Management Intelligent Systems Group
http://sicodinet.unileon.es

Propuesta de Módulo del Proyecto OSSIM: Desarrollo de un SEB para la Priorización y Valoración de Riesgos (*)

Enrique López González, Carlos Caño Alegre, Pedro Casis García, Sergio González García

Technical Report #MISYG-2004-05
September, 2004

(*) *Este trabajo está soportado por el proyecto de investigación DPI 2001–0105 del MCT.*

Dept. of Management and Economics Business
Business and Economics Sciences Faculty. University of León
E-24071 León, SPAIN
Tel. & Fax: +34 987 291742

Índice

Introducción

"En el éxito de una intrusión, una vez franqueadas las defensas perimetrales se sucede el compromiso de decenas de máquinas. Existe un flujo de conexiones permanentes y duraderas durante varias horas, conexiones anómalas que dibujan un camino completamente contrario a lo que debería ser aceptable; procedentes desde el exterior crean un puente de entrada a la red interna donde una tras otra van comprometiendo más máquinas trazando un camino cada vez más anómalo, y peligroso..

Los usuarios y administradores de la organización víctima, que en ese momento se encuentran trabajando en las máquinas nunca notan nada extraño en el momento, y rara vez localizan el ataque a posteriori."

La Seguridad Informática se está convirtiendo en estos últimos años en uno de los sectores más importantes de este campo, ya que no hay que olvidarlo, afecta de manera crítica a todas las demás áreas. Debemos pues tratar de intentar garantizar a las organizaciones redes corporativas seguras, aunque esto no es un problema trivial.

Siempre habrá hackers capaces de romper los sistemas más seguros, cada día aparecen nuevos agujeros de seguridad y virus informáticos más peligrosos, la seguridad informática está en acelerada evolución; los sistemas que hoy son seguros quizá mañana no lo sean.

La sofisticación de la tecnología actual hace posible, con los sistemas de detección de intrusos, detectar los eventos más concretos que ocurren en el sistema. Sin embargo, y debido a la cantidad de alertas recibidas y a la poca fiabilidad de éstas se provocan demasiados falsos positivos (alertas falsas). Al no haber capacidad de abstracción, debido a la información excesivamente concreta y parcial, se producen de igual modo falsos negativos (ataques no detectados).

	Propiedad	Efecto ante su ausencia
Fiabilidad	El grado de certeza que nos ofrece nuestro detector ante el aviso de un posible evento.	Falsos Positivos
Sensibilidad	La capacidad de análisis, en profundidad y complejidad, que posee nuestro detector a la hora de localizar un posible ataque	Falsos Negativos

Para solucionar estos problemas se hace necesario proveer a los administradores de herramientas de seguridad que proporcionen la información necesaria y actualizada de los eventos que ocurren en el sistema. Pero esto no es suficiente, en los sistemas actuales y con las técnicas de ataque que hay hoy en día resulta prácticamente imposible para un administrador asimilar toda la información relativa a la seguridad de su sistema. Aparece la necesidad de implementar herramientas que recopilen y analicen la información de la red, creando una capa intermedia entre los sistemas y el administrador, con la capacidad de abstracción suficiente.

Estas Frameworks recopilaran toda la información del sistema que sea relevante para su seguridad pero, además, se encargaran de procesarla y evaluarla para proporcionar al administrador información "útil" y global del sistema. Ofreciendo así

una visión de la red con un grado de abstracción que permita una revisión práctica y asumible para los administradores.

En este contexto se enmarca el proyecto OSSIM (Open Source Security Information Management) que pretende responder a estas necesidades y a otras que vayan apareciendo. En lo que sigue se explicará con el suficiente detalle el significado del proyecto OSSIM. Además se especificará el diseño de un sistema borroso que implementa una parte fundamental de este proyecto. Con todo esto se pretende demostrar que la Lógica Difusa puede ser una herramienta altamente útil en el diseño de sistemas de seguridad informática.

OSSIM

OSSIM es una distribución de productos open source integrados para construir una infraestructura de monitorización de seguridad. Su objetivo es ofrecer un marco para centralizar, organizar y mejorar las capacidades de detección y visibilidad en la monitorización de eventos de seguridad de la organización.

Este proyecto tiene la intención de mejorar la seguridad de la organización con la integración de los mejores productos desarrollados dentro del software libre en estos años en un Framework general, que aumenta sus posibilidades al interrelacionar todas sus funcionalidades.

OSSIM no quiere ser un producto, sino una solución, un sistema personalizado para las necesidades de cada organización formado por la conexión e integración de varios módulos especialistas.

Entre las herramientas de código libre integradas se incluyen algunas de las desarrolladas por los mejores especialistas del mundo (snort, nessus, rrd, nmap, ntop,...) y dada su naturaleza de software libre, probadas por miles de personas no se puede dudar de su robustez y auditabilidad.

En OSSIM destacan dos funciones:

- **Correlación**
- **Valoración de riesgos**

La correlación proporciona la visibilidad de todos los eventos de los sistemas en un punto y con un mismo formato y a través de esta situación privilegiada relacionar y procesar la información permitiendo aumentar la capacidad de detección, priorizar los eventos según el contexto en que se producen, y monitorizar el estado de seguridad de la red.

La Valoración de Riesgos es forma de decidir en cada caso la necesidad de ejecutar una acción a través de la valoración de la amenaza que representa un evento frente a un activo, teniendo en cuenta la fiabilidad y probabilidad de ocurrencia de este evento.

OSSIM consta de tres partes fundamentales y diferenciadas: Herramientas de Monitorización, preprocesadores y funciones de postproceso.

Las herramientas de monitorización son el Cuadro de Mandos con visibilidad de alto nivel, Monitores de Riesgo y Comportamiento de nivel medio y Consola Forense y Monitores de Red con visión de bajo nivel.

Los preprocesadores son los detectores y monitores conocidos y usados por la mayoría de administradores (IDS, detectores de anomalías, Firewalls y monitores varios).

El postproceso es lo que hace del OSSIM una solución nueva, altamente fiable y que mejora la capacidad de detección de sus preprocesadores. Además de las dos funciones anteriormente descritas (correlación y valoración de riesgos) se añade la de priorización, que posteriormente describiremos con detalle.

El objetivo fundamental del proyecto OSSIM es aumentar la capacidad de detección, que depende de la sensibilidad y la fiabilidad de los detectores. La detección se inicia con la generación de alertas por los detectores, **Preproceso**, a continuación la

información generada por todos los detectores se envía a punto central, **Colección**, por último se procede al tratamiento de esa información centralizada, **Postproceso**.

En el siguiente gráfico extraído de la descripción del OSSIM se puede observar las funcionalidades del sistema, vistas como una serie de capas.

En este trabajo se tratara con más detalle las funciones de postproceso, aunque no se debe olvidar que el proyecto OSSIM es un Framework que integra todas capas mencionadas, cada una de las partes es igual de necesaria para el funcionamiento global.

Lógica Difusa aplicada al OSSIM

Los Sistemas Borrosos basados en lógica difusa resultan especialmente útiles en la solución de problemas que por su naturaleza tienen una alta incertidumbre, frente a la certeza y poca flexibilidad de los sistemas tradicionales. Con los números borrosos se puede representar los eventos como se muestran en la realidad, sin que la representación conlleve una pérdida de información.

En vista de estas ventajas, la Lógica Difusa representa una herramienta muy valiosa que puede ser utilizada con éxito en el marco de la seguridad informática y, en este caso, en el proyecto OSSIM.

Aunque puede ser de gran utilidad en la mayoría de módulos que integran el OSSIM, ahora nos centraremos en la fase de Postproceso. En esta tabla se incluyen las funciones del Postproceso, el proceso que realizan y el efecto que tienen en el sistema.

	Proceso	Efecto
Priorización	Valoración de la amenaza mediante contextualización de un evento	Aumenta la Fiabilidad
Valoración de Riesgo	Valoración del Riesgo respecto del valor de activos	Aumenta la Fiabilidad
Correlación	Relación de varios eventos para obtener una información de mayor valor	Aumenta la Fiabilidad, Sensibilidad, y Abstracción

El Postproceso conlleva un análisis mucho más complejo que la mera detección de intrusos o de anomalías, es el encargado de priorizar y valorar el riesgo de cada evento y, además, interrelacionarlos para poder obtener información de mayor nivel de abstracción (Correlación).

El sistema se nutre de las alertas que los subsistemas de nivel más bajo van produciendo y dará, como resultado, alarmas, de mayor grado de abstracción en los niveles altos.

La utilización de Sistemas Borrosos para implementar la fase de Postproceso resulta una manera natural y eficaz para solucionar el problema, representando las situaciones tal como ocurren en el sistema y aportando la flexibilidad necesaria.

Desarrollaremos un Sistema Borroso que implemente los módulos de Priorización y Valoración de Riesgos. Aunque de igual modo se podría aplicar la Lógica Difusa para implementar la fase de Correlación.

A continuación se explica con más detalle que significan y como se desarrollan la Priorización y la Valoración de Riesgos, tal y como se describen en el OSSIM.

