez permitirían un mejor uso de materiales y energía. [OLIVARES03]

Un modelo difuso es una aplicación muy útil de los conjuntos difusos. Su objetivo es construir un modelo para un determinado sistema con las siguientes características:

- Operar a nivel de términos lingüísticos (Conjuntos difusos)
- Representar y procesar incertidumbre

Es importante destacar que la definición de las etiquetas lingüísticas afectá mucho al procesamiento que efectúe el modelo, la colección de conjuntos difusos debe elegirse cuidadosamente.

2.2.12 Fases del modelado de Sistemas

El desarrollo de un modelo tiene las siguientes fases principales:

- *Preprocesamiento:* Especificación de las variables de entrada y salida, además del estudio del conocimiento relevante.

- *Estimación de Parámetros:* Se eligen los parámetros del sistema usando alguna técnica de optimización.

- *Verificación del Modelo:* Se verifica su funcionamiento según los datos disponibles y se cuantifica el error producido.

- *Validación del Modelo*: Se trata de asegurar que el modelo es válido, soluciona los problemas planteados y se comporta como el usuario esperaba. [GALINDO03]

Un buen modelo para predicción a corto plazo, puede fallar si se intenta efectuar una predicción a largo plazo. A continuación en la figura se muestra la arquitectura de un modelo difuso.

Figura 2.8 *Arquitectura General de un Modelo Difuso*

Fuente: *Conjuntos y Sistemas Difusos [GALINDO03]*

2.2.13 Sistemas difusos

Cualquier sistema dinámico o estático que hace uso de la lógica difusa o de conjuntos difusos, se considera como un sistema difuso. Los sistemas difusos se pueden emplear el modelado y control de sistemas. Específicamente, en modelado el sistema se puede describir mediante una relación difusa o un conjunto de reglas proposicionales difusas (Cualitativas), que pueden tener distintas formas. Una proposición difusa se basa en la utilización de términos cualitativos *A*, asociados a una variable *x* definida dentro de un universo acotado [OLIVARES 03].

Principalmente, existen dos tipos de sistemas difusos, conocidos como Mandani y Takagi-Sugeno de acuerdo con la forma del consecuente de cada regla:

$$r^l: si\ x\ es\ A_i^l \quad entonces \quad y\ es\ B_j^l \qquad 1 \le i \le n_x, \qquad 1 \le j \le n_y \qquad (1)$$

$$r^l: si\ x\ es\ A_i^l \quad entonces \quad y \equiv f_j(x) \qquad 1 \le i \le n_x, \qquad 1 \le j \le n_y \qquad (2)$$

Un conjunto de reglas r^l forma una base de reglas. En (1) se representa un sistema difuso con antecedentes y consecuentes proposicionales en cambio en (2) se representa un sistema con entradas proposicionales y salidas precisas descritas mediante funciones estáticas o dinámicas.

Existen diferentes métodos de inferencia del resultado de cada regla, del tipo "*Modus Ponens Generalizado*", en conjunto con métodos de "conjunción" de proposiciones (para el caso de sistemas difusos con más de una entrada) y de "agregación" de los resultados individuales, para determinar el valor difuso o preciso de la salida del sistema completo, para un determinado valor de las entradas [OLIVARES 03]. Existen también, diversos métodos para convertir valores precisos en difusos y viceversa, denominados métodos de *"fuzificación"* y de "*desfuzificación*"[5] respectivamente.

[5] Algunos autores lo llaman clarificación o difuminación.

2.2.14 Metodología de desarrollo de sistemas difusos

Los modelos difusos empleados en el control de procesos tienden a seguir la misma metodología empleada en el diseño de sistemas de control clásico, esto es: En primer lugar el diseño conceptual es hecho en papel una vez que se ha entendido tanto la mecánica del comportamiento del sistema como su dinámica en términos de entrada/salida. Acto seguido se procede a un ciclo de modelado y simulación y así sucesivamente hasta obtener el resultado deseado [TSUKAMOTO 03].

Para los sistemas difusos, como se muestra en la figura, el método de diseño se efectúa de acuerdo con el siguiente ciclo.

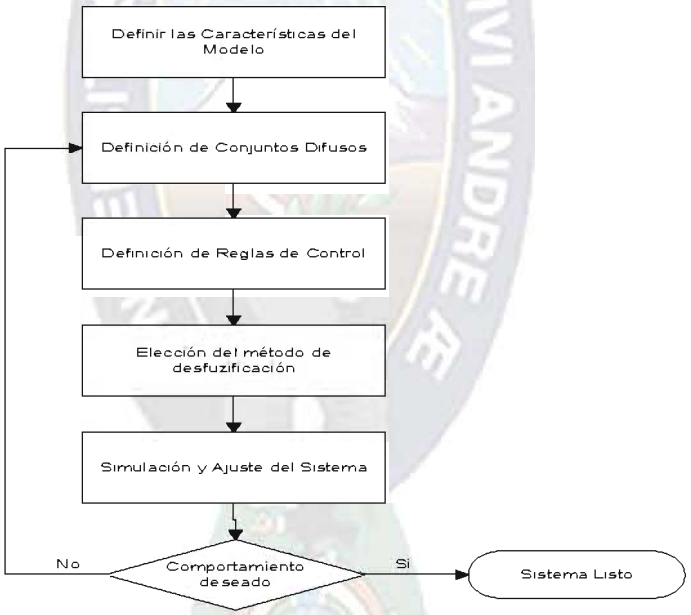


Figura 2.9 *Ciclo de Metodología de Diseño*

Fuente: *Fuzzy Tutorial [TSUKAMOTO03]*

2.2.15 Método de diseño

Aunque existen diferentes métodos, el presente trabajo adoptara el método de diseño de Tsukamoto, para desarrollar modelos difusos. El cual contempla las siguientes etapas:

- *Etapa 1:* Definición de características del modelo
- *Etapa 2:* Definición de conjuntos difusos
- *Etapa 3:* Definición de reglas de control
- *Etapa 4:* Selección del método de desdifuzificación
- *Etapa 5:* Simulación y ajuste del sistema. [TSUKAMOTO03]

El diseño del modelo difuso se realiza tomando en cuenta el conocimiento de un experto, o mediante algún mecanismo de optimización para el ajuste de sus parámetros.

✍ **Definición de Características del Modelo**

En esta etapa se definen las siguientes características funcionales y operacionales del modelo:

- Los datos de entrada al modelo
- Las transformaciones básicas que se aplicaran a los datos
- Los datos de salida del modelo

✍ **Definición de Conjuntos Difusos**

Para el proceso de definición de los conjuntos difusos en sistemas de control, primero se identifican y nombran las variables de entrada y de salida y se establecen sus rangos.

Algunas recomendaciones citadas para la definición de los conjuntos difusos sugeridas por Tsukamoto, son las siguientes:

- El número de términos difusos (conjuntos) asociados a cada variable debe ser generalmente un número impar entre 5 y 9.

- Para producir una acción de control suave, cada conjunto debe traslaparse un poco sobre los otros conjuntos vecinos.

- La densidad de los conjuntos difusos debe ser mayor alrededor del punto óptimo de control del sistema y menor conforme aumenta la distancia a ese punto.

✍ **Definición de Reglas de Control**

Las reglas de control engloban el conocimiento del sistema y los objetivos se control. Cada regla tiene un estado del sistema en su premisa y una acción de control sugerida en su conclusión. Las reglas de control difuso conectan los valores de entrada con las propiedades de la salida del modelo. Están expresadas como proposiciones condicionales:

Si **[*Estado del Proceso*]** ***entoces*** **[*Accion de Control*]**

Donde estado del proceso y acción de control es una proposición o un grupo de proposiciones ligadas por un conectivo. Como ejemplo de regla de control se tiene:

Si *posibilidad de investigación es media y posibilidad de ataque es excesiva y posibilidad de aprovechamiento es alta* **entonces** *la posibilidad de intrusión es segura.*

Las reglas de control difusas son declarativas y no secuenciales, lo que significa que el orden en que se expresan no es importante. El número de reglas que requiere un modelo o controlador difuso se halla multiplicando el número de términos difusos de las variables de entrada. De esta manera las reglas cubrirán todas las posibles combinaciones provenientes de las distintas entradas.

✍ **Selección del Método de Desdifuzificación**

En esta etapa se produce la acción de control no difusa, que es representada por la función de membrecía de la acción de control. Se debe seleccionar el método de

desdifuzificacion que más se ajuste al sistema. Algunos de los métodos más utilizados son los siguientes:

- Método de la Media de Máximo
- Método de Tsukamoto
- Método del Centro de Area o Centroide
- Método del Criterio Máximo

✍ **Simulación y Ajuste del Sistema**

Cuando el modelo difuso ha sido construido, el proceso de simulación y desarrollo del prototipo comienza. En lo que se refiere a la simulación, esta puede realizarse en varios paquetes computacionales disponibles en el mercado. Generalmente estos paquetes incluyen herramientas para evaluar el modelo difuso y aislar problemas en los conjuntos difusos o en la base de reglas.

Cuando los resultados de la simulación o las pruebas no son satisfactorias se realizan ajustes en las descripciones de los conjuntos difusos o en las acciones de control sugeridas por las reglas, hasta afinar el desempeño del control.

2.2.16 Aplicaciones

La Lógica difusa tiene gran utilidad ya que permite tratar problemas demasiado complejos, mal definidos o para los cuales no existen modelos matemáticos precisos. Gracias a este tipo de lógica se ha permitido modelar y resolver situaciones consideradas intratables desde el punto de vista de la Lógica Clásica.

En los últimos años la lógica difusa se ha utilizado en distintos tipos de instrumentos, máquinas y en diversos ámbitos de la vida cotidiana. Algunos casos por ejemplo son los estabilizadores de imágenes en grabadoras de vídeo, controladores de ascensores e ingeniería de terremotos. También se ha usado esta técnica en la industria, obteniéndose

excelentes resultados como en el caso del metro de Sendai en Japón, ya que permitía que el metro arrancara y frenara con gran suavidad, sin producir alteraciones entre los pasajeros.

Realizando una división de los ejemplos en tres grandes grupos se tiene:

- *Productos creados para el consumidor:* Lavadoras difusas (Matsuhita Electronic Industrial), hornos microondas, sistemas térmicos, traductores lingüísticos, cámaras de vídeo, televisores, estabilizadores de imágenes digitales (Matsuhita) y sistemas de foco automático en cámaras fotográficas.

- *Sistemas:* Elevadores, trenes, automóviles (caso de los sistemas de transmisiones, de frenos y mejora de la eficiencia del uso de combustible en motores), controles de tráfico, sistemas de control de acondicionadores de aire que evitan las oscilaciones de temperatura y sistemas de reconocimiento de escritura.

- *Software:* Diagnóstico médico, seguridad, comprensión de datos, tecnología informática y bases de datos difusas para almacenar y consultar información imprecisa (uso del lenguaje FSQL).

2.3 MATLAB

MATLAB (abreviatura de *MATrix LABoratory*, "laboratorio de matrices") es un software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows y Apple Mac OS X.

Entre sus prestaciones básicas se hallan: La manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros

dispositivos hardware. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI). Además, se pueden ampliar las capacidades de MATLAB con las *cajas de herramientas* (*toolboxes*); y las de Simulink con los *paquetes de bloques* (*blocksets*).

Es un software muy usado en universidades y centros de investigación y desarrollo. En los últimos años ha aumentado el número de prestaciones, como la de programar directamente procesadores digitales de señal o crear código VHDL.

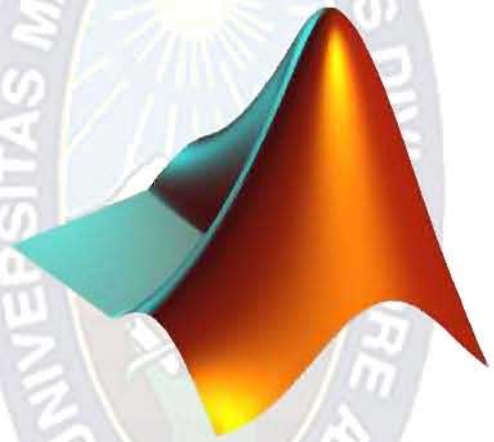


Figura 2.10 *MATLAB (MATrix LABoratory)*

Fuente: *MathWorks www.mathworks.com/products/matlab*

2.3.1 Historia

Fue creado por *Cleve Moler* en 1984, surgiendo la primera versión con la idea de emplear paquetes de subrutinas escritas en Fortran en los cursos de álgebra lineal y análisis numérico, sin necesidad de escribir programas en dicho lenguaje. El lenguaje de programación M fue creado en 1970 para proporcionar un sencillo acceso al software de matrices *LINPACK* y *EISPACK* sin tener que usar Fortran.

En 2004, se estimaba que MATLAB era empleado por más de un millón de personas en ámbitos académicos y empresariales.

2.3.2 Sintaxis

MATLAB es un programa de cálculo numérico orientado a matrices. Por tanto, será más eficiente si se diseñan los algoritmos en términos de matrices y vectores.

- **Ejemplos**

- Este es el tradicional programa Hello World hecho con el lenguaje de MATLAB:

```
>> disp('Hola mundo');       % Muestra el mensaje.
```

Hola mundo

- Otro ejemplo es la serie trigonométrica de tren de pulsos positivos y negativos.

```
n= input('número de sumandos= '); % creamos una serie de pulsos
útil para el procesados de señales y sistemas dentro del mundo de
las telecomunicaciones.
t=-2:.01:2;
pulso=zeros(1,length(t));
for k=1:n
pulso=pulso+sin(2*(2*k-1)*pi*t)/(2*k-1);
end
plot(t,pulso)
grid
```

2.3.3 Cajas de herramientas y paquetes de bloques

Las funcionalidades de Matlab se agrupan en más de 35 cajas de herramientas y paquetes de bloques (para Simulink), clasificadas en las siguientes categorías:

MATLAB (Cajas de herramientas)	Simulink
Matemáticas y Optimización	Modelado de punto fijo
Estadística y Análisis de datos	Modelado basado en eventos
Diseño de sistemas de control y análisis	Modelado físico
Procesado de señal y comunicaciones	Gráficos de simulación
Procesado de imagen	Diseño de sistemas de control y análisis
Pruebas y medidas	Procesado de señal y comunicaciones
Biología computacional	Generación de código
Modelado y análisis financiero	Prototipos de control rápido y SW/HW HIL
Desarrollo de aplicaciones	Tarjetas integradas
Informes y conexión a bases de datos	Verificación, validación y comprobación

Cuadro 2.1 *Caja de Herramientas de MatLab*

Fuente: *MathWorks www.mathworks.com/products/matlab*

2.3.4 Limitaciones y alternativas

Durante mucho tiempo hubo críticas porque MATLAB es un producto propietario de The Mathworks, y los usuarios están sujetos y bloqueados al vendedor. Recientemente se ha proporcionado una herramienta adicional llamada MATLAB Builder bajo la sección de herramientas "Application Deployment" para utilizar funciones MATLAB como archivos de biblioteca que pueden ser usados con ambientes de construcción de aplicación .NET o Java. Pero la desventaja es que el computador donde la aplicación tiene que ser utilizada necesita MCR(MATLAB Component Runtime) para que los archivos MATLAB funcionen correctamente. MCR se puede distribuir libremente con los archivos de biblioteca generados por el compilador MATLAB.

Capítulo III

Modelo para la Detección de Intrusos

RESUMEN

Se desarrollan las características propias que conlleva la construcción de un modelo difuso. Luego se realizan la evaluación del modelo difuso, para esto se hace la implementación en MatLab para simular su funcionamiento, se diseñan los casos de prueba. A través de los resultados obtenidos en las pruebas se evalúa el modelo difuso y se da una conclusión acerca de su rendimiento.