Priorización

Llamamos priorización al proceso de contextualización, es decir la evaluación de la importancia de una alerta respecto del escenario de nuestra organización. Este escenario está descrito en una base de conocimiento sobre la red del cliente formada por:

- *Inventario de Máquinas y Redes (identificadores, s. operativo, servicios, etc.)*
- *Política de Accesos (desde donde a donde está permitido o prohibido)*

Para realizar estas tareas (así como la valoración de riesgos explicada en el siguiente apartado) disponemos de un Framework donde podremos configurar:

1. *Política de Seguridad. O valoración de parejas activo-amenazas según la Topología y flujo de los datos.*
2. *Inventario*
3. *Valoración de activos*
4. *Valoración de amenazas (priorización de alertas)*
5. *Valoración de Fiabilidad de cada alerta*
6. *Definición de Alarmas.*

*Extractos de las especificaciones del OSSIM

Valoración de Riesgos

La importancia que debemos dar a un evento debe ser dependiente de estos tres factores:

a. El valor del Activo al que el evento se refiere

b. La Amenaza que representa el evento

c. La Probabilidad de que este evento ocurra

En nuestro caso, debido a la capacidad de medir en tiempo real, podremos medir el riesgo asociado a la situación actual, en términos instantáneos.

En este caso el riesgo será medido como la medida ponderada del daño que produciría y la probabilidad de que este ocurriendo en este momento la amenaza.

Esta probabilidad, derivada de la imperfección de nuestros sensores, no será más que el grado de fiabilidad de estos en la detección de la posible intrusión en curso.

Llamaremos Riesgo Instantáneo a la situación de riesgo producida por la recepción de una alerta, valorada de forma instantánea como la medida ponderada entre el daño que produciría el ataque y la fiabilidad del detector que lo reporta.

OSSIM calculará el Riesgo Instantáneo de cada evento recibido que será la medida objetiva que utilizaremos para valorar la importancia que un evento puede implicar en términos de seguridad, sólo a través de esta medida valoraremos la necesidad de actuar.

Incluiremos en nuestro sistema así mismo un Monitor de Riesgos descrito posteriormente que valorará el riesgo acumulado en el tiempo de redes y grupos de máquinas relacionados en un evento.

*Extractos de las especificaciones del OSSIM.

Diseño del Sistema Borroso

Ahora describiremos como se ha desarrollado el Sistema Borroso que intenta resolver el problema de la Priorización y la Valoración de Riesgos. En estas líneas se detallaran las variables de entrada al sistema, las utilizadas de manera intermedia y las variables de salida. También se describen las bases de reglas del sistema.

Este sistema tiene dos partes bien diferenciadas:

- Priorización
- Valoración de Riesgos

Aunque cada parte tiene sus características propias, inherentes a cada problema, se deben de ejecutar de manera secuencial, puesto que para valorar los riesgos de un evento del sistema antes debe ser priorizado. Por tanto, tendremos que la salida del subsistema de priorización será entrada del subsistema de valoración de riesgos.

Subsistema de Priorización

Antes se ha descrito esta parte del sistema según las especificaciones que aparecen en el OSSIM. En este trabajo se han adaptado las variables implicadas en el problema, ya que hay que afrontar las limitaciones que conlleva desarrollar estos módulos de manera independiente. A pesar de todo, se ha intentado ser lo más fiel posible a la especificación del OSSIM.

Este subsistema tendrá las siguientes entradas:

- El evento, que es representado por los ataques producidos
- Inventario de Máquinas y Redes, en nuestro caso se considerará los datos de una máquina o red en cada caso.
- Política de Accesos, los privilegios de los que goza el atacante

A continuación se detallan estas variables de entrada.

TIPO DE ATAQUE

Son aquellas variables que definen el ataque que se esta produciendo y que aportan la información necesaria para su priorización y también para valorar el riesgo que supone.

Podríamos definir como ataques todas aquellas acciones que supongan una violación de la seguridad de nuestro sistema (confidencialidad, integridad o disponibilidad).

Dichas acciones se pueden clasificar de modo genérico según los efectos causados:

- **Interrupción:** Un recurso del sistema es destruido o se vuelve no disponible. Éste es un ataque contra la disponibilidad. Ejemplos de este ataque son los Nukes, que causan que los equipos queden fuera de servicio. También la destrucción o sabotaje de un elemento hardware, como cortar una línea de comunicación.
- **Intercepción:** Una entidad no autorizada consigue acceso a un recurso. Éste es un ataque contra la confidencialidad. Ejemplos de este ataque son la obtención de datos mediante el empleo de programas troyanos o la copia ilícita de archivos o programas (intercepción de datos), o bien la lectura de las cabeceras de

paquetes de datos para desvelar la identidad de uno o más de los usuarios mediante el Spoofing o engaño implicados en la comunicación intervenida (intercepción de identidad).

- **Modificación:** Una entidad no autorizada no sólo consigue acceder a un recurso, sino que es capaz de manipularlo. Virus y troyanos poseen esa capacidad. Éste es un ataque contra la integridad. Ejemplos de este ataque son la modificación de cualquier tipo en archivos de datos, alterar un programa para que funcione de forma distinta y modificar el contenido de información que esté siendo transferida por la red.
- **Fabricación:** Una entidad no autorizada inserta objetos falsificados en el sistema. Éste es un ataque contra la autenticidad. Ejemplos de este ataque son la inserción de mensajes falsos en una red o añadir datos a un archivo. Asimismo estos ataques se pueden clasificar en términos de ataques pasivos y ataques activos:

  -*Ataques activos:* Estos ataques implican algún tipo de modificación de los datos o la creación de falsos datos: Suplantación de identidad, Modificación de mensajes, Web Spoofing,...

  -*Ataques pasivos:* En los ataques pasivos el atacante no altera la comunicación, sino que únicamente la escucha o monitoriza, para obtener de esta manera la información que está siendo transmitida. Los ataques pasivos son muy difíciles de detectar, ya que no provocan ninguna alteración de los datos. Sin embargo, es posible evitar su éxito mediante el cifrado de la información y otros mecanismos.

Esta clasificación resulta demasiado general y ambigua para que resulte interesante desde el punto de vista de la efectividad del sistema. Existen ataques complejos que incluyen características de varios de estos tipos. Realizar una clasificación que integre de manera disjunta todos los tipos de ataques posibles se hace demasiado compleja, no es el objetivo de este trabajo hacerla ya que complicaría demasiado el problema y distraería la atención de sus objetivos importantes.

En consecuencia, hemos escogido los tipos de ataque más importantes, según su "modus operandi" y que más frecuentemente se dan en la actualidad. Así tenemos los siguientes tipos de ataques:

- Escaneo de Puertos
- Ataques de Autentificación
- Denial of Service (DoS). Negación de servicio
- Ataques de Modificación-Daño
- Explotación de Errores

Escaneo de Puertos

Consiste en buscar puertos abiertos, determinando los que puedan ser receptivos o de utilidad. Es como si llamamos a un número de teléfono y según la señal que oigamos (comunicando, llamada, avería, etc.) sabemos el estado de ese teléfono en ese preciso momento. Después llamamos a otro número y así continuamente.

El escaneo tradicional consiste en seleccionar un rango de IPs y hacer esas "llamadas" a unas direcciones IP consecutivamente, aunque también se puede hacer un escaneo a una IP concreta. Los firewall actuales detectan esa llamada a puertos consecutivos y por lo tanto reconocen el escaneo. Así que se cambia el método y se escanean las IPs y los puertos de cada una de ellas de forma no consecutiva. También se puede intentar cambiar el método de comunicación entre ambas máquinas.

Dos PCs que se ponen en comunicación establecen una relación de Cliente/Servidor. El Servidor "escucha" todo lo que llega hasta sus puertos, se identifica por medio de su IP y de un puerto determinado. El Cliente establece la conexión con el Servidor a través de dicho puerto, que debe estar disponible o abierto.

Antes de empezar a intercambiar datos se realiza una operación cuya finalidad es la de reconocerse mutuamente. A esta operación se la conoce como HandShake. Esta operación se realiza en el protocolo TCP.

Este "saludo" entre ambos se realiza en tres pasos (Three-Way Handshake):

1. El Cliente dice al Servidor que quiere comunicarse con él enviándole un segmento SYN (Synchronize Sequence Number).

2. El servidor (si está abierto y escuchando) al recibir este segmento SYN (activa su indicador SYN) y envía un acuse de recibo al cliente. Si el servidor está cerrado envía un indicador RST.

3. El cliente comprueba la respuesta mediante paquetes ACK (Acknowledment, reconocimiento) y el estado del servidor (si está disponible o no) y dependiendo de ello comienza el intercambio de datos o no.

Si se da el caso en que la llamada es respondida y se produce un posterior intercambio de datos, cuando se acabe la transferencia, se realiza otra operación de 3 pasos, pero con segmentos FIN en vez SYN.

Pasemos a enumerar los distintos tipos de escaneo:

-Escaneo de conexión TCPconnect (): Es el sistema más simple de escaneo de puertos TCP. Si el puerto escaneado está abierto y a la escucha, devolverá una respuesta de éxito; cualquier otra respuesta conlleva que el puerto no está abierto o que no se puede establecer conexión con él. No necesita de privilegios especiales y se realiza a gran velocidad. Este método es fácilmente detectable. Se verá un gran número de conexiones y mensajes de error con una máquina que se conecta y desconecta continuamente.

-Escaneo TCP reverse ident: El protocolo ident permite averiguar el nombre de usuario y el dueño de cualquier servicio corriendo dentro de una conexión TCP. Conocido también como reverse DNS.

-FTP bounce attack: El protocolo ftp permite lo que se llama conexión proxy ftp. Es decir, conectarse a un ftp desde un servidor proxy y al hacer esto establecer una conexión y enviar un archivo a cualquier parte de Internet. De esto se aprovechan algunos atacantes para realizar escaneos, ya que se realizan detrás de un firewall (el del proxy) con la consiguiente dificultad para rastrear el origen del escaneo. Suelen ser muy lentos, por lo que son poco usados.

-Escaneo UDP ICMP port unreachable: En esta técnica no se usa el protocolo TCP, sino el UDP. Es un protocolo más simple que el TCP, lo cual tiene sus desventajas a la hora de escanear, ya que al llamar a un puerto, se encuentre éste abierto o no, no tiene por qué devolver una respuesta o un paquete de error. Pero el servidor del sistema

escaneado suele devolver un paquete de error “ICMP_PORT_UNREACH” cuando un puerto UDP esta cerrado.

-Fingerprinting: Consiste en determinar qué sistema operativo tiene el ordenador atacado. Lo normal es que se prueben varias técnicas y, según reaccione el ordenador de la víctima, se determina su sistema operativo. Hay programas específicos para esta labor. Cabe decir que no es un sistema completamente efectivo. El fingerprinting no es un escaneo en sí, pero estos son utilizados para poder llevarlo a cabo.

-Escaneo TCP SYN : Se envía un paquete SYN (como si se fuera a solicitar una conexión) y se espera por la respuesta. Al recibir un SYN/ACK se envía inmediatamente un RST (reset) para terminar la conexión y se registra este puerto como abierto. La principal ventaja de esta técnica de escaneo es que pocos sitios están preparados para registrarlos. La desventaja es que en algunos sistemas Unix, se necesitan privilegios de Administrador para construir estos paquetes SYN.

-Escaneo TCP FIN - Stealth Port Scanning : Escaneo invisible de puertos. Hay veces en que incluso el escaneo SYN no es lo suficientemente discreto. Algunos sistemas (Firewalls y filtros de paquetes) monitorizan la red en busca de paquetes SYN a puertos restringidos. En cambio los paquetes FIN podrían ser capaces de pasar inadvertidos.

Siguiendo las pautas y reglas del protocolo TCP, cuando un cliente envía un paquete FIN a un servidor, éste último reacciona de 2 maneras dependiendo del estado del puerto por el que se intenta la comunicación:

-Si está cerrado, devuelve un paquete RST

-Si está abierto, lo ignora.

Se da la curiosa situación por la que, debido a una corrección en la gestión del protocolo TCP en los sistemas Microsoft, independientemente del estado del puerto, siempre se devuelve un paquete RST. Esta corrección hace que los sistemas Microsoft sean invulnerables a este tipo de escaneo. Sin embargo, es posible realizarlo en sistemas Unix.

-Escaneo de fragmentación: En lugar de enviar paquetes completos de sondeo, se parten en un par de pequeños fragmentos IP. Así es más difícil de monitorizar por los filtros que pudieran estar ejecutándose en el sistema atacado. Esta técnica puede producir caídas de rendimiento tanto en el sistema del cliente como en el del servidor, por lo que lo hacen detectable.

-Eavesdropping-Packet Sniffing: Olfateo de paquetes sin modificarlos. Muchas redes son vulnerables al Eavesdropping o la interceptación pasiva (sin modificación) del tráfico de red. Esto se realiza con paquetes Sniffer, que monitorizan la información que circula por la red, que se centran en las IPs, ya que siempre que se produce una comunicación por la red, en esos paquetes de información se incluyen las IPs de los 2 sistemas que se están comunicando.

-Snooping-Downloading: Se parece al sniffing en el sentido de que obtienen información sin modificarla, pero el sistema empleado es distinto ya que con este sistema lo que se hace es copiar la información y bajarla al PC en forma de archivos.

Autentificación

Consisten, como su nombre indica, en la suplantación de una persona con autorización por parte del atacante. Se suele realizar de dos formas: obteniendo el

nombre y contraseña del atacado o suplantando a la víctima una vez ésta ya ha iniciado una sesión en su sistema.

Para realizar ataques de este tipo se utilizan varias técnicas, las cuales pasamos a describir a continuación.

- ***Simulación de identidad:*** Es una técnica para hacerse con el nombre y contraseña de usuarios autorizados de un sistema. El atacante instala un programa que recrea la pantalla de entrada al sistema, cuando el usuario intenta entrar en él teclea su login y password, el programa los captura y muestra una pantalla de "error en el acceso" al usuario. El usuario vuelve a teclear su login y password, entrando esta vez sin problemas. El usuario cree que en el primer intento se equivocó al teclear, sin embargo, su login y password han sido capturados por el atacante.

- ***Spoofing (Engaño):*** Este tipo de ataques suele implicar un buen conocimiento del protocolo en el que se va a basar el ataque. Consiste en sustituir la fuente de origen de una serie de datos (por ejemplo, un usuario) adoptando una identidad falsa para engañar a un firewall o filtro de red. Los ataques Spoofing más conocidos son el IP Spoofing, el DNS Spoofing , el Web Spoofing y el fake-mail.

  -*IP Spoofing:* Sustituir una IP. El atacante logra identificarse con una IP que no es la suya, con lo que a ojos del atacado, el agresor es una tercera persona, que nada tiene que ver en el asunto, en vez de ser el atacante real.

  -*DNS Spoofing:* Sustituir a un servidor DNS o dominio. Se usan paquetes UDP y afecta a sistemas bajo Windows NT. Se aprovecha de la capacidad de un servidor DNS resolver una petición de dirección IP a partir de un nombre que no figura en su base de datos, ya que éste es su método de trabajo por defecto.

  -*Web Spoofing:* El atacante crea un sitio web (falso) similar al que la víctima desea entrar. Los accesos a este sitio están dirigidos por el atacante, permitiéndole monitorizar todas las acciones de la víctima: datos, contraseñas, números de tarjeta de créditos, etc. El atacante también es capaz de modificar cualquier dato que se esté transmitiendo entre el servidor original y la víctima o viceversa. Es un ataque peligroso, y difícilmente detectable, que hoy por hoy se puede llevar a cabo en Internet Hoy en día tanto JavaScript como Active-X, como Java tienden a facilitar las técnicas de spoofing.

  -*Fake-mail:* Es otra forma de spoofing y consiste en el envío de e-mails con remitente falso. Aquí el atacante envía E-Mails en nombre de otra persona con cualquier motivo y objetivo.

- ***Looping:*** El intruso usualmente utiliza algún sistema para obtener información e ingresar en otro, que luego utiliza para entrar en otro, y así sucesivamente. Este proceso se llama looping y tiene como finalidad hacer imposible localizar la identificación y la ubicación del atacante, de perderse por la red. Entre el origen físico y el sistema que finalmente se utilice para realizar una fechoría puede estar plagado de muchos sistemas intermedios, rebasando las fronteras de varios países, dificultando aún más su localización. Otra consecuencia del Looping es que la víctima puede suponer que están siendo atacada por nosotros (si somos el sistema final del looping del atacante), cuando en realidad está siendo atacada por un Insider, o por alguien que se encuentra a miles de kilómetros tanto de nosotros como de la víctima.

La localización de la procedencia de un looping es casi imposible de determinar, ya que el investigador debe contar con la colaboración de cada Administrador de cada red utilizada en la ruta del looping, incluso de gobiernos de otros países.

-Ip Splicing-Hijacking: Es un método de sustitución que consiste en que el atacante espera a que la víctima entre en una red usando su nombre, contraseña y demás y una vez que la víctima ha superado los controles de identificación y ha sido autorizada la "tira" del sistema y se hace pasar por ella.

-Utilización de Backdoors (puertas traseras): Las puertas traseras son trozos de código en un programa que permiten a quien los conocen saltarse los métodos usuales de autentificación para realizar ciertas tareas. Habitualmente son insertados por los programadores del sistema para agilizar la tarea de probar código durante la fase de desarrollo. No es por tanto un método de suplantación, si no de saltarse los controles de autentificación o, como su nombre indica, entrar por la "puerta de atrás".

-Exploits: Es muy frecuente ingresar a un sistema aprovechándose de agujeros (bugs, holes) en los algoritmos de encriptación utilizados, en la administración de las claves por parte la empresa, o simplemente encontrado un error en los programas utilizados. Los programas para aprovechar o explotar estos "agujeros", tanto en el software como en el hardware, se denominan "exploits". Cada día aparecen varios agujeros y se cuentan por miles en los sistemas.

-Obtención de contraseñas: Es la obtención por "Fuerza Bruta" de nombres de usuarios y claves de acceso. Casi todas las contraseñas que utilizamos habitualmente están vinculadas a nuestros nombres reales, nombres de familiares y/o mascotas, fechas significativas,,... etc. Además, no las solemos cambiar periódicamente. También se suele realizar este tipo de ataques usando una clase de programas llamados diccionarios.

-Diccionarios: Los Diccionarios son programas que en su base de datos contienen millones de palabras. Van probando con millones de combinaciones de letras y números encriptados, incluso con caracteres especiales hasta descubrir la combinación correcta de nombre y usuario de la víctima. Son pues programas de fuerza bruta.

Negación de servicio. DENIAL of SERVICE (DoS)

Se produce la imposibilidad por parte de la víctima de acceder y/o permitir el acceso a un recurso determinado. Por ejemplo, imposibilidad de conectarse, de usar el correo electrónico o, a un mayor nivel, imposibilidad de un servidor de prestar sus servicios. Se basa en el hecho comprobado de que es más fácil corromper un sistema que acceder clandestinamente al mismo. Estos ataques intentan corromper o saturar los recursos de la víctima por medio de peticiones de conexión para lograr desactivarla o impedir el acceso a otros usuarios por medio de la saturación.