3.1 Introducción

En el presente capítulo se realiza el diseño del modelo para la detección de intrusos con el cual se interpreta la información de forma adecuada filtrando el exceso de datos que pueden hacer a un administrador disminuir la atención que debe de prestar al sistema. Dicho modelo, se basa en la interpretación de información para evaluar a partir de ella la posibilidad de que el usuario en el que se centra el interés tenga intenciones nocivas.

3.2 Definición de las Características del Modelo

Para llevar a cabo tal análisis se ha optado por enfocar este modelo difuso en la constatación de ciertos síntomas que pueden indicar que el presunto atacante se encuentra en alguna de las cinco fases por las que se suele pasar para atacar un sistema:

- **Investigación externa**: Es la menos peligrosa con amplia diferencia. De hecho, no suele estar abordada como tal en los sistemas de detección de intrusos comerciales al uso. Consiste en la recopilación de información del sistema a través de procedimientos no intrusivos, que no deberían de despertar la más mínima sospecha y son complicados de detectar al confundirse entre la gran multitud anónima de usuarios. Incluye entre otras cosas la navegación por las páginas del sistema averiguando información al respecto de sitios, sus usuarios y

administradores, la investigación de las direcciones de que dispone a través de sus DNS's, etc...

- **Investigación interna**: Son otros métodos más de recopilación de información del sistema antes de intentar acceder a él. La diferencia respecto a la fase anterior, es que en este caso la recopilación de información se hace de forma más intrusiva, a través de peticiones que los usuarios comunes raramente realizan y por lo tanto es más fácil de detectar. De la misma manera, la peligrosidad de esta investigación ya puede calificarse de media, ya que puede dejar en manos del atacante información que revele importantes problemas de seguridad del sistema objeto de estudio. Incluye procedimientos como los escaneos de puertos, la verificación de existencias de programas vulnerables instalados o las consultas a servicios que proporcionar datos sobre los usuarios del sistema.

- **Ataque**: Son los intentos de conseguir acceso al sistema de forma ilícita a través de cualquier sistema. Estas actividades implican de forma inequívoca un alto riesgo y ya se incluyen dentro de lo que es el ataque a un sistema propiamente dicho. Incluyen tareas como el intento de ataques de diccionario para acceso a una máquina, el overflow a determinadas aplicaciones o el envío de parámetros con líneas de comando a determinados programas CGI.

- **Borrado de huellas**: Es la eliminación de las pistas que puedan haber quedado en un sistema señalando la intrusión del presunto atacante vigilado. En sí misma no sería peligrosa, pero acompañada de señales de que se ha producido algún intento de los explicados en las fases anteriores indica una seria peligrosidad, ya que confirmaría que el usuario ha conseguido acceder al sistema. Incluye tareas como el borrado de logs, la creación de nuevas cuentas de usuario o la instalación de troyanos.
- **Aprovechamiento**: Consiste en sacar partido del acceso que se ha conseguido al sistema para el provecho del intruso. Es muy peligroso, porque en muchas

ocasiones se realizan tareas altamente nocivas para el sistema. Incluye tareas como la copia de ficheros, el borrado de ficheros o el uso de recursos del sistema.

Combinando estas cinco etapas, se obtiene un análisis suficientemente concreto y desde un punto de vista que normalmente no suele contemplarse en este tipo de herramienta.

3.3 Diseño del Modelo

A objeto de facilitar una comprensión global del sistema de detección de intrusos, una de las primeras actividades a llevar a cabo consiste en la determinación de las relaciones que pueden existir entre las diferentes variables del modelo, es decir, establecer el conocimiento o expertizaje que revela la opinión o estimación de las vinculaciones entre las variables de entrada y de salida.

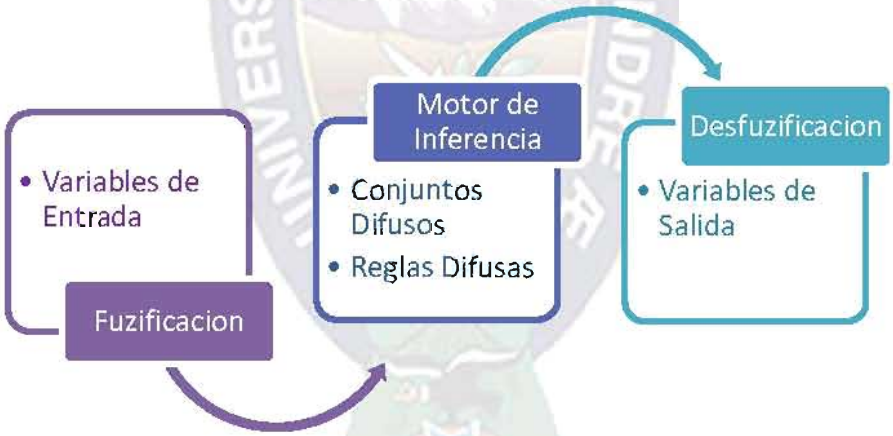


Figura 3.1 *Esquema general del modelo planteado*

Fuente: *Elaboración propia*

3.3.1 Mapa Cognoscitivo de las Variables

Se utiliza la figura 3.2 para mostrar el esquema asociativo entre variables que proporcionarían los datos de entrada y los agrupa en cinco variables intermedias que representan las cinco fases del ataque de un intruso que se han explicado anteriormente. Estas variables son indicativas de las siguientes circunstancias:

- Si se efectuó una investigación externa
- Si se efectuó una investigación interna
- Si se cometió un ataque al sistema
- Si se realizó el borrado huellas que muestren la intrusión
- Si se verifica que haya aprovechamiento de acceso al sistema

Estas variables son agrupadas en una única variable final que define la posibilidad de intrusión del usuario y si se precisa impedirle el acceso al sistema objeto de atención.

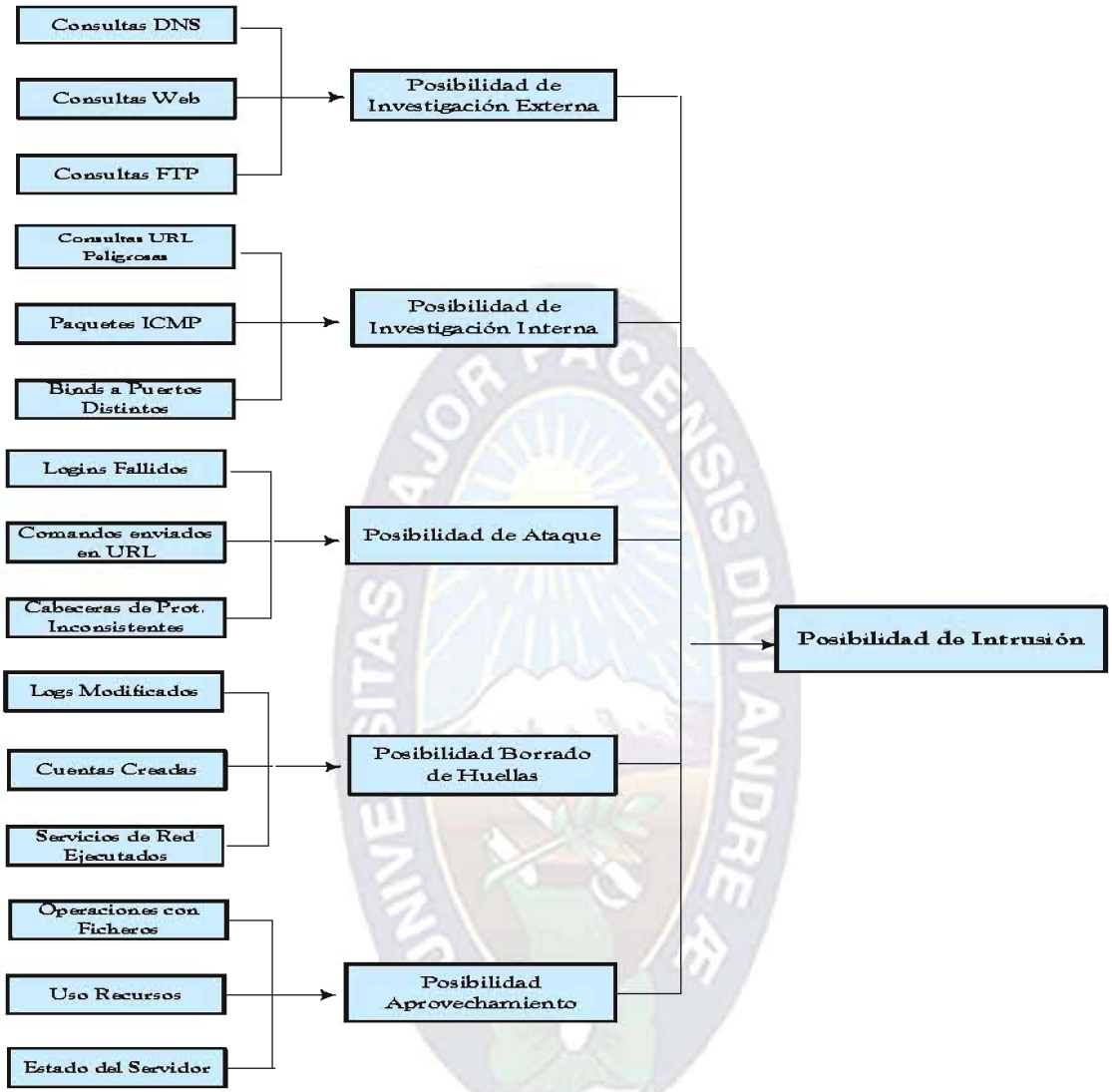


Figura 3.2 *Mapa cognoscitivo de las variables de entrada y salida*

Fuente: *Elaboración propia*

3.4 Etapa de Fuzificación

3.4.1 Variables de entrada

Se han considerado como variables de entrada diversos factores, que se trasladan como entradas al modelo propuesto. Todas las medidas se llevan a cabo dentro de un plazo

temporal limitado que por defecto ha sido establecido una hora y adicionalmente cabe considerar que gozan de "memoria" o "persistencia", esto es, una vez alguien ha hecho uno de los intentos desde una dirección concreta queda registrado el valor de esa variable establecido y no se "borra" más que con los valores de utilización posterior o superiores.

3.4.2 Diseño de los Conjuntos Difusos

A continuación se describen las variables de entrada, con sus correspondientes distribuciones (valores posibles que pueden tomar).

- **Consultas DNS**

Esta variable muestra la posibilidad de que alguien esté intentando recopilar información sobre el sistema a través de consultas aparentemente inofensivas a los servidores DNS que alojan información sobre el sistema protegido. Para esta variable se toman los siguientes aspectos:

a) Intentos de Resolución Inversa

La resolución inversa DNS es el mecanismo del sistema de nombres de Internet por el cual, a partir de una dirección IP, se proporciona el nombre de ese servidor. Este mecanismo de resolución, si bien puede ser utilizado por cualquier usuario, no suele ser habitual, ya que al navegar los usuarios normales únicamente necesitan la resolución directa (conseguir la dirección IP a partir de un nombre). En numerosas ocasiones esta suele ser una de las primeras maneras por las que un intruso comienza a conocer un sistema, intentando averiguar su nombre a partir de su dirección, pues este proporciona en casi todas las ocasiones una pista infalible del servicio que tiene albergado. (www, ftp, chat...).

b) Intentos de "ls" del Dominio

El protocolo estándar utilizado por los servidores DNS incluye una gran cantidad de comandos adicionales a los habitualmente utilizados por los usuarios en su acceso. Entre estos comandos, hay uno que es utilizado entre los servidores DNS maestro – esclavo para sincronizar sus contenidos, que es el comando "ls". Este comando devuelve una lista de todos los registros existentes dentro de un determinado dominio del DNS, lo cual es una información enormemente valiosa para que un hacker comience a investigar las máquinas de las que dispone una corporación. El uso de este comando está habitualmente limitado por los servidores, pero un potencial asaltante podría comprobar si el administrador del sistema DNS ha sido lo suficientemente descuidado como para dejar esta información a la libre disposición de cualquier usuario.

c) *Consultas DNS normales*

El servicio de nombres de dominio (DNS) es el que proporciona, a partir de un nombre de servidor, la dirección IP que le corresponde. Los usuarios habitualmente utilizan el servidor DNS de su red o proveedor para consultar la dirección IP que le corresponde a los sitios a los que se quieren conectar. Por ello un servidor DNS normalmente sólo recibirá peticiones desde el servidor DNS de un proveedor, aunque las hagan distintos usuarios de él. Por otra parte, para los casos normales, para un mismo dominio no es habitual que los clientes tengan que resolver más de dos o tres direcciones IP (el servidor www, el ftp y quizás otro como chat o un webmail). Si se observa que se interroga al servidor DNS objeto de la vigilancia por una cantidad apreciable de nombres de máquinas distintos, puede ocurrir que se encuentre ante un usuario que está intentando averiguar si existen máquinas como aquellas por cuyos nombres pregunta en el dominio para averiguar su IP.

La variable *consultas DNS* indica el número de peticiones de resolución de dominios distintos pertenecientes al rango protegido que se han hecho a dicho DNS (que es el que alberga la resolución inversa) desde la red a la que pertenece la dirección IP vigilada. En este sentido, cabe considerar tres posibles estados de esta variable: Normales, Altas y

Excesivas. Los números borrosos asociados a estas tres etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

CND	Normales	Altos	Excesivos
No menor que	0	20	60
Igual que	0	30	75
Igual que	15	50	85
No mayor que	35	65	100

Cuadro 3.1 *Variable de Entrada CND*

Fuente: *Elaboración Propia*

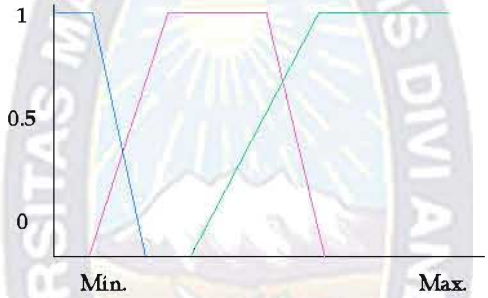


Figura 3.3 *Función de pertenencia variable CND*

Fuente: *Elaboración Propia*

- **Consultas Web**

Esta variable indica la posibilidad de que alguien esté intentando recopilar información sobre el sistema a través de consultas aparentemente inofensivas a los servidores web de dicho sistema. Para esta variable se toman los siguientes aspectos:

a) Páginas web visitadas

La navegación por las páginas web públicas de un servidor web es probablemente la acción más común y consecuentemente inofensiva que se realiza a través de Internet. Sin embargo, en una fase inicial, cuando un intruso decide averiguar información sobre un sistema, uno de los sitios donde primero puede conseguir información es la propia web pública, explorando todas sus páginas hasta el mínimo detalle. No es muy común, pero

en ocasiones podría obtener detalles sobre la red de la que dispone la empresa o los nombres de los usuarios que pueda tener. Lo que es más probable es que pueda averiguar multitud de detalles sobre la empresa que con un poco de suerte y un administrador descuidado pueden llevarle a la obtención de contraseñas que tengan que ver con nombres mencionados en las páginas.

b) Pedido robots.txt

El fichero robots.txt es un fichero estándar que se encuentra en el directorio raíz de la práctica totalidad de los servidores web y que da información sobre la estructura de directorios para acceso público de la que dispone un servidor web. Esta información normalmente es pedida de forma automática por los robots buscadores cada vez que visitan la página web vigilada para saber que páginas deben de indexar y cuáles no, pero un intruso también la podría utilizar para averiguar más información sobre el sistema custodiado. Para diferenciar en este factor lo que pueden ser visitadas de auténticos buscadores, sólo se tiene en cuenta cuando ha habido otros factores anteriores que hacen sospechar que quien accede es realmente un usuario malintencionado.