Veamos los ataques de este tipo más usuales:

-Jamming o Flooding: Son ataques que saturan los recursos del sistema de la víctima dejándola sin memoria, sin espacio libre en su disco duro o saturando sus recursos de red. Por ejemplo, el atacante satura el sistema con peticiones de conexión. Sin embargo, en vez de enviar la IP real del emisor, envía una falsa. Al no encontrar el sistema una respuesta desde esa IP falsa, mantiene ese buffer abierto esperando información pero bloqueando la comunicación con la IP verdadera. Los ataques más frecuentes a proveedores son usando el ping de la muerte (que bloquea el equipo) o enviando miles de correos electrónicos los usuarios de ese servidor, de forma continuada, saturando los sistemas. Relacionados con el tema están los rabbits (conejos), que son programas que

provocan procesos inútiles y se reproducen como conejos hasta que satura la capacidad del sistema, provocando su colapso.

-SYN Flood (Inundación con paquetes SYN): Como ya se explicó en el escaneo TCP SYN el protocolo TCP se inicia con una conexión en tres pasos. Si el paso final no llega a establecerse, la conexión permanece en un estado denominado "semiabierto". El Syn Flood es el más famoso de los ataques tipo Denial of Service (DoS). Se basa en un "saludo" incompleto entre los dos sistemas. El Cliente envía un paquete SYN pero no responde al paquete ACK del 2º paso del saludo ocasionando que el servidor permanezca a la escucha un determinado tiempo hasta cancelar la llamada. Si se envían muchos saludos incompletos, se consigue que el servidor se paralice o por lo menos se ralentice. Para operar con este sistema hay que mantener el Sys Flood activo, ya que la mayoría de los sistemas tienen un límite de espera muy corto para conexiones semiabiertas. Cuando se ha esperado mucho tiempo, se libera un hueco para aceptar otras posibles peticiones, lo cual puede ser aprovechado para “colar” un paquete SYN destructivo o malicioso. Así que, este ataque se puede utilizar tanto para consumir los recursos de un sistema como para abrir el camino a otro tipo de ataque. Si este ataque es potente es porque el atacante no necesita apenas potencia en su PC, con mandar un SYN cada 4 segundos es suficiente. Esta velocidad se consigue de sobra con cualquier modem. Este ataque suele combinarse también con el IP Spoofing (suplantación de una IP), para ocultar el origen del ataque.

> *-Connection Flood (inundación de la conexión):* Se basa en la característica de la mayoría de los proveedores de Internet (ISP) de tener un tope máximo de conexiones simultaneas, que tras ser alcanzado no acepta más conexiones. Si por ejemplo un servidor Web tiene un tope de 1000 conexiones, y el atacante establece mil conexiones y no realiza ninguna petición sobre ellas, monopolizará la capacidad del servidor. Las conexiones van caducando por inactividad poco a poco, pero el atacante sólo necesita establecer nuevas conexiones para mantener fuera de servicio el servidor.
>
> *-Net Flood (inundación de la red):* En este ataque se envían tantas solicitudes de conexión que las conexiones de los demás usuarios no pueden llevarse a cabo. Es un ataque muy dañino y con poca defensa por parte de la red atacada. Para solucionarlo el Proveedor tiene que detectar el origen del ataque, bloquear la comunicación desde esa dirección y avisar al administrador de la misma para que actúe, ya que lo normal es que el administrador de esa dirección no sepa nada y que esté siendo utilizada su red para llevar a cabo el ataque por medio de algún spoofing. De cualquier manera, saber el origen real del ataque es prácticamente imposible. Una vez se ha solucionado el problema, el servidor puede haber estado colgado durante horas.

-Land Attack: Este ataque consiste en un Bug en la implementación de la pila TCP/IP en sistemas Windows. El ataque envía a algún puerto abierto de un servidor (generalmente al 113 o al 139) un paquete, maliciosamente construido con la IP y puerto origen igual que la IP y puerto destino. Al final la máquina termina por colapsarse.

Existen ciertas variantes a este método consistente, por ejemplo, en enviar el mensaje a una dirección específica sin especificar el puerto Smurf o Broadcast Storm. Este ataque es bastante simple y a su vez devastador. Consiste en recolectar una serie de direcciones para a continuación mandar una petición ICMP (simulando un Ping) a cada una de ellas en serie, varias veces, falsificando la dirección IP de origen. Este paquete

maliciosamente manipulado, será repetido en Broadcast, y cientos ó miles de hosts (según la lista de direcciones de Broadcast disponible) mandarán una respuesta a la víctima cuya dirección IP figura en el paquete ICMP.

-Supernuke o Winnuke: El Nuke es el ataque más común de los equipos Windows, que hace que los equipos que escuchan por el puerto UDP 137 a 139 (utilizados por los protocolos NetBios), queden fuera de servicio o disminuyan su rendimiento al enviarle paquetes o fragmentos UDP manipulados. Generalmente se envían fragmentos de paquetes, que la máquina víctima detecta como erróneos pasando a un estado inestable.

> -*ICMP Nuke:* Éste es un nuke que se basa en el protocolo de control ICMP (Internet Control Message Protocol) que es incorporado al protocolo IP de Internet. Este protocolo se encarga de avisar cuando hay un fallo en el sistema de envío de paquetes de un protocolo TCP/IP. Si hay algún fallo (por ejemplo: No route to host), este protocolo se encarga de avisar al TCP/IP, y se corta el envío.
>
> -*OOB Nuke:* Este ataque, sin el debido parche por parte de la victima, provoca una caída del sistema bastante violenta. El OOB Nuke (Out Of Band) consiste en mandar cierto paquete de información al puerto 139 de la maquina victima. Se trata de un bug en los protocolos de comunicación del windows 3.x, 9x y NT, provocando que cuando recibe un paquete con el flag OOB al puerto 139, se produce un fallo de protección general en el sistema.

-Teardrop: Al igual que el Supernuke, los ataques Teardrop I y Teardrop II afectan a fragmentos de paquetes. Algunas implementaciones de colas IP no vuelven a recomponer correctamente los fragmentos ya que se superponen, haciendo que el sistema se cuelgue. El Windows NT 4.0 es especialmente vulnerable a este ataque. Aunque existen parches que pueden aplicarse para solucionar el problema. Los ataques tipo Teardrop son especialmente peligrosos ya que existen multitud de implementaciones (algunas de ellas forman paquetes), que explotan esta debilidad. Las más conocidas son aquellas con el nombre Newtear, Bonk y Boink.

-Mail Bombing-Mail Spamming-Junk Mail: El Mail Bombing consiste en un envío indiscriminado y masivo de un mensaje idéntico a una misma dirección, saturando así buzón de correo (mailbox) del destinatario.

El Mail Spamming en cambio es un bombardeo publicitario que consiste en enviar un e-mail a miles de usuarios, hayan estos solicitado el mensaje o no. Es muy utilizado por las empresas para publicitar sus productos. El Spamming esta siendo actualmente tratado por las leyes europeas como una violación de los derechos de privacidad del usuario. El junk mail o correo basura es una propaganda indiscriminada y masiva a través de e-mails.

El mail spamming y el junk mail no pueden ser considerados como ataques DoS auténticos, porque por mucho que molesten no están encaminados a saturar ningún sistema, aunque en ocasiones se utilicen para que éste se produzca.

Modificación-Daño

-Tampering o Data Diddling: Consiste en una modificación no autorizada de datos o software instalado en el sistema víctima, incluyendo borrado de archivos. Es decir, una vez se ha entrado en un sistema, se modifican o borran datos del mismo.

-Borrado de huellas: Es una tarea fundamental que realiza un intruso tras haber hurgado en un sistema y terminado su fechoría. Si no lo hace se puede descubrir tanto el origen del ataque como el agujero de seguridad por el que ha entrado el atacante, que

será subsanado inmediatamente. Las Huellas son todas las tareas que realizó el intruso en el sistema y por lo general son almacenadas en los Logs de los sistemas operativos, de monitores de red y/o sistema, firewalls,...

-Ataques usando Java Applets: Los java Applets son ejecutados desde otra aplicación, como un navegador de Internet. Su mayor popularidad la merece en su alto grado de seguridad. Los más usados navegadores actuales, implementan Máquinas Virtuales Java o JVM (Java Virtual Machine) para poder ejecutar programas (Applets) de Java. Los Applets no son más que un código ejecutable y como tal, puede ser manipulado por intrusos. Sin embargo las restricciones a las que están sometidos los Applets son muchas (imposibilidad de trabajar con ficheros a no ser que el usuario especifique lo contrario, imposibilidad de acceso a zonas de memoria y disco directamente, firma digital, etc.) por lo que es muy difícil atacar por medio de Java Applets. Sin embargo, es posible si se es un experto en Java. Aunque se han detectado algunos agujeros de seguridad, han sido corregidos.

-Ataques usando JAVAScript y VBScript: JavaScript (de empresa Netscape) y VBScript (de Microsoft) son dos lenguajes usados por los diseñadores de sitios Web evitando el uso de Java. Los programas realizados son interpretados por el navegador. A diferencia del applets de JAVA, los JavaScripts y los VBSscripts son capaces de iniciar aplicaciones tales como grabar información en el disco, por lo que atentan seriamente contra la seguridad de un sistema ya que pueden colar, por ejemplo, algún tipo de virus. Aunque los fallos son mucho más numerosos en versiones antiguas de JavaScript.

-Ataques usando Actives: La seguridad ActiveX se basa en certificados y firmas digitales. Una Autoridad Certificadora (AC) expende un certificado que acompaña a los controles activos y a una firma digital del programador. Cuando un usuario descarga una página con un control, se le preguntará si confía en la AC que expendió el certificado y/o en el control ActiveX. Si el usuario acepta el control, éste puede pasar a ejecutarse sin ningún tipo de restricciones (sólo las propias que tenga el usuario en el sistema operativo). Es decir, la responsabilidad de la seguridad del sistema se deja en manos del usuario. Este es el punto débil de la seguridad en el ActiveX ya que la práctica mayoría de los usuarios aceptan todos los certificados sin pararse a leerlos. Aunque leamos el certificado de seguridad y nos parezca correcto, eso no nos libra de sufrir un ataque. Los certificados se basan en que el programador del control ActiveX asegura que su control no es nocivo, lo cual es falso en bastantes casos.

-Cookies: Aunque son motivo de controversia, las cookies o galletas solo representan una amenaza para la privacidad ya que pueden cumplir la función de espiar. No son capaces de causar daño al sistema. Básicamente se utilizan para reconocer al usuario al entrar en determinados sitios web (foros, por ejemplo) y para contadores de visitas. De todas formas, los navegadores tienen opciones para configurar el ingreso de cookies en nuestro sistema.

-Ataques usando Vulnerabilidades de los Navegadores: Son fallos del navegador en sí y no de programas implementados, como pueden ser los controles de JAVA, ActiveX, etc. Los más comunes son los "Buffer Overflow" (desbordamiento de buffer). Debido a una debilidad de los buffers de algunos navegadores para el procesamiento de las entradas de usuario, se puede llegar a manipular datos. Una manipulación común es la de URLs almacenadas. Por ejemplo, se puede modificar una URL para que cuando sea ejecutada por el usuario, se ponga en funcionamiento otra aplicación, como el format, un virus,...

Explotación de Errores

Muchos sistemas están expuestos a "agujeros" de seguridad que son explotados para acceder a archivos, obtener privilegios o realizar sabotaje. Estas vulnerabilidades ocurren por variadas razones, y miles de "puertas traseras" y bugs son descubiertas diariamente en sistemas operativos, aplicaciones, protocolos de red, navegadores de Internet, servidores de correo electrónico,...

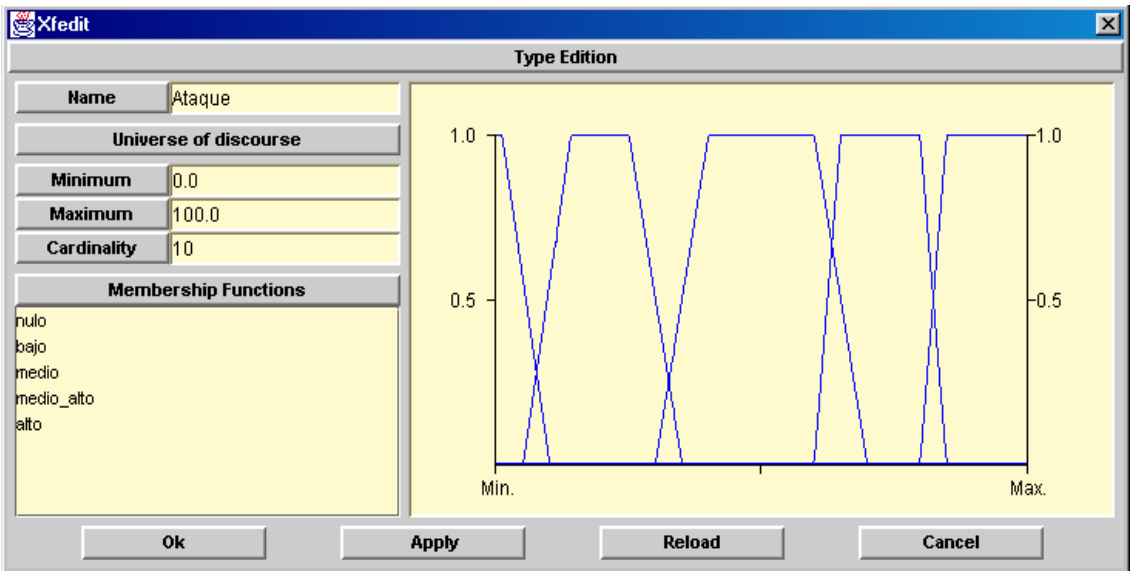
Los Sistemas operativos abiertos (como Unix y Linux) tienen agujeros mas conocidos y controlados que aquellos que existen en sistemas operativos cerrados (como Windows). La importancia y ventaja del código abierto de sistemas operativos como Linux radica en que miles de usuarios analizan dicho código en busca de posibles bugs y ayudan a obtener soluciones en forma inmediata. En sistemas cerrados como Windows suele ocurrir lo contrario. Se ocultan los errores para impedir una caída de ventas del producto debido a una mala reputación. No obstante existen varias organizaciones que se ocupan de investigar esos sistemas en busca de agujeros de seguridad.

Constantemente encontramos en Internet avisos de nuevos descubrimientos de problemas de seguridad (y herramientas de Hacking que los explotan), por lo que hoy también se hace indispensable contar con productos que conozcan esas debilidades, que puedan diagnosticarlas y que sean capaces de actualizar el programa afectado con el parche adecuado.

➢ Type Ataque

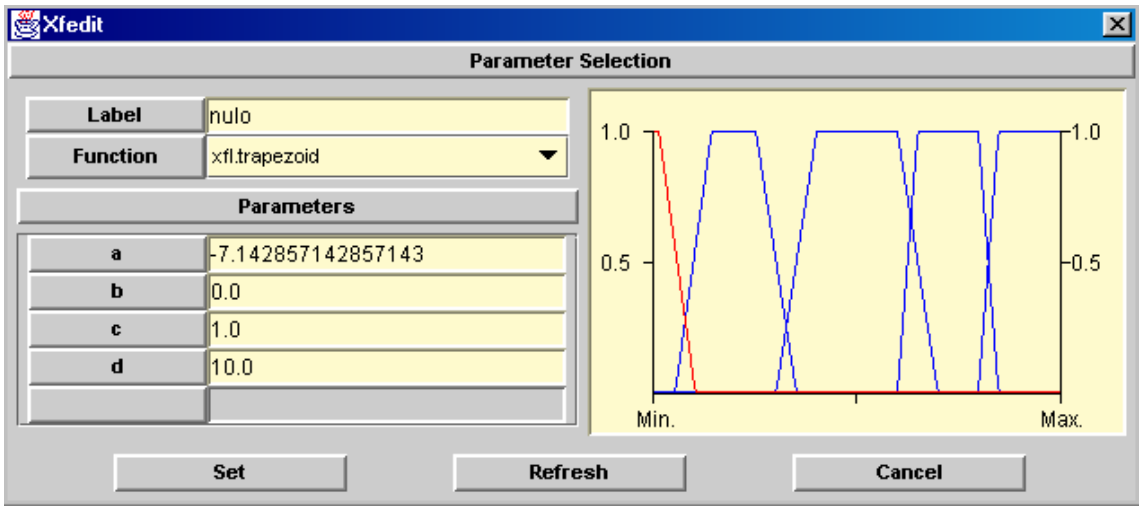
En este trabajo no se ha entrado en los detalles más específicos de cada tipo de ataque, ya que esto supera las limitaciones que este sistema borroso tiene dentro del marco en que ha sido elaborado. Aunque en las posibles ampliaciones que pudiera tener el sistema, este punto debería ser considerado con más detalle. También cabe decir que la determinación y valoración de cada ataque no procede al sistema que ahora nos ocupa, por tanto no se ha resuelto en él.

En este sistema se considera a un ataque como perteneciente a alguno de los tipos anteriores, pudiendo ser un ataque múltiple (varios ataque relacionados incluidos en un evento). El tipo Ataque tipifica exclusivamente una valoración de la peligrosidad inherente a cada tipo de ataque. Se ilustra a como sigue.

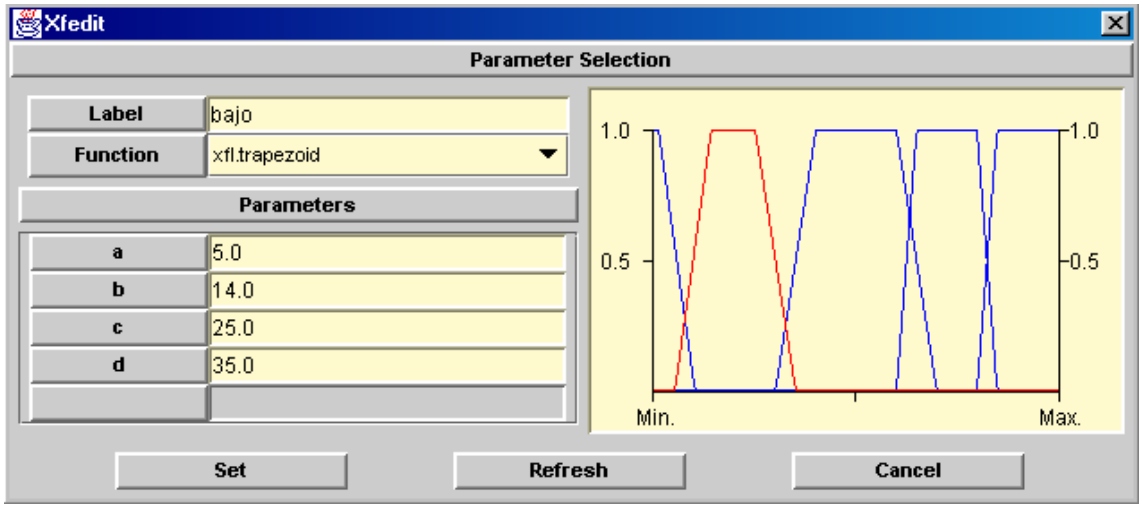


A partir del cual se definen las variables *escaneo_puertos*, *autentificación*, *negación_servicio*, *modificacion_dano* y *explotacion_errores*. Todos los ataques se basan en el mismo tipo. A continuación especificamos las etiquetas de este tipo. Se consideran *nulo*, *bajo*, *medio*, *medio_alto* y *alto*.