La variable *consultas Web* indica el número de páginas web distintas por las que se ha navegado y también si se ha hecho una petición del fichero robots.txt al servidor web al que se presta atención desde la dirección IP vigilada. En este sentido, cabe considerar tres posibles estados de esta variable: Normales, Altas y Excesivas. Los números difusos asociados a estas tres etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

CNW	Normales	Altos	Excesivos
No menor que	0	35	100
Igual que	0	60	150
Igual que	40	100	∞
No mayor que	60	130	∞

Cuadro 3.2 *Variable de Entrada CNW*

Fuente: *Elaboración Propia*

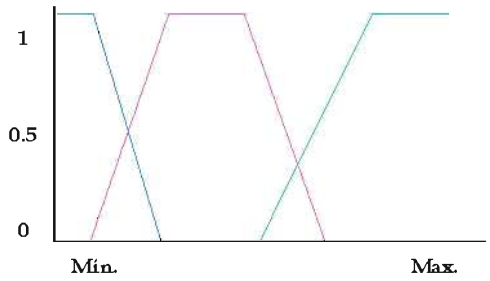


Figura 3.4 *Función de pertenencia variable CNW*

Fuente: *Elaboración Propia*

- **Consultas FTP**

Esta variable indica la posibilidad de que alguien esté intentando recopilar información sobre el sistema a través de consultas aparentemente inofensivas a los servidores FTP del mismo. Para esta variable se toman los siguientes aspectos:

a) Pedidos ficheros FTP sensibles

En los sistemas basados en Unix hay una serie de ficheros de los cuales se puede intentar obtener información altamente valiosa para alguien que quiera penetrar en dicho sistema. Los más sensibles son los que incluyen nombres de usuarios y contraseñas encriptadas, pero también hay otra información sensible aunque lo sea menos como son los servicios que se encuentran abiertos, o su configuración, donde se puede detectar si se está utilizando alguna configuración o versión vulnerable. En ocasiones, un usuario que entre al servidor vigilado a través del servicio FTP, puede intentar conseguir alguno de estos ficheros por si se hubieran dejado disponibles. Por ello, hay una lista de ficheros, entre los cuales se incluyen los antes mencionados (como son el /etc/passwd, el /etc/shadow, /etc/groups...) que se encuentran en una "lista negra" y que harían levantar serias sospechas sobre los intereses de un usuario que los pide.

b) Ficheros de FTP listados

El servicio de FTP pone a disposición de los usuarios, muchas veces de forma anónima, un gran número de ficheros para que puedan ser transferidos desde el servidor remoto a los ordenadores locales de cada interesado. El listado y descarga de ficheros mediante

FTP es una tarea en principio inofensiva, pero al igual que un usuario malintencionado puede buscar información acerca de una compañía mediante su página web, también puede hacerlo indagando en los ficheros que ofrece al público para su descarga. Es una amenaza no muy grande, pero si se une el hecho de que un usuario este haciendo uso excesivo de este tipo de ficheros a otros factores, se pueden aumentar las sospechas sobre sus malas intenciones.

La variable *consultas FTP* indica la suma del número de peticiones de listados de directorios más el número de peticiones de descarga de ficheros que se han hecho al servidor FTP, además si se han pedido mediante FTP alguno de los ficheros que podrían contener información sensible para el acceso al sistema desde la dirección IP vigilada. En este sentido, cabe considerar tres posibles estados de esta variable: Normales, Altos y Excesivos. Los números difusos asociados a estas tres etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

CNF	Normales	Altos	Excesivos
No menor que	0	70	150
Igual que	0	80	250
Igual que	20	100	∞
No mayor que	90	180	∞

Cuadro 3.3 *Variable de Entrada CNF*

Fuente: *Elaboración Propia*

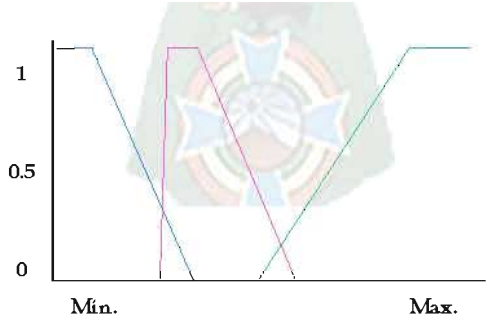


Figura 3.5 *Función de pertenencia variable CNF*

Fuente: *Elaboración Propia*

Consultas URLs peligrosas

Es bastante común en los últimos tiempos que se descubran agujeros de seguridad en servidores web o aplicaciones web comunes a través de los cuales se puede conseguir acceso a importante información de servidores. Estos agujeros de seguridad se basan habitualmente en invocar a determinados ejecutables (CGIs u otro tipo de aplicativos) con determinados parámetros.

Uno de los problemas de este tipo más comunes es el que ocurría en versiones antiguas del servidor web IIS a través del que se podía conseguir acceso a la línea de comando pidiendo URL similares `http://destino/scripts/..%../winnt/system32/cmd.exe` Como ejemplo otras URLs que pueden denotar que alguien está intentando aprovecharse de un error de un software para penetrar en el sistema, se dispone la siguiente línea `http://destino:8080/%2e%2e/%2e%2e%5cfichero%00`.
Para servidores de aplicaciones Tomcat o la `http://destino/cgi-bin/webdist.cgi?distloc=;cat%20/etc/passwd` para el script webdist que se distribuye por defecto con algunos servidores web. Para que el sistema funcionase correctamente se debería de mantener una "lista negra" con estas URLs a través de de algún servicio como el CERT.

La variable *consultas URL peligrosas* indica el número de URLs de las incluidas en la lista de las que tienen notificados problemas de seguridad han sido consultadas (o intentadas consultar) desde la dirección IP vigilado. En este sentido, cabe considerar tres posibles estados de esta variable: Normales, Altas y Excesivas. Los números borrosos asociados a estas tres etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

CUP	Normales	Altos	Excesivos
No menor que	0	1	12
Igual que	0	12	20
Igual que	0	10	∞
No mayor que	2	15	∞

Cuadro 3.4 *Variable de Entrada CUP*

Fuente: *Elaboración Propia*

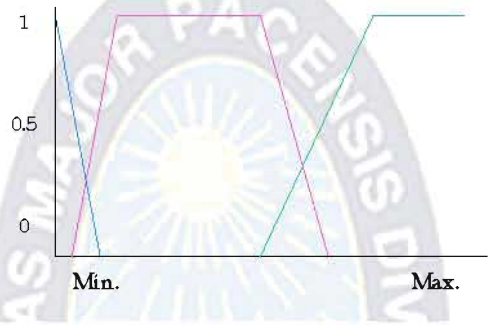


Figura 3.6 *Función de pertenencia variable CUP*

Fuente: *Elaboración Propia*

- **Paquetes ICMP**

El ICMP es uno de los protocolos incluidos en la pila IP que se utiliza principalmente para comprobar conectividades entre diferentes equipos de una red. La gran mayoría (o totalidad) de las aplicaciones utilizadas por un usuario final, no necesitan de este protocolo para nada. Sin embargo, cuando se "cruza" con algún usuario que esté intentando averiguar información sobre sistema custodiado, sí que es bastante habitual que una de las primeras cosas que haga para comprobar si una máquina realmente existe es enviarle un paquete ICMP mediante ping, sabiendo que si obtiene respuesta, es más que posible que detrás de esa dirección haya una máquina a la que poder atacar.

Esta variable indica el número de paquetes ICMP enviados al servidor protegido desde la dirección IP vigilada. En este sentido, cabe considerar tres posibles estados de esta variable: Normales, Altos y Excesivos. Los números borrosos asociados a estas tres

etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

PAI	Normales	Altos	Excesivos
No menor que	0	12	60
Igual que	0	40	100
Igual que	0	80	∞
No mayor que	0	120	∞

Cuadro 3.5 *Variable de Entrada PAI*

Fuente: *Elaboración Propia*

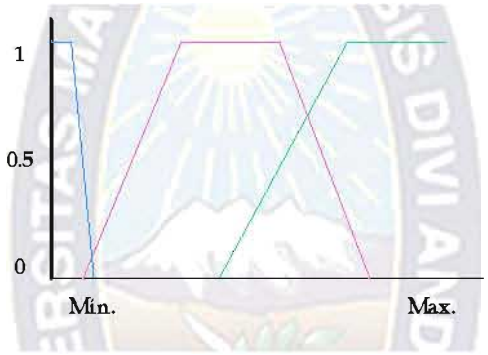


Figura 3.7 *Función de pertenencia variable PAI*

Fuente: *Elaboración Propia*

- **Binds a Puertos Diferentes**

La gran mayoría de las comunicaciones en Internet se realizan a través del protocolo TCP/IP, que es un protocolo orientado a conexión, que establece para cada máquina un total de 65536 puertos diferentes en los cuales pueden albergarse servicios. Este protocolo está orientado a conexión por lo cual, antes de establecer cualquier conexión, lo primero que hace es un intercambio de paquetes sobre el puerto que desee para confirmar que ésta se puede producir. El primero de esos paquetes se llama "BIND" y si obtiene una respuesta a él, aunque no obtenga más información posterior, ya sabe que hay algún servicio funcionando a través de ese puerto. Sabiendo esto, muchos de los usuarios que desean atacar un sistema, antes de probar nada, realizan lo que se denomina un "escaneo de puertos", que consiste en el envío de estos paquetes de "BIND" a

diferentes puertos del servidor para comprobar si hay respuesta y por lo tanto pueden tener algún servicio al que atacar en ese sitio.

Los programas más automatizados hacen un auténtico bombardeo de peticiones a todos los puertos de un servidor que son fácilmente detectables, pero otros usuarios más precavidos sólo lo realizan sobre determinados puertos que saben que pueden contener puntos débiles. Esta variable indica el número puertos diferentes del servidor protegido a los que se han hecho bind TCP o UDP desde la dirección IP vigilada. En este sentido, cabe considerar tres posibles estados de esta variable: Normales, Altas y Excesivas. Los números borrosos asociados a estas tres etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

BPD	Normales	Altos	Excesivos
No menor que	0	7	50
Igual que	0	12	300
Igual que	5	60	∞
No mayor que	8	100	∞

Cuadro 3.6 *Variable de Entrada BPD*

Fuente: *Elaboración Propia*

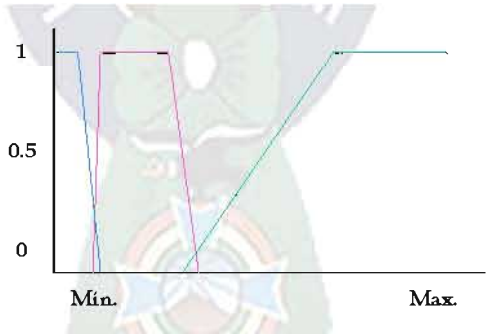


Figura 3.8 *Función de pertenencia variable BPD*

Fuente: *Elaboración Propia*

- **Logins Fallidos**

Cuando se hace referencia al login, se cuestiona el intento de un usuario de entrar en el sistema vigilado a través de alguno de los métodos habituales para el sistema operativos,

tales como pueden ser un telnet o ssh para unix, o pc-anywhere, VNC o terminal services para un servidor windows. También se puede incluir dentro de esta clasificación los intentos de acceso por FTP con login/password, ya que puede entrañar riesgos comparables a los de los otros métodos de acceso.

La variable *logins fallidos* indica la posibilidad de que alguien esté intentando emplear el método prueba - error para entrar en el sistema vigilado con algún par usuario/password que confía en que puedan ser los adecuados. En este sentido, cabe considerar tres posibles estados de esta variable: Normales, Altos y Excesivos. Los números difusos asociados a estas tres etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

LOF	Normales	Altos	Excesivos
No menor que	0	2	10
Igual que	0	3	20
Igual que	1	12	∞
No mayor que	4	18	∞

Cuadro 3.7 *Variable de Entrada LOF*

Fuente: *Elaboración Propia*

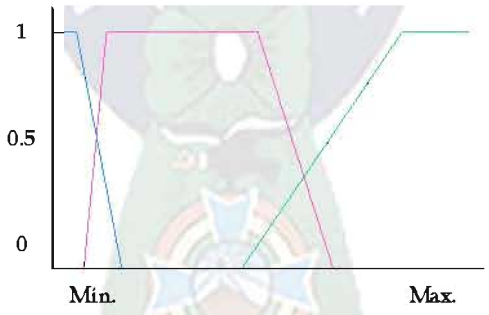


Figura 3.9 *Función de pertenencia variable LOF*

Fuente: *Elaboración Propia*

↳ Comandos enviados en URL

Según se puede observar estando al tanto de las listas relativas a aspectos de seguridad, buena parte de los agujeros de seguridad que pueden tener aplicaciones web, se refieren

a la posibilidad de que, por errores en la programación, se pueda acceder a ficheros o ejecutables localizados fuera del directorio de la aplicación pasando valores inadecuados a algunas de las variables utilizadas. Por ello, un buen método preventivo para detectar ataques de este tipo que se puedan estar dando, es el detectar ciertos comandos o ficheros del sistema especialmente sensibles a los que se puede intentar acceder aprovechándose de errores. Ejemplos de estos ficheros pueden ser "/etc/passwd", "/bin/bash", "system32\cmd.exe" o "root.exe".

La variable *comandos enviados en URL* indica el número de comandos de los considerados como malintencionados enviados al sistema, que se envía en peticiones de URL al sistema protegido desde la dirección IP vigilada. En este sentido, cabe considerar tres posibles estados de esta variable: Normales, Altos y Excesivos. Los números difusos asociados a estas tres etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

CEU	Normales	Altos	Excesivos
No menor que	0	2	8
Igual que	0	5	20
Igual que	2	10	∞
No mayor que	3	15	∞

Cuadro 3.8 *Variable de Entrada CEU*

Fuente: *Elaboración Propia.*

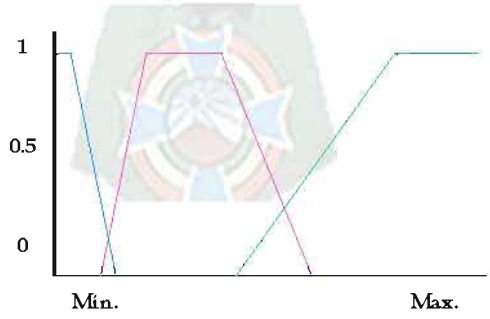


Figura 3.10 *Función de pertenencia variable CEU*

Fuente: *Elaboración Propia*

Cabeceras de protocolo inconsistentes

Cada protocolo tiene sus propias reglas para construir paquetes e in- dicar por medio de diferentes campos, el tipo de transmisión, longitud, número de orden, así como gran multitud de datos dependientes de ellos. Una fuente de multitud de ataques (principalmente de denegación de servicio) es precisamente el crear paquetes con cabeceras que se salgan de los estándares, y que provoquen el funcionamiento del sistema de una forma inesperada al no saber cómo tratarlos. El sensor encargado de controlar las cabeceras de protocolo, debería de tener información sobre todos los protocolos IP, así como los valores válidos para sus cabeceras, y debería de chequear todos los paquetes que pasan a su través para detectar aquellos que no cumplan los estándares. Un número bajo de éstos puede ser debido a problemas en la comunicación a través de la red, pero si se incrementa puede suscitar la sospecha de un ataque.