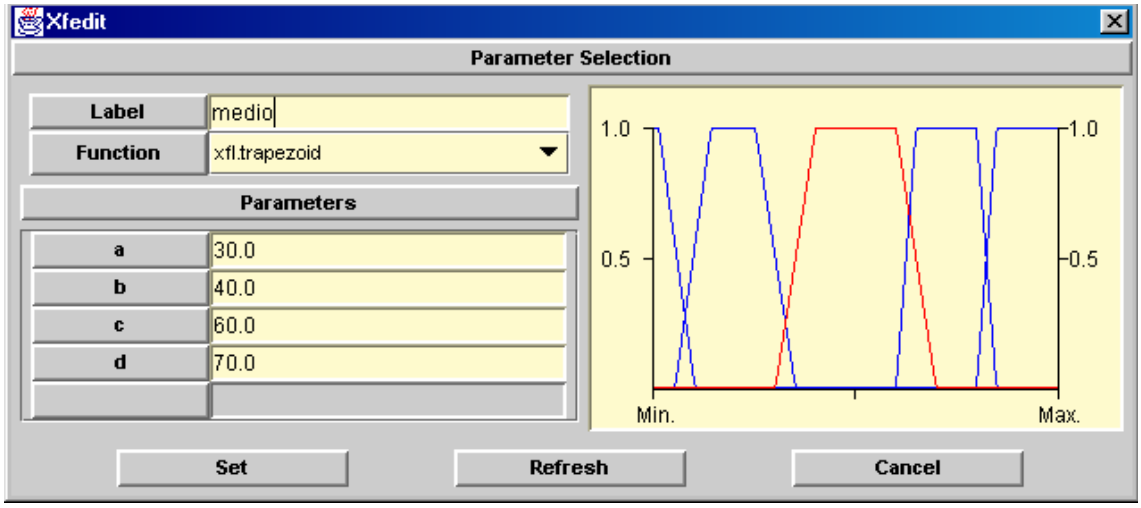
- *Nulo:*



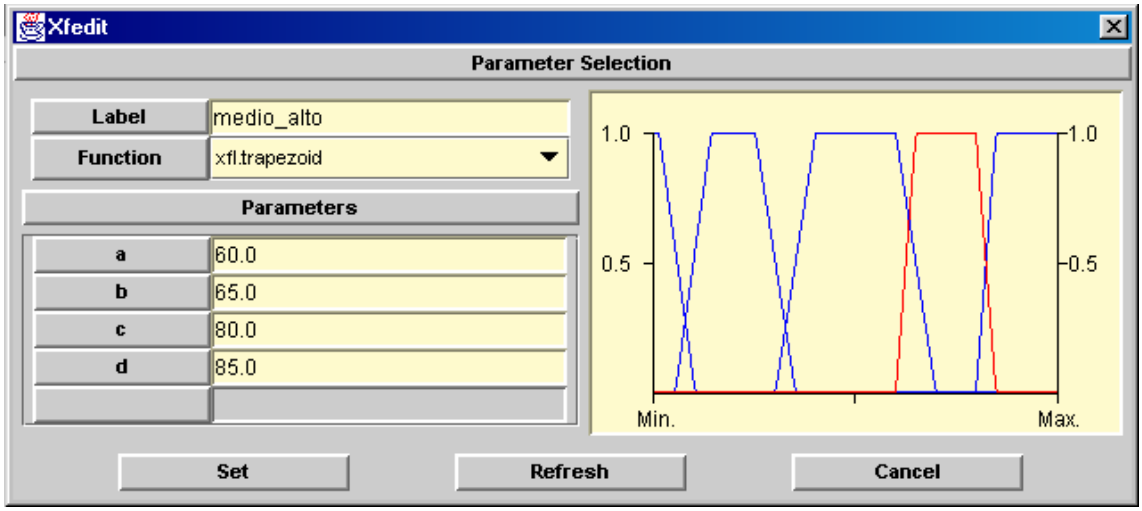
- *Bajo:*



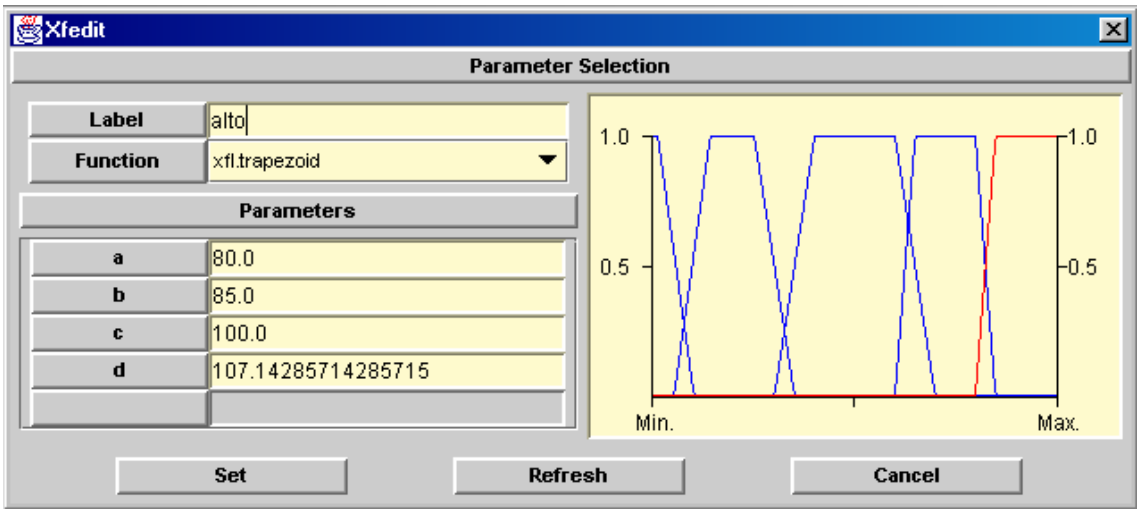
- *Medio:*



- *Medio_alto:*



- *Alto:*


Xfedit
Parameter Selection
Label
alto
Function
xfl.trapezoid
Parameters
a
80.0
b
85.0
c
100.0
d
107.14285714285715
1.0
0.5
1.0
0.5
Min.
Max.
Set
Refresh
Cancel

INVENTARIO DE SISTEMAS

Obviamente, es necesario tener una relación de los datos actualizados de todas las máquinas y redes pertenecientes a la organización. Aunque para este trabajo solo interesan los datos del sistema concreto que ha sido atacado. En la implementación del OSSIM se incluye diversas bases de datos con los datos referentes al inventario de sistemas. Los inputs referentes a los sistemas atacados son:

- Estado de los **Puertos**
- Disponibilidad de **Firewall**
- **Sistema Operativo** usado

Puertos

El protocolo de transmisión de información por defecto utilizado en Internet es TCP/IP (Transmision Control Protocol / Internet Protocol). Se basa en usa estructura Cliente/Servidor. Es decir, para que un sistema funcione debe de haber al menos un servidor al que se conecten los clientes. Internet también se puede basar en conexiones Punto a Punto en las que se utilizan direcciones de broadcast para comunicarse con los ordenadores.

Cada ordenador conectado por TCP/IP tiene una estructura lógica de cara a la red. La conexión a la red esta dividida en Puertos. Un puerto es una zona en la que dos ordenadores intercambian información. Una configuración típica de TCP/IP habilita 65536 puertos (0...65535) para las conexiones de red.

Cada puerto tiene cinco estados posibles: Cerrado, Rechazado, Abierto, Conectado o Escucha:

- Un puerto *Cerrado* es aquel en el que no se está realizando ninguna actividad ni está precedida ninguna actividad.
- Un puerto *Abierto* es aquel que está esperando una orden del programa.
- Un puerto *Conectado* es un puerto abierto que ha recibido la orden de conectarse a un puerto de otro ordenador y cuya petición de conexión ha sido aceptada.
- Un puerto *Rechazado* es un puerto abierto que ha recibido la orden de conectarse a un puerto de otro ordenador pero cuya petición ha sido denegada.
- Un puerto en estado de *Escucha* es un puerto que está esperando la conexión de otro ordenador.

Firewall

Un Firewall es un sistema o grupo de sistemas que impone una política de seguridad entre la organización de red privada e Internet. El firewall determina cual de los servicios de red pueden ser accedidos dentro de esta por los que están fuera, es decir quien puede entrar para utilizar los recursos de red pertenecientes a la organización. Para que un firewall sea efectivo, todo trafico de información a través de Internet deberá pasar a través del mismo donde podrá ser inspeccionada la información. El firewall podrá únicamente autorizar el paso del trafico, y el mismo podrá ser inmune a la penetración. desafortunadamente, este sistema no puede ofrecer protección alguna una vez que el agresor lo traspasa o permanece entorno a este.

El firewall permite al administrador de la red definir un “choke point” (embudo), manteniendo al margen los usuarios no-autorizados fuera de la red, prohibiendo potencialmente la entrada o salida al vulnerar los servicios de la red, y proporcionar la protección para varios tipos de ataques posibles.

Sistema Operativo

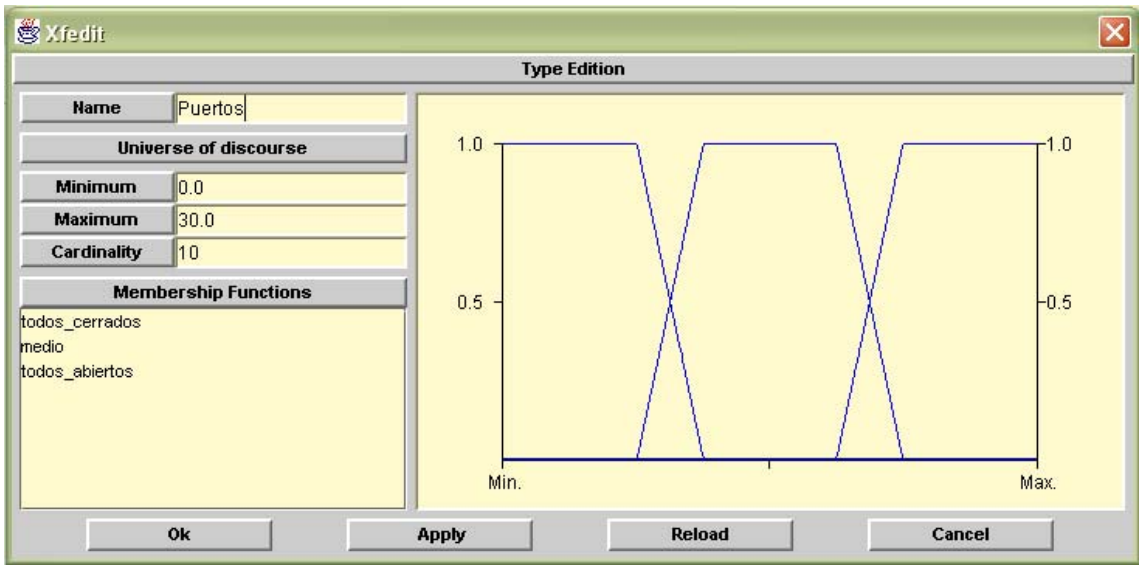
La influencia del Sistema Operativo en el éxito o fracaso de un ataque es determinante, aunque, por supuesto, no solo depende de este. Cada tipo de ataque intenta explotar alguna de las debilidades o fallos de seguridad del mismo. Sin embargo, hay SO considerados como más seguros para un tipo de ataque determinado.

En general, consideraremos los sistemas Windows como menos seguros para la mayoría de ataques. El código fuente de Windows no está a disposición de los usuarios, por lo tanto, no puede ser probado y testado por estos. Al contrario, los sistemas Linux, de código libre, han sido depurados por toda la comunidad Linux lo que facilita la detección y corrección de errores de seguridad. Algo parecido pasa con Unix, en el cual se basa Linux.

➢ Type Puertos

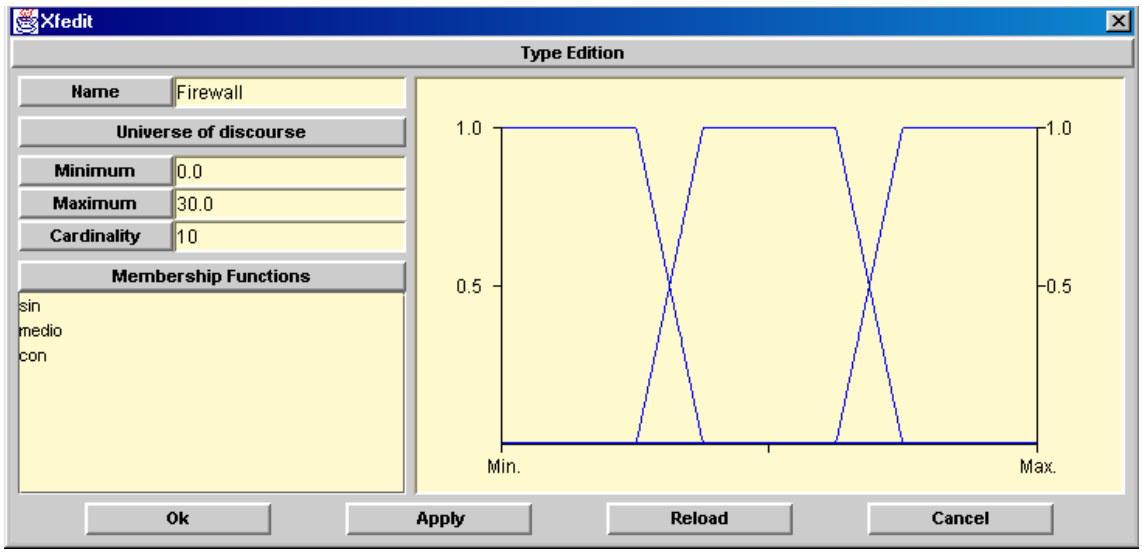
Distinguimos entre *todos_cerrados* (nivel de seguridad alto), *medio* (nivel de seguridad estándar) y *todos_abiertos* (nivel de seguridad bajo). La etiqueta todos_cerrados no significa, necesariamente, que todos los puertos lo estén sino que el número de puertos sensibles a un ataque que están cerrados es mayoritario. La etiqueta medio indica que la cantidad de puertos abiertos está en una proporción similar a la de cerrados. La etiqueta todos_abiertos alude a que la mayoria de los puertos sensibles están abiertos con el consiguiente riesgo de seguridad.

Aquí tenemos la definición del tipo *puertos*



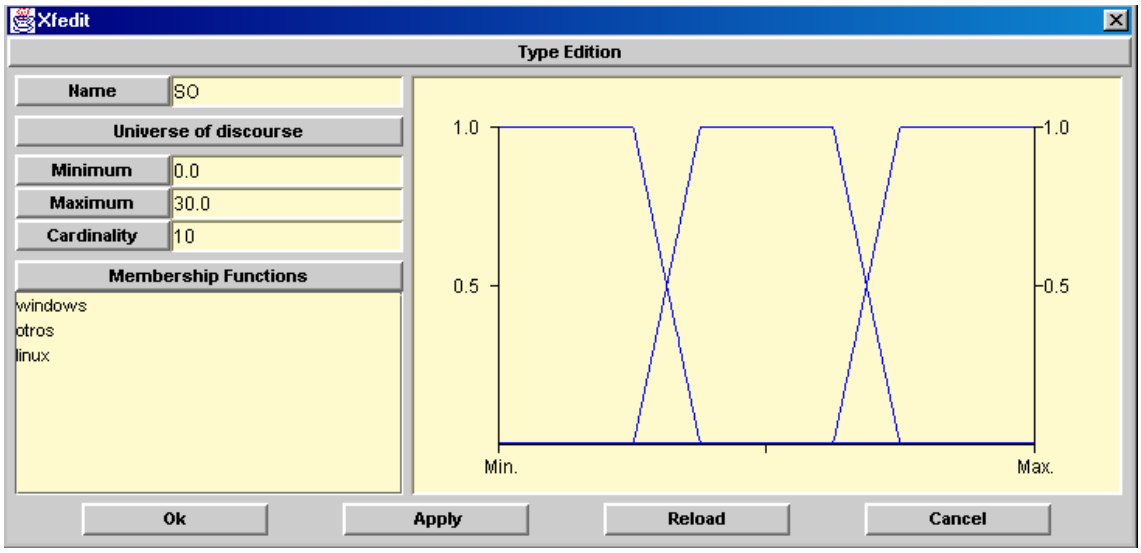
➢ Type Firewall

En lo que respecta al control del tráfico, distinguimos entre *sin*, que implica la ausencia de capacidades Firewall, *medio*, que dispone de capacidades medias de Firewall y *con*, que supone capacidades altas y avanzadas de Firewall.



➢ Tipo SO

En este tipo hay tres posibles etiquetas: *windows*, *otros* y *linux*. El primero tiene el menor nivel de seguridad (mayor cantidad de agujeros y fallos de seguridad), el segundo agruparía a todos los sistemas Unix y tiene implícita una seguridad mayor que el anterior. El último de todos sería el que mayores capacidades de seguridad tiene.