La variable *cabeceras de protocolo inconsistentes* indica el número de cabeceras de protocolo no estándar que se detectan en la red protegida con origen en la dirección IP vigilada. En este sentido, cabe considerar tres posibles estados de esta variable: Normales, Altas y Excesivas. Los números difusos asociados a estas tres etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

CPI	Normales	Altos	Excesivos
No menor que	0	2	3
Igual que	0	4	6
Igual que	1	5	∞
No mayor que	3	8	∞

Cuadro 3.9 *Variable de Entrada CPI*

Fuente: *Elaboración Propia*

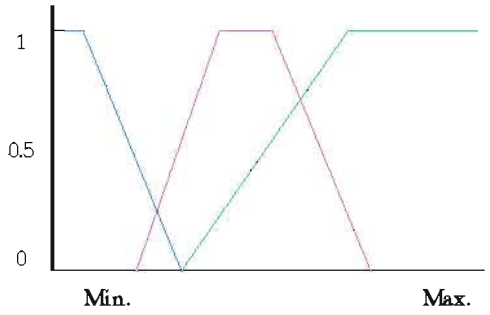


Figura 3.11 *Función de pertenencia variable CPI*

Fuente: *Elaboración Propia*

Logs Modificados

Todos los sistemas servidores disponen de ficheros de trazas en los que se almacenan multitud de datos de uso y acceso al sistema. Estos son los casos de ficheros como el `access_log` de Apache, los `/var/adm/messages`, `/var/log/syslog`, `/var/adm/lastlog` o `xferlog` de Unix o los del visor de eventos de Windows. En caso de que un usuario no autorizado esté entrando en un sistema es posible que sea posible su detección a través de un examen de éstos. Para prevenir esta detección, los intrusos pueden intentar en ocasiones borrar sus huellas modificando los ficheros, cosa que no es en absoluto normal, ya que estos ficheros únicamente varían para añadir nueva información y no para modificar ninguna antigua. En caso de que se detecte algún cambio en estos ficheros hay serias posibilidades de que se esté enfrentando a un intruso.

La variable *logs modificados* indica el número de ficheros de trazas (logs) que han sufrido modificaciones en su contenido por parte de un usuario que haya entrado desde la dirección IP vigilada. En este sentido, cabe considerar tres posibles estados de esta variable: Normales, Altos y Excesivos. Los números borrosos asociados a estas tres etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

LOM	Normales	Altos	Excesivos
No menor que	0	1	2
Igual que	0	1	4
Igual que	1	3	∞
No mayor que	2	4	∞

Cuadro 3.10 *Variable de Entrada LOM*

Fuente: *Elaboración Propia*

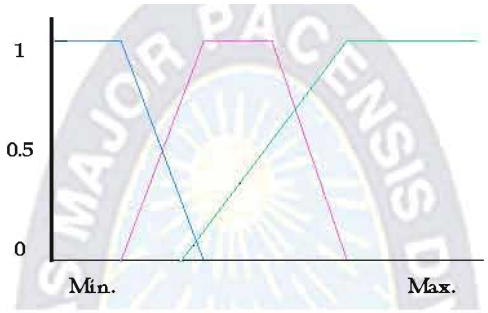


Figura 3.12 *Función de pertenencia variable LOM*

Fuente: *Elaboración Propia*

Cuentas creadas

Cuando un usuario consigue entrar en un sistema a través de un error o los datos de otro usuario, suele intentar hacerse con otro medio de acceso al sistema para poder seguir entrando en él en el caso de que se corrigiese el error o bien el usuario cambiarse de contraseña. El medio más sencillo para esto es la creación de una nueva cuenta de usuario. Normalmente el intruso suele hacerlo de forma que el usuario que se cree pueda parecerse a algún usuario anterior o parezca un administrador o usuario importante del sistema. En cualquiera de los casos, la creación de esta cuenta es fácilmente detectable por un sistema que lleve el control de usuarios y puede dar una alerta a tener en cuenta.

La variable *cuentas creadas* indica el número de cuentas que han sido creadas por un usuario que haya entrado desde la dirección IP vigilada. En este sentido, cabe considerar tres posibles estados de esta variable: Normales, Alguna y Excesivas. Los números difusos asociados a estas tres etiquetas lingüísticas y la representación grafica de los mismos se encuentran recogidos en el siguiente cuadro.

CUC	Normales	Altos	Excesivos
No menor que	0	1	2
Igual que	0	2	3
Igual que	1	3	∞
No mayor que	2	4	∞

Cuadro 3.11 *Variable de Entrada CUC*

Fuente: *Elaboración Propia*

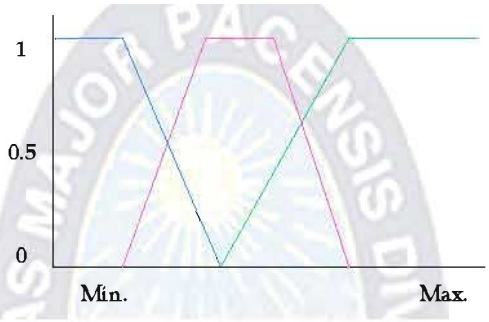


Figura 3.13 *Función de pertenencia variable CUC*

Fuente: *Elaboración Propia*

- **Servicios de Red Ejecutados**

Cuando un usuario consigue entrar en un sistema a través de un error o los datos de otro usuario, suele intentar hacerse con otro medio de acceso al sistema para poder seguir entrando en él en el caso de que se corrigiese el error o bien el usuario cambiase de contraseña. Aparte de la creación de un usuario otra forma de hacerlo es instalando un programa que sirva de "puerta trasera" para entrar al sistema de forma discreta sin pasar por los medios habituales de acceso que pueden estar más controla- dos. Estos programas reciben el nombre de troyanos y se caracterizan por quedarse residentes en el sistema como un servicio de red escuchando en alguno de los puertos TCP, esperando una conexión. En este apartado se tratará de vigilar los procesos no estándar (excluyendo servicios como la web, ftp, telnet, etc.) que están corriendo en el sistema ligados a alguno de los puertos de red y se controlará que el usuario que los ha ejecutado para detectar la posibilidad de que el servicio de red ejecutado y por el que se transmite información pueda ser un troyano.

La variable *servicios de red ejecutados* indica el número de programas asociados a un puerto de la red que se han ejecutado por el usuario que haya entrado desde la dirección IP vigilada y se encuentran activos. En este sentido, cabe considerar tres posibles estados de esta variable: Normales, Altos y Excesivos. Los números borrosos asociados a estas tres etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

SRE	Normales	Altos	Excesivos
No menor que	0	1	1
Igual que	0	1	5
Igual que	0	2	∞
No mayor que	2	3	∞

Cuadro 3.12 *Variable de Entrada SRE*

Fuente: *Elaboración Propia*

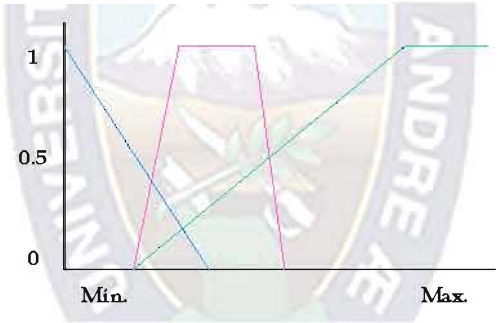


Figura 3.14 *Función de pertenencia variable SRE*

Fuente: *Elaboración Propia*

Operaciones con Ficheros

Una vez que un usuario malintencionado ha accedido al sistema objeto de atención, una de las cosas que puede intentar hacer es extraer de él información que puede ser valiosa. Para detectar estas acciones, este apartado trata del control del número de ficheros que se copian desde el servidor al exterior por alguno de los protocolos disponibles. Esta información se ha de tomar de forma cautelosa si no está acompañada de otros síntomas de ataque, ya que la mera copia de un número alto de ficheros puede deberse a razones ajenas que no tengan que ver con ningún intruso.

Otra de las opciones que puede tomar un usuario malintencionado que entra en sistema custodiado es el sabotear o destruir la información que esté contenida en los sistemas de información allí activos. Para detectar estas acciones, se puede controlar el número de ficheros que el usuario del que se sospecha borra. Dado que aunque también pueda ser realizada por usuarios normales, el borrado es una acción más peligrosa, por lo que será preciso tenerla más en cuenta que la copia de ficheros.

Esta variable indica el número de ficheros copiados del servidor a la dirección IP vigilada a través de alguno de los protocolos previstos, además del número de ficheros borrados del servidor por el usuario que entra desde la dirección IP vigilada. En este sentido, cabe considerar tres posibles estados de esta variable: Normales, Altos y Excesivos. Los números borrosos asociados a estas tres etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

OPF	Normales	Altos	Excesivos
No menor que	0	20	40
Igual que	5	30	80
Igual que	15	60	∞
No mayor que	50	70	∞

Cuadro 3.13 *Variable de Entrada OPF*

Fuente: *Elaboración Propia*

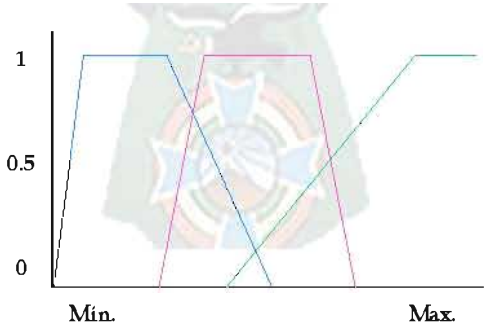


Figura 3.15 *Función de pertenencia variable OPF*

Fuente: *Elaboración Propia*

- **Utilización de recursos**

Esta variable indica el consumo de los recursos del sistema que está haciendo el usuario al que estamos vigilando. Para la variable se consideran los siguientes aspectos:

a) Utilización de CPU

Una de las formas en las que un intruso puede aprovecharse del sistema es utilizando sus recursos para sus propios propósitos: ya sea para procesar información o para saltar a otros sistemas desde él. Una manera sencilla de detectar que alguien está haciendo un abuso de los recursos del sistema protegido es ver el consumo de CPU que están haciendo los procesos que este ha ejecutado. Esto es lo que se encarga de medir esta variable, que de paso sirve para detectar el empleo excesivo de CPU como situación anómala.

b) Utilización de Memoria

El otro recurso que suele ser más consumido en un computador, es la memoria. Contabilizando el espacio ocupado en memoria por los procesos de un usuario, se puede detectar si está haciendo algún abuso del sistema y detectar además situaciones anómalas.

La variable *utilización de recursos* indica el porcentaje de memoria consumida por los procesos ejecutados, también indica el porcentaje de utilización de CPU consumido por los procesos ejecutados por el usuario que entra desde la dirección IP vigilada. En este sentido, cabe considerar tres posibles estados de esta variable: Normal, Mucho y Demasiado. Los números borrosos asociados a estas tres etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

UTR	Normales	Altos	Excesivos
No menor que	0	10	40
Igual que	5	25	60
Igual que	10	30	100
No mayor que	15	50	100

Cuadro 3.14 *Variable de Entrada UTR*

Fuente: *Elaboración Propia*

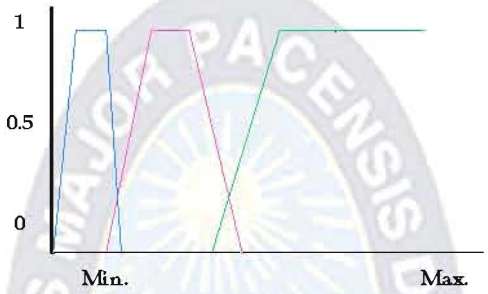


Figura 3.16 *Función de pertenencia variable UTR*

Fuente: *Elaboración Propia*

Estado del Servidor

Algunos de los ataques sobre servidores, lo único que persiguen es provocar que el rendimiento del sistema baje o simplemente deje de funcionar adecuadamente. Estos son los ataques denominados DoS (Denial of Servide o de denegación de servicio). Para detectar si el estado del servidor es el correcto o no se encuentra prestando servicio de red adecuadamente, se pretende medir este factor a través de un "sensor" que haga un ping al sistema vigilado a través de la intranet o red interna y mida el retraso con el que llega, para de esta forma poder conocer su estado de funcionamiento. El ping tiene un timeout de 4000 milisegundos, por lo cual, todo ping que tarde en volver más de este tiempo será descartado y se dará al sistema como totalmente caído.

La variable *estado del servidor* indica el tiempo de respuesta del servidor protegido a un ping desde uno de los sensores situados en su propia red. En este sentido, cabe considerar tres posibles estados de esta variable: Bueno, Malo y Caído. Los números borrosos asociados a estas tres etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

ESE	Bueno	Malo	Caído
No menor que	0	30	3000
Igual que	1	50	4000
Igual que	15	500	5000
No mayor que	60	4000	5000

Cuadro 3.15 *Variable de Entrada ESE*

Fuente: *Elaboración Propia*

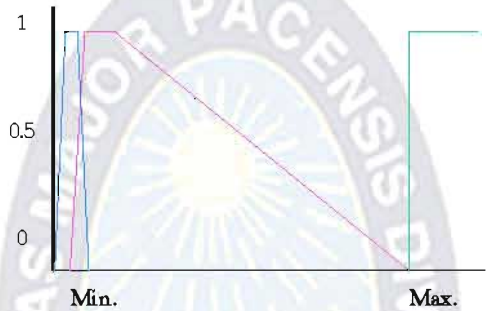


Figura 3.17 *Función de pertenencia variable ESE*

Fuente: *Elaboración Propia*

3.5 Establecimiento del sistema de inferencia

Una vez especificadas las variables de entrada, esto es, las definiciones de cada una de las etiquetas lingüísticas en las que se dividen tales variables y dando por conocidas las relaciones entre las mismas, el siguiente paso en la construcción del sistema de detección de intrusos consiste en el establecimiento de las bases de reglas difusas del tipo:

SI…ENTONCES que denotarán el grado de influencia de cada variable y permitirán alcanzar la solución final del problema.

En primer término, estas reglas permitirán definir "las variables intermedias" a partir de la combinación de variables de entrada, y a su vez, definir la variable de salida del modelo, a partir de la combinación de las intermedias.

A este respecto, las variables intermedias que serán utilizadas en el modelo difuso para la detección de intrusos son las siguientes:

- Posibilidad investigación externa
- Posibilidad investigación interna
- Posibilidad de ataque
- Posibilidad borrado huellas
- Posibilidad aprovechamiento

3.5.1 Variables Intermedias

Posibilidad de Investigación Externa

Esta variable indica la posibilidad de que alguien esté intentando conseguir alguna información sobre el sistema de información de la organización objeto de estudio desde fuera a través de métodos aparentemente inocuos, aprovechando la información que se dispone pública y abiertamente a comunidad de Internet. En este sentido, cabe considerar cinco posibles estados de esta variable: Muy Baja(MB), Baja (B), Media (M), Alta (A) y Excesiva (E). Los números borrosos asociados a estas cinco etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

Investigación Externa	Muy Baja	Baja	Media	Alta	Excesiva
No menor que	0	10	30	60	85
Igual que	0	20	40	70	90
Igual que	10	25	60	85	100
No mayor que	15	35	65	90	100

Cuadro 3.16 *Variable Intermedia Investigación Externa*

Fuente: *Elaboración Propia*

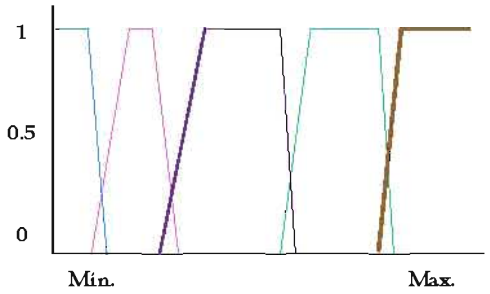


Figura 3.18 *Función de pertenencia variable intermedia*

Fuente: *Elaboración Propia*

Base de Reglas: El valor de "Posibilidad investigación externa" se obtiene relacionando las variables que indican las posibilidad de que alguien esté intentando obtener información del sistema protegido a través de "consultas DNS", "consultas web" y "consultas FTP" tal como se muestra en el siguiente cuadro.

ENTRADAS			SALIDA
Consultas DNS	**Consultas web**	**Consultas FTP**	***Posibilidad investigación externa***
Baja	Baja	Baja	Muy baja
Baja	Baja	Media	Muy baja
Baja	Baja	Alta	Media
…….	…….	…….	…….
Baja	Alta	Media	Media
Baja	Alta	Alta	Alta
Media	Baja	Baja	Muy baja
Media	Baja	Media	Baja
…….	…….	…….	…….
Media	Alta	Media	Alta
Media	Alta	Alta	Alta
Alta	Baja	Baja	Baja
Alta	Baja	Media	Media
…….	…….	…….	…….
Excesiva	Alta	Baja	Alta
Excesiva	Alta	Media	Excesiva
Excesiva	Alta	Alta	Excesiva

Cuadro 3.17 *Base de Reglas para la variable Investigación Externa*

Fuente: *Elaboración Propia*

Posibilidad de Investigación Interna

Esta variable indica la posibilidad de que alguien esté intentando conseguir alguna información sobre el sistema de información haciendo intentos de conexión más intrusivos, que no son propios de usuarios en condiciones normales de uso. En este sentido, cabe considerar cinco posibles estados de esta variable: Muy Baja, Baja, Media, Alta y Excesiva. Los números borrosos asociados a estas cinco etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

Investigación Interna	Muy Baja	Baja	Media	Alta	Excesiva
No menor que	0	10	30	60	85
Igual que	0	20	40	70	90
Igual que	10	25	60	85	100
No mayor que	15	35	65	90	100

Cuadro 3.18 *Variable Intermedia Investigación Interna*

Fuente: *Elaboración Propia*

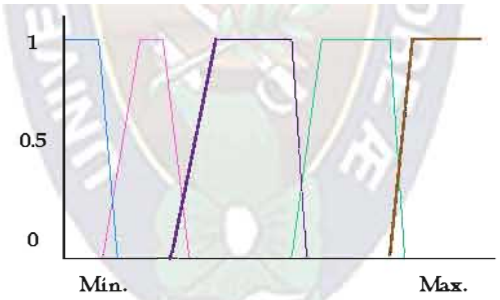


Figura 3.19 *Función de pertenencia variable intermedia*

Fuente: *Elaboración Propia*

Base de Reglas: El valor de "Posibilidad investigación interna" se obtiene relacionando las variables que indican las posibilidades de que alguien esté intentando obtener información de dicho sistema a través de "consultas a URLs peligrosas", "paquetes ICMP", y "binds a puertos distintos" tal como se muestra en el siguiente cuadro.

ENTRADAS			SALIDA
Consultas a URLs peligrosas	**Paquetes ICMP**	**Binds a puertos distintos**	***Posibilidad investigación interna***
Normales	Normales	Normales	Baja
Normales	Normales	Normales	Muy baja
Normales	Normales	Normales	Media
…….	…….	…….	…….
Normales	Normales	Excesivas	Excesiva
Normales	Normales	Excesivas	Alta
Normales	Normales	Excesivas	Excesiva
Normales	Normales	Excesivas	Alta
…….	…….	…….	…….
Normales	Altos	Excesivas	Excesiva
Normales	Altos	Excesivas	Alta
Normales	Altos	Excesivas	Excesiva
Normales	Altos	Excesivas	Alta
…….	…….	…….	…….
Normales	Excesivos	Excesivas	Excesiva
Normales	Excesivos	Excesivas	Excesiva
Altas	Normales	Normales	Media
Altas	Normales	Normales	Baja
…….	…….	…….	…….
Excesivas	Normales	Normales	Media
Excesivas	Normales	Normales	Media
Excesivas	Normales	Normales	Alta
Excesivas	Normales	Normales	Media
…….	…….	…….	…….
Excesivas	Altos	Excesivas	Excesiva
Excesivas	Altos	Excesivas	Alta
Excesivas	Altos	Excesivas	Excesiva
Excesivas	Altos	Excesivas	Excesiva

Cuadro 3.19 *Base de Reglas para la variable Investigación Interna*

Fuente: *Elaboración Propia*

↳ Posibilidad de Ataque

Esta variable indica la posibilidad de que alguien esté haciendo intentos (sean o no acertados) para entrar en el sistema vigilado de una forma no lícita. En este sentido, cabe considerar cuatro posibles estados de esta variable: Baja, Media, Alta y Excesiva. Los

números borrosos asociados a estas cuatro etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

Posibilidad de Ataque	Baja	Media	Alta	Excesiva
No menor que	0	20	60	80
Igual que	0	30	70	90
Igual que	15	50	85	100
No mayor que	35	65	90	100

Cuadro 3.20 *Variable Intermedia Posibilidad de Ataque*

Fuente: *Elaboración Propia*

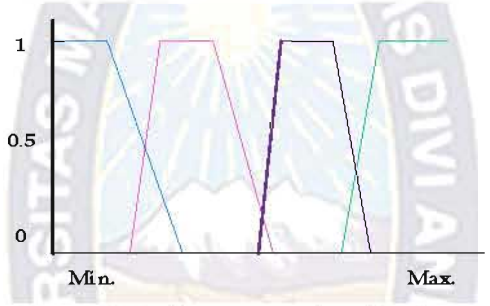


Figura 3.20 *Función de pertenencia variable intermedia*

Fuente: *Elaboración Propia*

Base de Reglas: El valor de "Posibilidad de ataque" se obtiene relacionando las variables que indican la posibilidad de que un usuario esté intentando entrar en el sistema por el método prueba - error ("logins fallidos"), el número de "comandos enviados en URLs" detectados y el número de "paquetes con cabeceras de protocolo inconsistentes" detectados tal como se muestra en el siguiente cuadro.

ENTRADAS			SALIDA
Logins Fallidos	Comandos enviados en URLs	Paquetes con cabeceras inconsistentes	*Posibilidad de Ataque*
Baja	Normales	Normales	Baja
Baja	Normales	Altas	Baja
Baja	Normales	Excesivas	Baja
Baja	Altos	Normales	Media
…….	…….	…….	…….
Media	Excesivos	Altas	Alta
Media	Excesivos	Excesivas	Excesiva

Alta	Normales	Normales	Media
Alta	Normales	Altas	Media
…….	…….	…….	…….
Excesiva	Altos	Altas	Excesiva
Excesiva	Altos	Excesivas	Excesiva
Excesiva	Excesivos	Normales	Excesiva
Excesiva	Excesivos	Altas	Excesiva
Excesiva	Excesivos	Excesivas	Excesiva

Cuadro 3.21 *Base de Reglas para la variable Posibilidad de Intrusión*

Fuente: *Elaboración Propia*

↳ Posibilidad de Borrado de Huellas

Esta variable indica la posibilidad de que alguien esté intentando borrar del sistema que estamos vigilando, huellas que puedan delatar su paso o intrusión o bien esté preparando el sistema para no dejar huellas en sus futuras incursiones. En este sentido, cabe considerar cuatro posibles estados de esta variable: Baja, Media, Alta y Excesiva. Los números borrosos asociados a estas cuatro etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

Borrado de Huellas	Baja	Media	Alta	Excesiva
No menor que	0	20	60	80
Igual que	0	30	70	90
Igual que	15	50	85	100
No mayor que	35	65	90	100

Cuadro 3.22 *Variable Intermedia Borrado de Huellas*

Fuente: *Elaboración Propia*

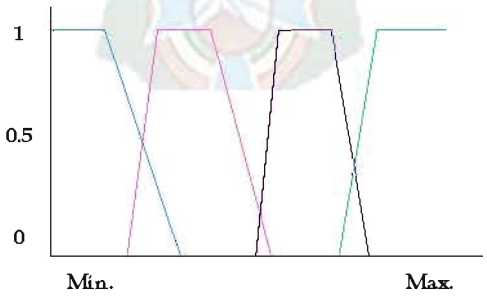


Figura 3.21 *Función de pertenencia variable intermedia*

Fuente: *Elaboración Propia*

Base de Reglas: El valor de "Posibilidad de borrado de huellas" se obtiene relacionando las variables que indican, para el usuario vigilado, el número de ficheros de "logs modificados" por él, las "cuentas creadas", así como los "servicios de red ejecutados" por posible intruso, tal como se muestra en el siguiente cuadro:

ENTRADAS			SALIDA
Logs modificados	Cuentas creadas	Servicios de red ejecutados	*Posibilidad Borrado de huellas*
Normales	Normales	Normales	Baja
Normales	Normales	Altos	Baja
Normales	Normales	Excesivos	Media
…….	…….	…….	…….
Normales	Excesivas	Altos	Alta
Normales	Excesivas	Excesivos	Alta
Altos	Normales	Normales	Baja
Altos	Normales	Altos	Media
…….	…….	…….	…….
Altos	Excesivas	Altos	Alta
Altos	Excesivas	Excesivos	Excesiva
Excesivos	Normales	Normales	Media
Excesivos	Normales	Altos	Alta
…….	…….	…….	…….
Excesivos	Excesivas	Normales	Excesiva
Excesivos	Excesivas	Altos	Excesiva
Excesivos	Excesivas	Excesivos	Excesiva

Cuadro 3.23 *Base de Reglas para la variable Posibilidad de Borrado de Huellas*

Fuente: *Elaboración Propia*

Posibilidad de Aprovechamiento

Esta variable indica la posibilidad de que el usuario al que vigilamos esté haciendo un aprovechamiento del sistema para fines ilícitos o perniciosos. En este sentido, cabe considerar cuatro posibles estados de esta variable: Baja, Media, Alta y Excesiva. Los números borrosos asociados a estas cuatro etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

Posibilidad de Aprovechamiento	*Baja*	*Media*	*Alta*	*Excesiva*
No menor que	0	20	60	80
Igual que	0	30	70	90
Igual que	15	50	85	100
No mayor que	35	65	90	100

Cuadro 3.24 *Variable Intermedia Posibilidad de Aprovechamiento*

Fuente: *Elaboración Propia*

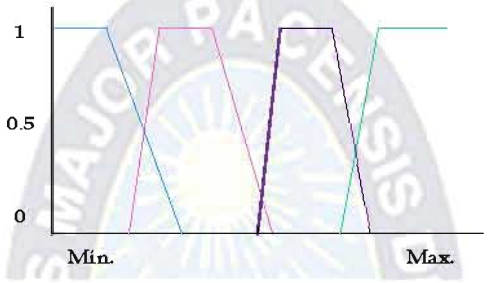


Figura 3.22 *Función de pertenencia variable intermedia*

Fuente: *Elaboración Propia*

Base de Reglas: El valor de "Posibilidad de aprovechamiento" se obtiene relacionando las variables que indican para el usuario vigilado, el número de operaciones con ficheros", el "uso de recursos" del sistema que hace y el "estado del servidor" durante su conexión, de la siguiente forma:

ENTRADAS			SALIDA
Operacion con Ficheros	Estado del Servidor	Utilización de Recursos	*Posibilidad de Aprovechamiento*
Normales	Bueno	Normal	Baja
Normales	Bueno	Alto	Baja
…….	…….	…….	…….
Normales	Malo	Excesivo	Alta
Altos	Bueno	Normal	Baja
…….	…….	…….	…….
Altos	Caido	Excesivo	Excesiva
Excesivos	Bueno	Normal	Media
…….	…….	…….	…….
Excesivos	Malo	Normal	Excesiva
Normales	Bueno	Excesivo	Alta
…….	…….	…….	…….

Normales	Caido	Excesivo	Excesiva
Altos	Bueno	Normal	Baja
…….	…….	…….	…….
Altos	Malo	Excesivo	Excesiva
Excesivos	Bueno	Normal	Media
…….	…….	…….	…….
Excesivos	Malo	Alto	Excesiva
Excesivos	Malo	Excesivo	Excesiva

Cuadro 3.25 *Base de Reglas para la variable posibilidad de Aprovechamento*

Fuente: *Elaboración Propia*

3.6 Desfuzificación

3.6.1 Variable de Salida

Las variables de salida son las que indican la valoración que hace el sistema de una determinada cuestión en función de todos los factores de entrada que conocemos. En el caso de estudio, se dispone de una única variable de salida que indica la posibilidad de intrusión del usuario que está siendo vigilado tenga "malas intenciones" al respecto del sistema o red.

Esta variable indica la posibilidad de que el usuario vigilado esté en el interior del sistema realizando alguna acción ilícita o no deseada. En este sentido, cabe considerar cinco posibles estados de esta variable: Imposible, Improbable, Posible, Probable y Seguro. Los números difusos asociados a estas cinco etiquetas lingüísticas y la representación gráfica de los mismos se encuentran recogidos en el siguiente cuadro.

Pos. de Intrusión	Imposible	Improbable	Posible	Probable	Seguro
No menor que	0	10	30	60	85
Igual que	0	20	40	70	90
Igual que	10	25	60	85	100
No mayor que	16	35	65	90	100

Cuadro 3.26 *Variable de Salida Posibilidad de Intrusión*

Fuente: *Elaboración Propia*

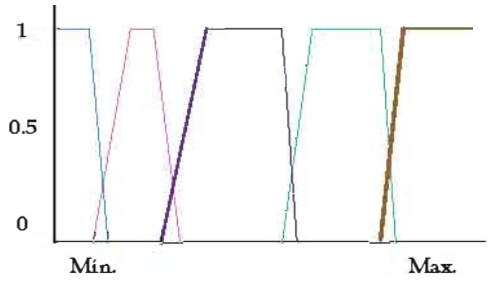


Figura 3.27 *Función de pertenencia variable de salida*

Fuente: *Elaboración Propia*

Base de Reglas: El valor de "Posibilidad de intrusión" se obtiene relacionando las variables intermedias, tal como se muestra en el siguiente cuadro:

ENTRADAS					SALIDA
Inv. Externa	Inv. Interna	P. de Ataque	P. Borrado	P. Aprovechamiento	Posibilidad de Intrusión
Muy Baja	Muy Baja	Baja	Baja	Baja	Imposible
Muy Baja	Media	Baja	Baja	Alta	Posible
Muy Baja	Alta	Baja	Baja	Excesiva	Probable
…….	…….	…….	…….	…….	…….
Baja	Alta	Baja	Excesiva	Alta	Seguro
Baja	Excesiva	Baja	Excesiva	Excesiva	Seguro
Baja	Baja	Media	Baja	Baja	Improbable
…….	…….	…….	…….	…….	…….
Media	Excesiva	Media	Excesiva	Excesiva	Seguro
Media	Baja	Alta	Baja	Baja	Posible
Media	Baja	Alta	Baja	Media	Posible
…….	…….	…….	…….	…….	…….
Alta	Muy Baja	Alta	Excesiva	Alta	Seguro
Alta	Media	Excesiva	Baja	Baja	Probable
Alta	Alta	Excesiva	Baja	Media	Probable
…….	…….	…….	…….	…….	…….
Excesiva	Baja	Alta	Media	Media	Seguro
Excesiva	Baja	Baja	Excesiva	Media	Seguro
Excesiva	Media	Media	Media	Alta	Seguro

Cuadro 3.27 *Variable de Salida Posibilidad de Intrusión*

Fuente: *Elaboración Propia*

Finalmente, y una vez que se ha obtenido el grado de pertenencia difuso de cada ejemplo a cada subconjunto difuso de la variable final, el último paso en la construcción del modelo difuso de detección de intrusos consiste en determinar, a partir de dicha evaluación difusa, el valor o calificación concreta de la variable de salida posibilidad de intrusión.

Se precisa entonces proceder a la fase de clarificación o desfuzificación del resultado obtenido, el método utilizado es el centro de gravedad o centroide, atendiendo al tipo de evaluación de reglas difusas, esto es, se precisa proceder a la concretización o reducción a un valor específico de toda la información contenida en el modelo planteado.

3.7 Evaluación del Modelo

Después del diseño del modelo difuso, el siguiente paso es evaluar el modelo. Para realizar la evaluación del modelo se implementa el mismo en MatLab 7.6 debido a las facilidades que ofrece, ya que cuenta con una caja de herramientas de lógica difusa, lo cual facilita el diseño del modelo debido a su interfaz amigable y flexible.

Para la evaluación del funcionamiento del modelo difuso, se efectuaran un conjunto de pruebas. A través de los resultados obtenidos en las pruebas se evalúa el modelo.

3.8 Implementación del Modelo

3.8.1 Modelado difuso

Los sistemas difusos se utilizan para modelar comportamiento de un sistema real. Para modelar usamos el modelo difuso *(fuzzy modeling)*. El sistema de reglas difusas hace que el incorporar conocimiento experto al sistema sea muy fácil. Por lo tanto, el modelado difuso tiene como ventaja que aprovecha **conocimiento del dominio**. Se pueden usar técnicas convencionales para el aprendizaje estructural y paramétrico de las reglas.

3.8.2 Selección de las Variables de Entrada y Salida

Primeramente se realiza la selección de las características de entrada y salida más relevantes.

- Variables de Entrada:

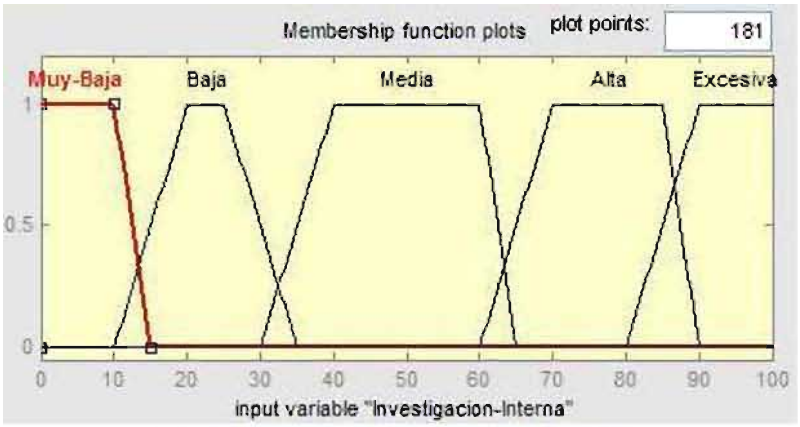


Figura 3.24 *Variable de Entrada Posibilidad de Investigación Interna*

Fuente: *Elaboración Propia*

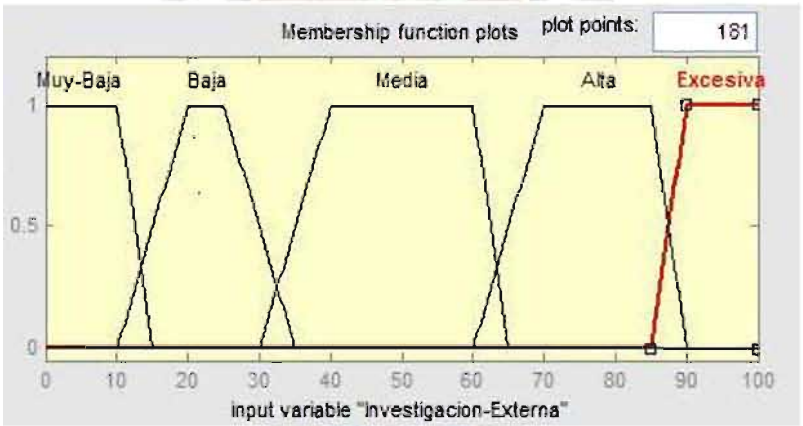


Figura 3.25 *Variable Posibilidad de Investigación Externa*

Fuente: *Elaboración Propia*

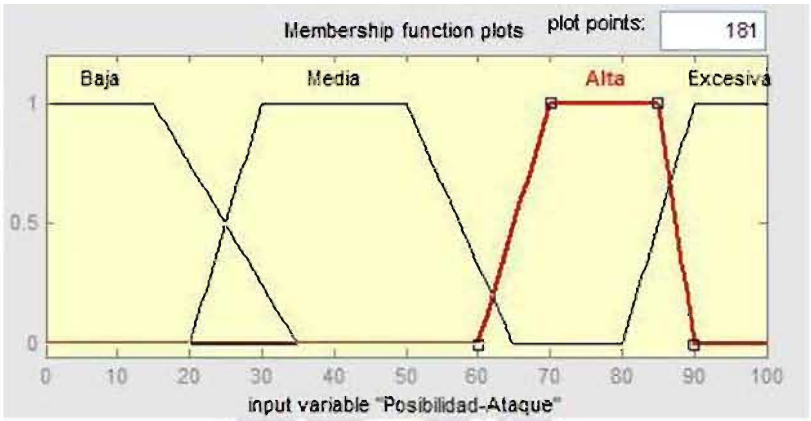


Figura 3.26 *Variable Posibilidad de Ataque*

Fuente: *Elaboración Propia*

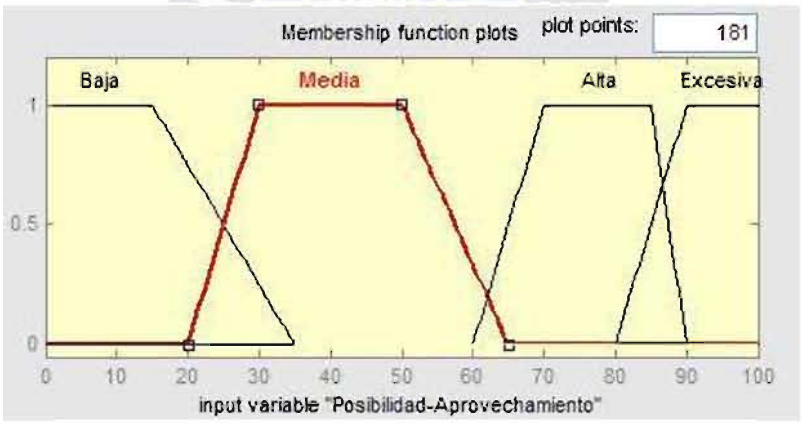


Figura 3.27 *Variable de Entrada Posibilidad de Aprovechamiento*

Fuente: *Elaboración Propia*

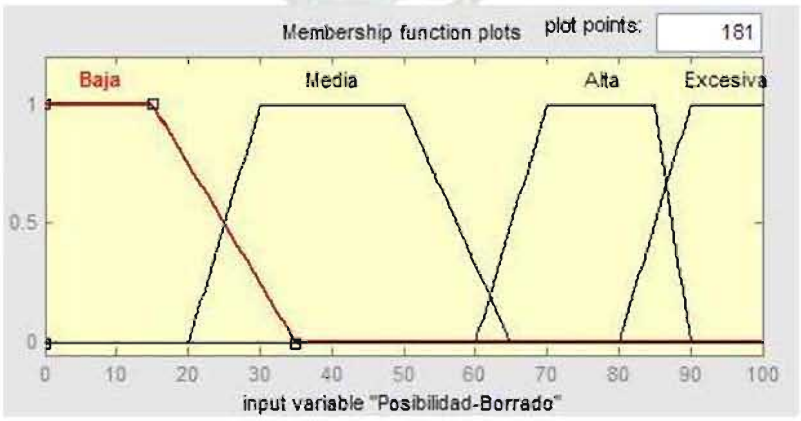


Figura 3.28 *Variable de Entrada Posibilidad de Borrado de Huellas*

Fuente: *Elaboración Propia*

- Variable de Salida

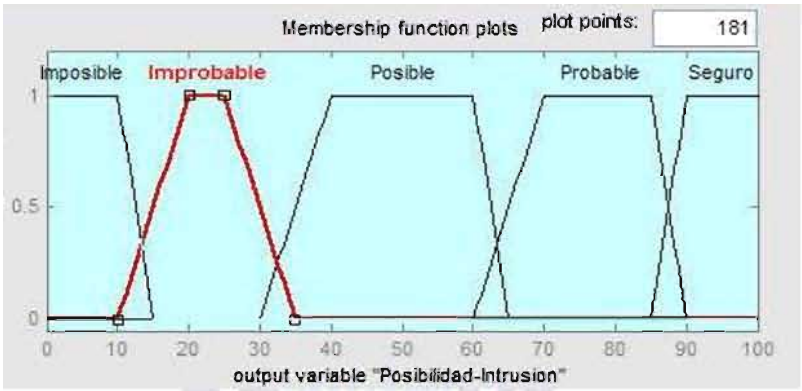


Figura 3.29 *Variable de Salida Posibilidad de Intrusión*

Fuente: *Elaboración Propia*

3.8.3 Elección del Tipo de Sistema de Razonamiento Difuso

Elegimos el sistema difuso tipo *Mandani*, debido a la forma de las entradas y salidas, ya que estas se adecuan a nuestro modelo. También se determino cuatro términos lingüísticos para las variables de entrada y cinco términos lingüísticos para la variable de salida.

3.8.4 Diseño del Conjunto de Reglas Difusas

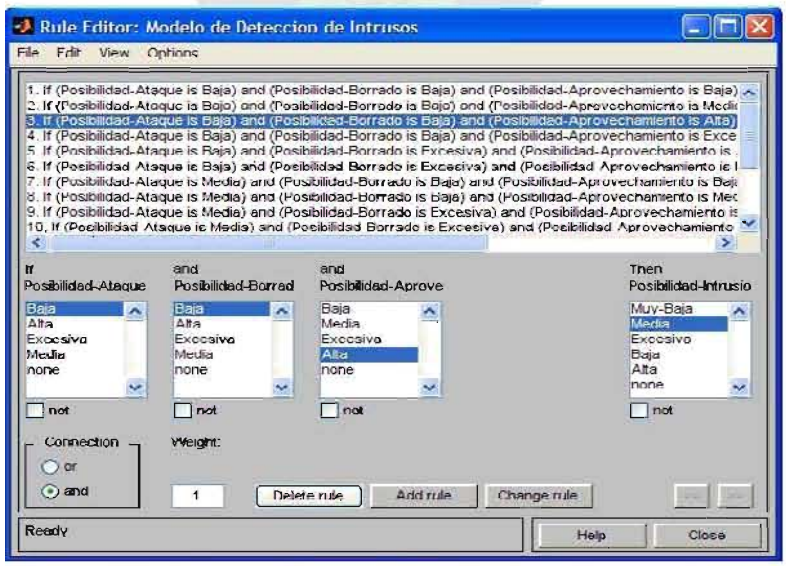


Figura 3.30 *Establecimiento de las Reglas Difusas*

Fuente: *Elaboración Propia*

Se diseño un conjunto determinado de reglas difusas para el modelo planteado. Finalmente el modelo difuso obtenido es el siguiente:

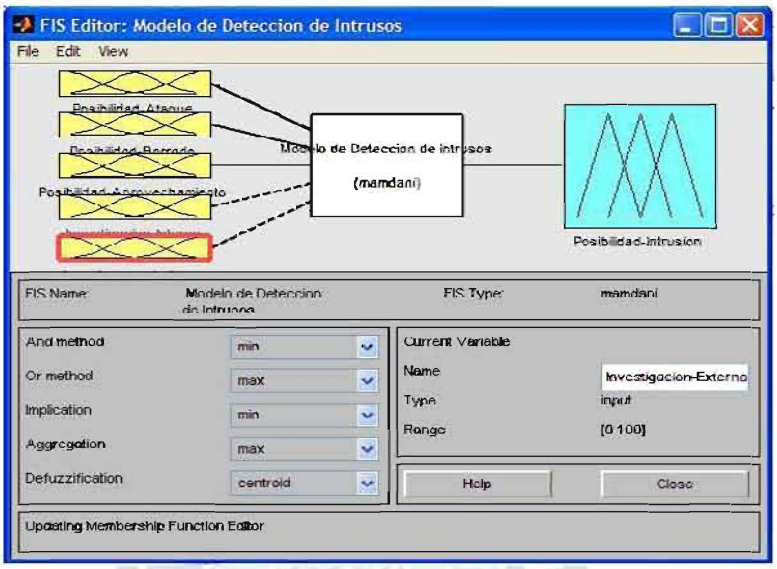


Figura 3.31 *Modelo Difuso para la Detección de Intrusos*

Fuente: *Elaboración Propia*

3.9 Casos de Prueba

Para la evaluación del funcionamiento del modelo difuso, se han efectuado un conjunto de pruebas con la caja de herramientas de lógica difusa de MatLab.

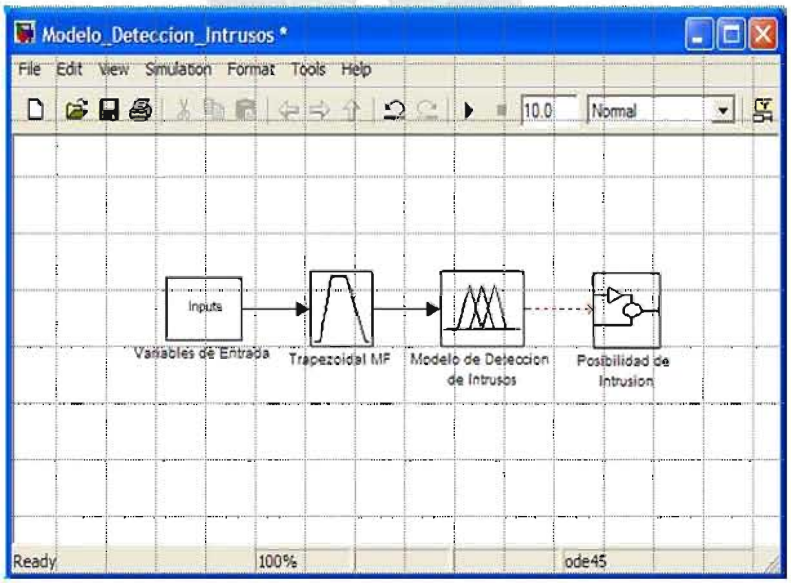


Figura 3.32 *Modelo Difuso para la Detección de Intrusos (Simulink)*

Fuente: *Elaboración Propia*

En primer lugar, se obtuvieron un conjunto de datos del trafico de red del *Institute for Visualization and Perception Research (IVPR),* institución dedicada a la recopilación de datos empleados para temas de investigación. Luego, se realizo un filtrado de los datos obtenidos donde se selecciono 50 registros para la evaluación del modelo. Los siguientes casos de prueba fueron seleccionados por ser los más significativos para evidenciar el funcionamiento del modelo propuesto.

3.9.1 Primer Caso Prueba

Los datos y resultados concretos para esta prueba son los que se indican en el siguiente cuadro:

ENTRADAS	
Consultas DNS	3
Consultas Web	15
Consultas FTP	10
Consultas URL Peligrosas	3
Paquetes ICMP	1
Binds a Puertos	0
Logins Fallidos	1
Comandos enviados en URL	0
Cabeceras de protocolo inconsistentes	1
Logs modificados	0
Cuentas creadas	1
Servicios de red ejecutados	0
Operaciones con Ficheros	0
Uso de recursos	13
Estado del servidor	22
SALIDA	
Posibilidad de Intrusión	**Imposible (6,08)**

Una vez introducida dicha información, se podrá procesar la información y se obtendrá el valor de la variable de salida, en este caso la información acerca la posibilidad de que se esté produciendo una intrusión al sistema es ***imposible***.

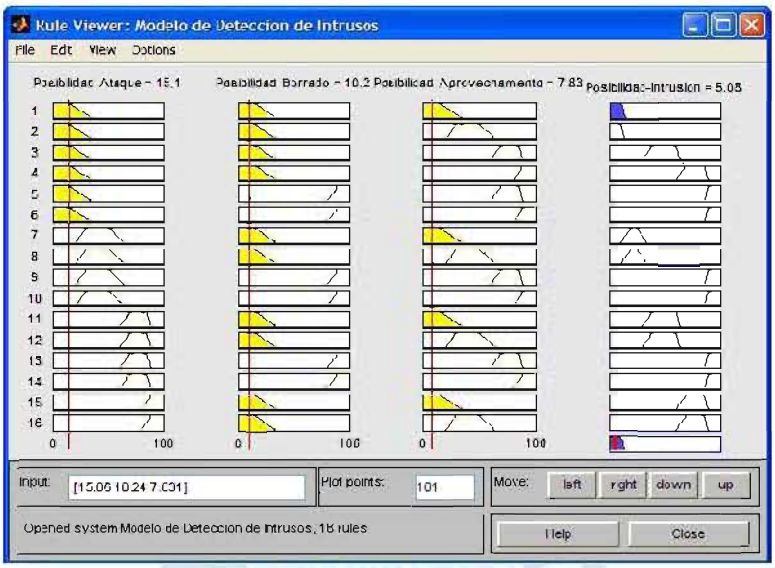


Figura 3.33 *Primer Caso de Prueba para la Evaluación del Modelo*

Fuente: *Elaboración Propia*

3.9.2 Segundo Caso de Prueba

Los datos y resultados concretos para esta prueba son los que se indican en el siguiente cuadro:

ENTRADAS	
Consultas DNS	7
Consultas Web	20
Consultas FTP	19
Consultas URL Peligrosas	5
Paquetes ICMP	10
Binds a Puertos	7
Logins Fallidos	2
Comandos enviados en URL	4
Cabeceras de protocolo inconsistentes	1
Logs modificados	0
Cuentas creadas	0
Servicios de red ejecutados	0
Operaciones con Ficheros	1
Uso de recursos	15
Estado del servidor	50
SALIDA	
Posibilidad de Intrusión	**Improbable** (22,5)

Una vez introducida dicha información, se podrá procesar la información y se obtendrá el valor de la variable de salida, en este caso la información acerca la posibilidad de que se esté produciendo una intrusión al sistema es ***improbable***.

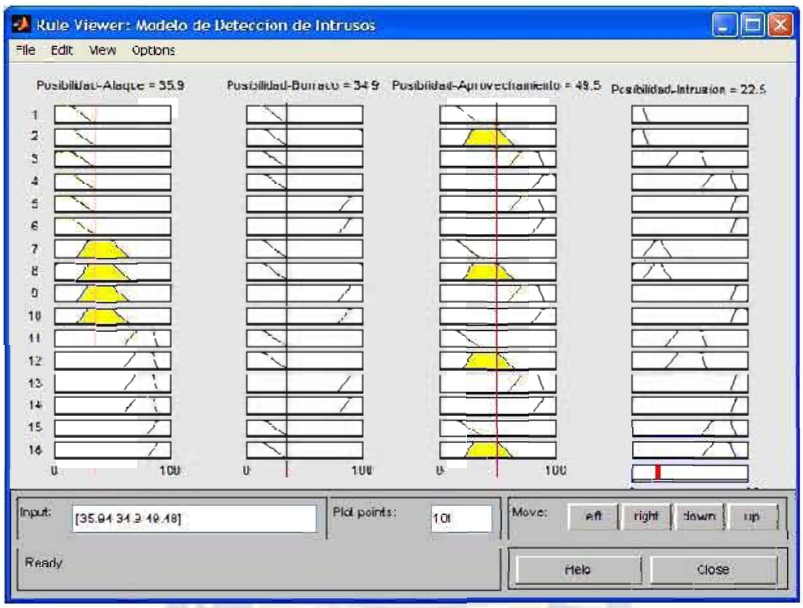


Figura 3.34 *Segundo Caso de Prueba para la Evaluación del Modelo*

Fuente: *Elaboración Propia*

3.9.3 Tercer Caso de Prueba

Los datos y resultados concretos para esta prueba son los que se indican en el siguiente cuadro:

ENTRADAS	
Consultas DNS	10
Consultas Web	17
Consultas FTP	10
Consultas URL Peligrosas	7
Paquetes ICMP	25
Binds a Puertos	10
Logins Fallidos	3
Comandos enviados en URL	7
Cabeceras de protocolo inconsistentes	3
Logs modificados	1
Cuentas creadas	0

Servicios de red ejecutados	1
Operaciones con Ficheros	5
Uso de recursos	40
Estado del servidor	175
SALIDA	
Posibilidad de Intrusión	**Posible (50)**

Una vez introducida dicha información, se podrá procesar la información y se obtendrá el valor de la variable de salida, en este caso la información acerca la posibilidad de que se esté produciendo una intrusión al sistema es ***posible***.

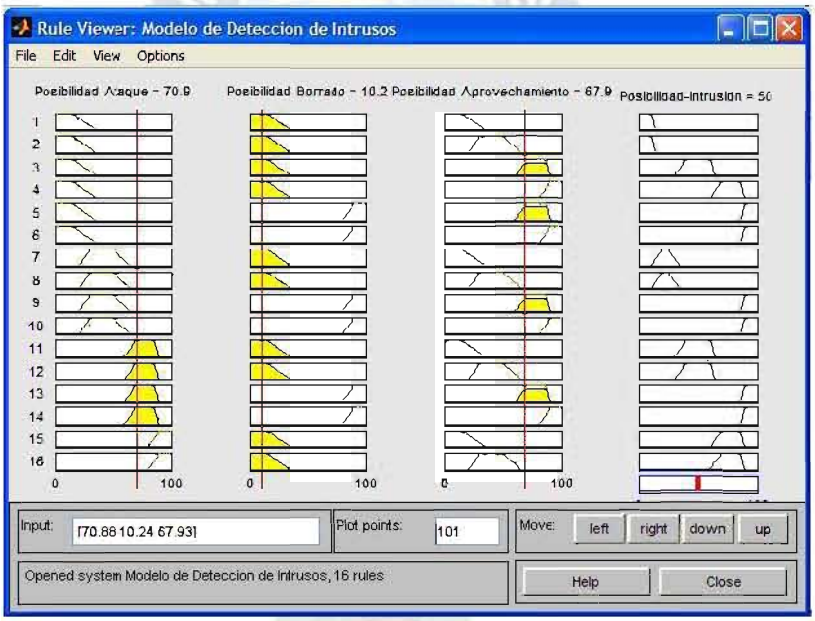


Figura 3.35 *Tercer Caso de Prueba para la Evaluación del Modelo*

Fuente: *Elaboración Propia*

3.9.4 Cuarto Caso de Prueba

Los datos y resultados concretos para esta prueba son los que se indican en el siguiente cuadro:

ENTRADAS	
Consultas DNS	15
Consultas Web	10
Consultas FTP	25
Consultas URL Peligrosas	10
Paquetes ICMP	30
Binds a Puertos	15
Logins Fallidos	3
Comandos enviados en URL	10
Cabeceras de protocolo inconsistentes	4
Logs modificados	1
Cuentas creadas	1
Servicios de red ejecutados	2
Operaciones con Ficheros	9
Uso de recursos	70
Estado del servidor	300
SALIDA	
Posibilidad de Intrusión	**Probable (76%)**

Una vez introducida dicha información, se podrá procesar la información y se obtendrá el valor de la variable de salida, en este caso la información acerca la posibilidad de que se esté produciendo una intrusión al sistema es ***probable.***

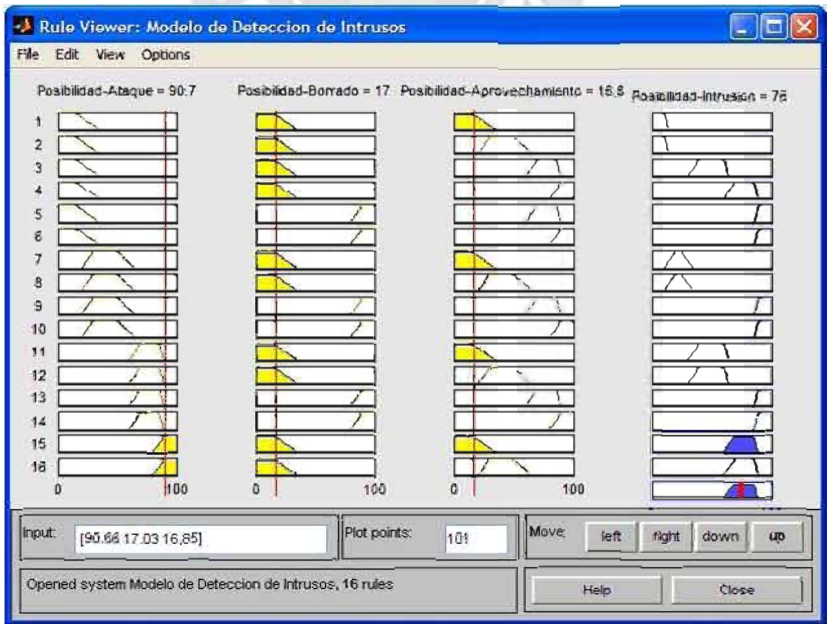


Figura 3.36 *Cuarto Caso de Prueba para la Evaluación del Modelo*

Fuente: *Elaboración Propia*

3.10 Resultados

Para comprobar el correcto funcionamiento del modelo, se realizaron pruebas con datos de flujo de red normal y de ataque. En las pruebas realizadas, se obtuvieron los siguientes resultados:

- Al iniciar el proceso de evaluación, los datos de entrada fueron de un flujo normal de información, para lo cual el modelo difuso determino la posibilidad de intrusión como ***imposible***, evitando de esta manera producir un *falso positivo*.

- Posteriormente se evaluó el modelo con datos de entrada de un flujo de ataque para lo cual el modelo difuso determino la posibilidad de intrusión como ***probable***, evitando de esta manera producir un *falso negativo*.

Los resultados obtenidos en las pruebas, demuestran que el modelo difuso determina la posibilidad de intrusión de un usuario a un sistema informático, a partir de variables utilizadas en la detección de intrusos.

3.11 Análisis de los Resultados

Para realizar el análisis de los resultados del modelo se calculo el error que producen las salidas generadas por el modelo planteado.

$$E_P = \frac{1}{2}\sum_{k}(d-y)^2$$

Donde $\boldsymbol{d}$ representa a la salida esperada y $\boldsymbol{y}$ es la salida obtenida, entonces se calcula $\boldsymbol{(d-y)^2}$ para cada una de las salidas obtenidas, como se observa en el siguiente cuadro:

N° de Prueba	D	y	$(d-y)$	$(d-y)^2$
1	6.05	6.08	-0.03	0.0009
2	22.45	22.5	0.05	0.0025
3	49.96	50	0.04	0.0016
4	76.01	76	-0.01	0.0001
5	55.05	55	0.05	0.0025
6	12.70	12.69	0.01	0.0001
7	95.10	95.1	0.00	0.0000
8	11.63	11.7	0.07	0.0049
9	19.98	20.01	-0.03	0.0009
10	88.07	88.09	-0.02	0.0004
11	68.53	68.58	-0.05	0.0025
12	82.90	82.99	-0.09	0.0081
13	76.30	76.35	-0.05	0.0025
14	58.36	58.42	-0.06	0.0036
15	17.97	17.90	0.07	0.0049
16	10.66	10.61	0.05	0.0025
17	44.1	44.09	0.01	0.0001
18	21.36	21.15	0.21	0.0441
19	74.23	74.28	-0.05	0.0025
20	11.59	11.52	0.07	0.0049
21	66.98	66.89	0.09	0.0081
22	96.9	96.87	0.03	0.0009
23	52.01	51.97	0.04	0.0016
24	32.85	32.90	-0.05	0.0025
25	35.44	35.45	-0.01	0.0001
26	45.09	45.1	-0.01	0.0001
27	51.9	51.85	0.05	0.0025
28	64.78	64.72	0.06	0.0036
29	74.23	74.32	-0.09	0.0081
30	17.1	17.15	-0.05	0.0025
31	62.7	62.65	0.05	0.0025
32	95.65	95.54	0.11	0.0121
33	28.21	28.11	0.1	0.0100
34	41.2	41.33	-0.13	0.0169
35	90.35	90.32	0.03	0.0009
36	45.12	45.09	0.03	0.0009
37	16.96	16.92	0.04	0.0016
38	32.14	32.09	0.05	0.0025
39	19.83	19.78	0.05	0.0025
40	26.13	25.99	0.06	0.0036
41	30.15	**30.14**	0.01	0.0001
42	52.11	**52.07**	0.04	0.0016
43	10.36	**10.4**	-0.04	0.0016
44	7.90	7.95	-0.05	0.0025
45	67.54	67.57	-0.03	0.0009
46	30.73	30.70	0.03	0.0009
47	10.52	10.49	0.03	0.0009
48	14.5	14.56	-0.06	0.0036
49	74.58	74.6	-0.02	0.0004
50	53.96	53.90	0.06	0.0036

Figura 3.28 *Calculo del error que produce el modelo*

Fuente: *Elaboración Propia*

Luego se obtiene $\sum_k (d-y)^2$, donde $k=1, 2.... 50$ debido a que son 50 los registros de flujo de red que fueron analizados, por tanto:

$$\sum_k (d-y)^2 = 0.16$$

Finalmente tenemos:

$$E_P = \frac{1}{2}\sum_k (d-y)^2 = 0.08$$

El error que produce el modelo permite conocer el porcentaje de precisión del mismo:

$$Precisión = (1 - E_P) \times 100\% = 92\%$$

Capítulo IV

Conclusiones y Recomendaciones

Resumen

En este capítulo se resumen las conclusiones a las que se llegan al realizar la investigación, el cumplimiento de los objetivos planteados, también se describe el aporte del trabajo y por último se realizan recomendaciones para trabajos futuros.

4.1 Conclusiones

Con todo lo expuesto en los anteriores capítulos se ha llegado a las siguientes conclusiones:

- Respecto al objetivo general, se ha podido realizar un modelo para determinar la posibilidad de intrusión de un usuario a un sistema informático, a partir de variables utilizadas en la detección de intrusos.

- Para los objetivos específicos:

  - Se realizó un estudio de los sistemas de detección de intrusos, para la elaboración del modelo.

  - Se identificó la estructura superficial del modelo a desarrollar, determinando las variables de entrada y salida.

- Se formalizó el conocimiento obtenido haciendo uso del modelado difuso, estableciendo el universo del discurso de cada variable para determinar sus funciones de pertenencia.

- Se implementó el modelo en MatLab 7.6 con la caja de herramientas de lógica difusa para la evaluación del mismo.

- Se demostró la aplicabilidad de la lógica difusa para la Detección de Intrusos en la Seguridad de Sistemas Informáticos.

4.2 Estado de la Hipótesis

Las pruebas realizadas mostraron que el modelo planteado para la detección de intrusos, interpreta la información recibida 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 una vez efectuado una descripción del tópico de estudio, se presento el modelo propugnado, analizando las fases de su diseño e implementación práctica, donde se evidencia la validez del enfoque propuesto.

La hipótesis planteada en la investigación, se logro demostrar con los resultados obtenidos por el modelo. En base a estos resultados, se concluye que: *La aplicación de la Lógica Difusa, si determina la posibilidad de intrusión de un usuario a un sistema informático.*

4.3 Recomendaciones

El desarrollo del presente trabajo no sólo ha permitido obtener resultados de interés, sino que también ha abierto nuevos caminos cuya exploración parece ser prometedora para desarrollos futuros, mencionándose aquí tan sólo dos líneas de investigación y desarrollo al respecto:

i. Ampliar la base de expertizaje, tanto en cantidad como en calidad de la información y el conocimiento requeridos.

ii. Avanzar en el estudio de las técnicas de ajuste de la solución del modelo difuso con la aplicación de las Redes Neuronales Artificiales como herramienta para el aprendizaje de la base de reglas difusas.

La unión de la lógica difusa con otras técnicas de la inteligencia artificial, permite que las aplicaciones sean más adecuadas y precisas.

Referencias

Referencias Bibliográficas

[AXELSSON99] Axelsson, S. (1999b). Research in Intrusion-Detection Systems: A Survey. Department of Computer Engineering, Chalmers University of Technology, Goteborg, Sweden.

[BEZDEK91] Bezdek, J. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press. New York, 1981

[BRIDGES00] Bridges, S. M. and Vaughn, R. B. Intrusion Detection via fuzzy Data Mining. Annual Canadian Information Technology Security Symposium. 2000.

[BRIO01] Bonifacio Martin del Brio, Alfredo Sanz Molina. Redes Neuronales y Sistemas Difusos Segunda Edición Ampliada

[CASTILLO97] Castillo, E., Gutierrez, J., Hadi, A. (1997). *Sistemas expertos y modelos de redes probabilisitcas*. España: Universidad de Cantabria.

[CHOQUE02] Choque Aspiazu Guillermo. Inteligencia Artificial Perspectivas y Realizaciones. La Paz- Bolivia. 2002

[DENNING82] Dorothy E. Denning. Cryptography and Data Security. Addison-Wesley, Reading,Massachusetts, 1982.

[DYCHKHOFF94] Dichkhoff, H & W. Pedrycz. Generalized Means as a Model of Compensative Connectives, Fuzzy Sets and Systems 14.

[DUBOIS90] Dubois, D., Prade, H. *Fuzzy sets and systems: Theory and applications*, Academic Press, 1990.

[GALINDO03] Galindo Gomez J. (2003), Conjuntos y Sistemas Difusos, Departamento de Lenguajes y Ciencias de la Computación, Universidad de Malaga.

[GARFINKEL99] Garfinkel Simson and Gene Spafford. *Practical UNIX Security*. O'Reilly and Associates, Sebastopol, California, 1999.

[KANTROWITZ95] Kantrowitz, M. *et al*, *FAQ: Fuzzy Logic and Fuzzy Expert Systems*, disponible en *ftp.cs.cmu.edu:/user/ai/pubs/faqs/fuzzy/fuzzy.faq*.

[KUMAR00] Sandeep Kumar. Classification and Detection of Computer Intrusions. PhD thesis,Purdue University, West Lafayette, IN, USA, Aug 2000.

[LINDQVIST00] U. Lindqvist and P.A. Porras, Detecting Computer and Network Misuse Through the Production-Based Expert, May 2000.

[LEXUS06] Diccionario Enciclopedico Lexux Color, Lexus Editores, 2006

[MARTINEZ06] Asier Martínez Martínez. Técnicas de Detección de Intrusiones en Redes 802.11. (*Noviembre de 2006*.)

[NORTCUFF01] Nortcuff, S. and Novak, J. (2001). "Detección de intrusos. Guía avanzada". Prentice Hall (2ª Edición)

[NSA89] National Security Agency. A Guide to Understanding Identificationand Authentication in Trusted Systems NCSC-TG-017 Library No. 5-235,479

[OLIVARES03] Olivares S. Manuel Rojas (2003), Modelado y Control Difuso, Departamento de Electrónica Universidad Técnica Federico Santa María.

[RANUM03] Marcus J. Ranum, "*Intrusion detection: challenges and myths*", Network Flight Recorder, Inc., March 2003.

[RUSELL97] Deborah Rusell and G. T. Gangemi Sr. Computer Security Basics. O'Reilly & Associates, Inc., Sebastopol, California, December 1991.

[URKO04] Urko Zurutuzaga Ortega. Sistemas de Detección de Intrusos como Elemento de Seguridad Perimetral. *(Octubre 2004.)*

[TSUKAMOTO03] *An approach to fuzzy reasoning method.* Incluido en "Advances in Fuzzy Set Theory and Applications" GUPTA, M., RAGADE, R., YAGER, R.

[ZADEH65] Zadeh, L. Fuzzy sets, *Information & Control.*, 8, 1965.

Referencias Electronicas

[ACMT10] *Conceptos sobre IDS*
http://www.acm.org/crossroads/xrds2-4/intrus.html

[CERT10] Computer Emergency Response Team/Coordination Center, statistics.
http://www.cert.org/stats/ *(Octubre 2010)*

[CSI09] Computer Security Institute / Datacenter
http://www.csi.org/data/ *(Septiembre 2010)*

[DESAI07] Desai, Neil. (2007). *Intrusion Prevention Systems:* the Next Step in the Evolution of IDS. Security Focus.
http://www.securityfocus.com/infocus/1670

[ISIS05] Fuzzy Logic and Neurofuzzy Resources
http://www.isis.ecs.soton.ac.ik/fuzzy.html

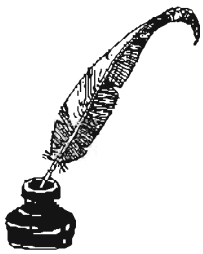
[LAPRENSA09] La prensa edición virtual
http://www.laprensa.com.bo/noticias/07-12-09/index.php

[LINUXFOCUS03] *Tipos de ataques.* http://www.linuxfocus.org/Castellano/March2003/article282.shtml

[KANTROWITZ95] Kantrowitz, M. *et al*, *FAQ: Fuzzy Logic and Fuzzy Expert Systems*, disponible en *ftp.cs.cmu.edu:/user/ai/pubs/faqs/fuzzy/fuzzy.faq*.

[SUN03] *Tutorial de Base* http://www.sun.com/bigadmin/jsp/descFile.jsp?url=descAll/analyzing_snort_dat

[SYMANTEC10] Informe sobre las amenazas a la seguridad en la red, desarrollado por la empresa de software antivirus Symantec *(Enero 2010)* http://www.symantec.com/about/news/release/article.jsp?prid=20100922502.



ANEXOS

ANEXOS

Sistemas de Detección de Intrusos

Introducción

Existen numerosas medidas de seguridad para proteger los recursos informáticos de una empresa, pero aunque se sigan todas las recomendaciones de los expertos, no estaremos libres de posibles ataques con éxito. Esto se debe a que conseguir un sistema virtualmente invulnerable es sumamente costoso, además de que las medidas de control reducirían la productividad de la empresa.

Dentro de las soluciones tecnológicas que en la actualidad están disponibles para reforzar la seguridad de una red, los firewalls son muy populares. Un firewall es un sistema encargado del cumplimiento de las políticas de control de acceso a la red, lo cual se hace a través de reglas. Un firewall actúa como guardia perimetral de una red: protege una red de ataques que provengan del exterior de ésta. Pero el escenario se puede complicar de la siguiente forma:

1. Un atacante puede lograr pasar el firewall, dejando la red a su merced.
2. Un firewall protege de los accesos no autorizadas hacia la red interna, pero no protege a las máquinas ubicadas en la red perimetral como servidores web, servidores de correo, servidores FTP, en otras palabras, a las bases funcionales de Internet.
3. Un firewall no protege contra ataques desde adentro.

En estos casos lo que nos queda detectar el ataque o la intrusión lo antes posible para que cause el menor daño en el sistema. Antes de continuar vamos a definir qué se entiende normalmente por intrusión. Normalmente un intruso intenta:

- Acceder a una determinada información.

- Manipular cierta información.
- Hacer que el sistema se no funcione de forma segura o inutilizarlo.

Una intrusión es cualquier conjunto de acciones que puede comprometer la integridad, confidencialidad o disponibilidad de una información o un recurso informático. Los intrusos pueden utilizar debilidades y brechas en la arquitectura de los sistemas y el conocimiento interno del sistema operativo para superar el proceso normal de autenticación.

La detección de intrusos se puede detectar a partir de la caracterización anómala del comportamiento y del uso que hacen de los recursos del sistema. Este tipo de detección pretende cuantificar el comportamiento normal de un usuario. Para una correcta distinción hay que tener en cuenta las tres distintas posibilidades que existen en un ataque, atendiendo a quién es el que lo lleva a cabo:

- *Penetración externa.* Que se define como la intrusión que se lleva a cabo a partir un usuario o un sistema de computadores no autorizado desde otra red.
- *Penetraciones internas.* Son aquellas que llevan a cabo por usuarios internos que no están autorizados al acceso.
- *Abuso de recursos.* Se define como el abuso que un usuario lleva a cabo sobre unos datos o recursos de un sistema al que está autorizado su acceso.

La idea central de este tipo de detección es el hecho de que la actividad intrusiva es un subconjunto de las actividades anómalas. Esto puede parecer razonable por el hecho de que si alguien consigue entrar de forma ilegal en el sistema, no actuará como un usuario normal. Sin embargo en la mayoría de las ocasiones una actividad intrusiva resulta del agregado de otras actividades individuales que por sí solas no constituyen un comportamiento intrusivo de ningún tipo. Idealmente el conjunto de actividades anómalas es el mismo del conjunto de actividades intrusivas, de todas formas esto no siempre es así:

1. *Intrusivas pero no anómalas.* Se les denomina *falsos negativos* y en este caso la actividad es intrusiva pero como no es anómala y no se consigue detectarla. Se denominan *falsos negativos* porque el sistema erróneamente indica ausencia de intrusión.
2. *No intrusivas pero anómalas.* Se denominan *falsos positivos* y en este caso la actividad es no intrusiva, pero como es anómala el sistema decide que es intrusiva. Se denominan *falsos positivos*, porque el sistema erróneamente indica la existencia de intrusión.
3. *Ni intrusiva ni anómala.* Son negativos verdaderos, la actividad es no intrusiva y se indica como tal.
4. *Intrusiva y anómala.* Se denominan positivos verdaderos, la actividad es intrusiva y es detectada.

Los primeros no son deseables, porque dan una falsa sensación de seguridad del sistema y el intruso en este caso puede operar libremente en el sistema. Los falsos positivos se deben de minimizar, en caso contrario lo que puede pasar es que se ignoren los avisos del sistema de seguridad, incluso cuando sean acertados. Los detectores de intrusiones anómalas requieren mucho gasto computacional, porque se siguen normalmente varias métricas para determinar cuánto se aleja el usuario de lo que se considera comportamiento normal.

Hoy día existen en el mercado una buena cantidad de productos conocidos como SDI (Sistemas de Detección de Intrusos) o en inglés IDS (Intrusión Detection System).

Estos sistemas basan su funcionamiento en la recolección y análisis de información de diferentes fuentes, que luego utilizan para determinar la posible existencia de un ataque o penetración de intrusos.

En caso de que exista la suficiente certeza de la detección de un incidente, el SDI tiene como función principal alertar al administrador o personal de seguridad, para que tome acciones al respecto. Otras implementaciones más complejas son capaces de ir más allá

de la notificación de un posible ataque, es decir pueden ejecutar acciones automáticas que impidan el desarrollo de éste.

Clasificación de los SDI

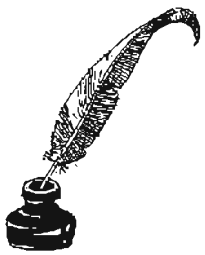
Los SDI pueden clasificarse en base a varios aspectos: método de detección, tipo de monitoreo y forma de recolección y análisis de la información.

- Según el método de detección, los hay de detección de mal uso y detección de anomalías.
- Según el tipo de monitoreo, hay SDI con detección orientada al host o detección orientada a la red.

Características deseables de un SDI

1. Debe ejecutarse continuamente sin intervención o supervisión de un operador humano.
2. Debe ser confiable, lo suficiente como para ejecutarse en background, pero no debe ser una caja negra, es decir, que su funcionamiento interno pueda ser examinado.
3. Debe ser capaz de tolerar fallas, en el sentido de que pueda sobrevivir a una caída del sistema, sin tener que reconstruir su base de datos de conocimientos al reiniciarse.
4. El sistema debe estar en capacidad de automonitorearse para asegurar su correcto funcionamiento.
5. Debe ser ligero, es decir su ejecución no debe cargar al sistema de una manera tal que le impida ejecutar otras tareas con relativa normalidad
6. Debe observar desviaciones del comportamiento estándar.
7. Debe poder adaptarse al comportamiento cambiante del sistema, es decir, si la configuración del sistema cambia, el SDI se adaptará.
8. Debe ser difícil de engañar.

DOCUMENTACION



La Paz, 14 de Noviembre de 2011

Señor
Lic. Eufren Llanque Quispe
Docente Taller de Licenciatura II
Carrera de Informática
Presente.-

Ref.: Conformidad y Aval de Trabajo de Tesis de Grado

Distinguido licenciado:

Tengo a bien dirigirme a su persona para darle a conocer que luego de realizar el seguimiento y la revisión del documento final de la tesis de grado y el prototipo de la misma bajo el título: ***"Modelo para Detección de Intrusos en la Seguridad de Sistemas Informáticos aplicando Lógica Difusa"***, elaborado por la postulante Ines Margarita Ramos, quien habiendo realizado las respectivas correcciones a mis observaciones, tanto en la documentación como en el prototipo y haber demostrado conocimiento y dominio del tema, me corresponde dar mi conformidad, recomendando que la mencionada universitaria inicie sus correspondientes tramites, para su respectiva solicitud de defensa de tesis de grado.

Sin otro particular, saludo a su persona con las consideraciones más distinguidas.

Atentamente,

Lic. Carlos Mullisaca Choque
Docente Revisor

La Paz, 25 de Noviembre de 2011

Señora
Lic. Menfy Morales R.
Jefe Carrera de Informática
Presente.-

Ref.: Aval para Defensa Final de Tesis de Grado

De mi consideración:

En mi calidad de Tutor Colectivo de la materia de Taller de Licenciatura II tengo a bien presentar el Aval de la Tesis de Grado de la Srta. Ines Margarita Ramos desarrollado bajo la denominación "***Modelo para la Detección de Intrusos en la Seguridad de Sistemas Informáticos aplicando Lógica Difusa***" dando cumplimiento al reglamento y sustentación del Trabajo de Tesis de Grado.

Efectuado el trabajo de revisión y las orientaciones pertinentes de la Tesis de Grado se ha constatado que la postulante ha completado el documento que responde a los objetivos planteados. Por todo lo expuesto estimo procedente la otorgación del aval para la continuación de los trámites para la presentación y sustentación de la defensa final de la Tesis de Grado.

Es cuanto informo a su autoridad para fines consiguientes, sin otro particular saludo a su persona con las consideraciones más distinguidas.

Atentamente,

Lic Eufren Llanque Quispe
Docente Taller de Licenciatura II