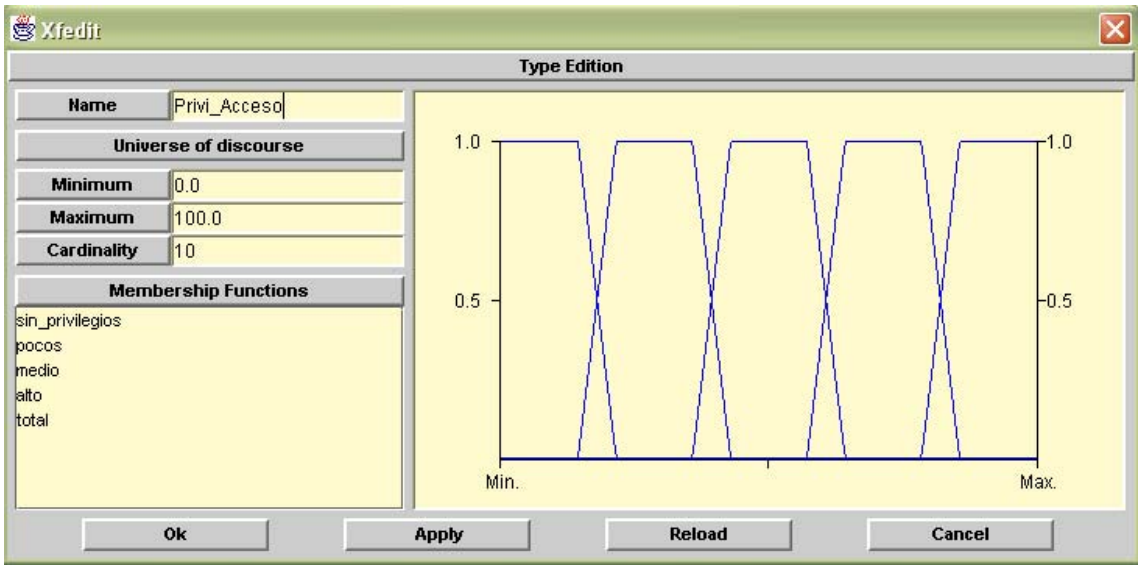
POLÍTICAS DE ACCESO

La política de accesos se refiere a desde donde a donde esta permitido o prohibido. En este sistema serían los privilegios de acceso de que dispone la ubicación desde donde proviene el posible ataque.

Recordemos aquí el principio de mínimos privilegios, todos los programas y usuarios del sistema deben operar utilizando el menor conjunto de privilegios necesarios para completar la labor. Los derechos de acceso deben adquirirse sólo por permiso explícito; por omisión deberían ser "sin acceso". Lamentablemente esto no ocurre así en la mayoría de ocasiones, a raíz de lo cual ocurren numerosos desastres.

➢ Type Privilegios (Priv_Acceso)

Las etiquetas son *sin_privilegios*, *pocos*, *medio*, *alto* y *total*. Se refieren a la cantidad de privilegios de que dispone el origen del ataque. La morfología de las etiquetas es trapezoidal, ya que esta se adapta bien al universo del discurso

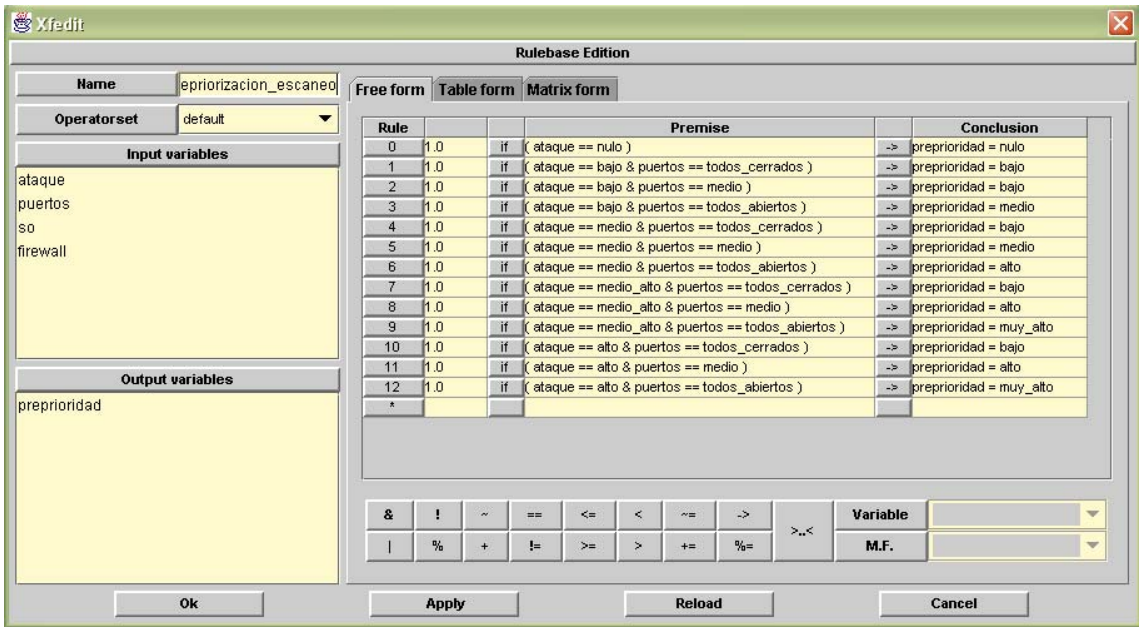


VARIABLES INTERMEDIAS Y FUNCIONAMIENTO DEL SUBSISTEMA

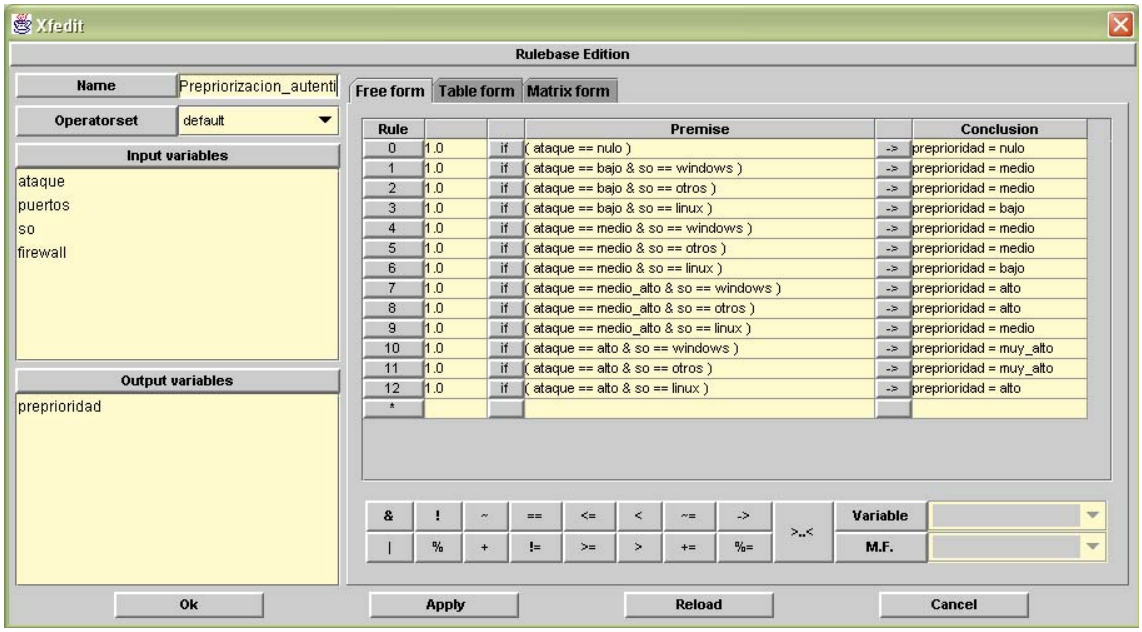
Se ha diseñado una base de reglas para cada tipo de ataque. Cada una de estas bases recibirá como entradas, además del tipo de ataque, los datos del sistema atacado (puertos, firewall y so). Estos módulos de preprioritización son los encargados de priorizar cada tipo de ataque independientemente de que halla otros ataques en el evento.

Aquí se ilustran las Bases de Reglas de Preprioritización.

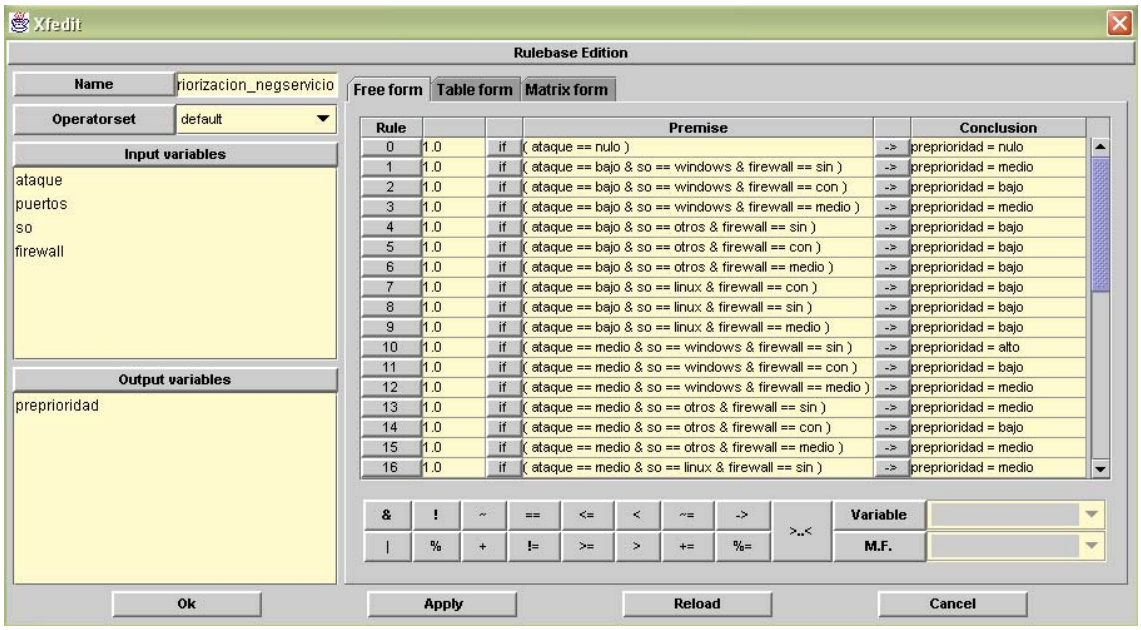
- *Preprioritización-Escaneo de Puertos:*



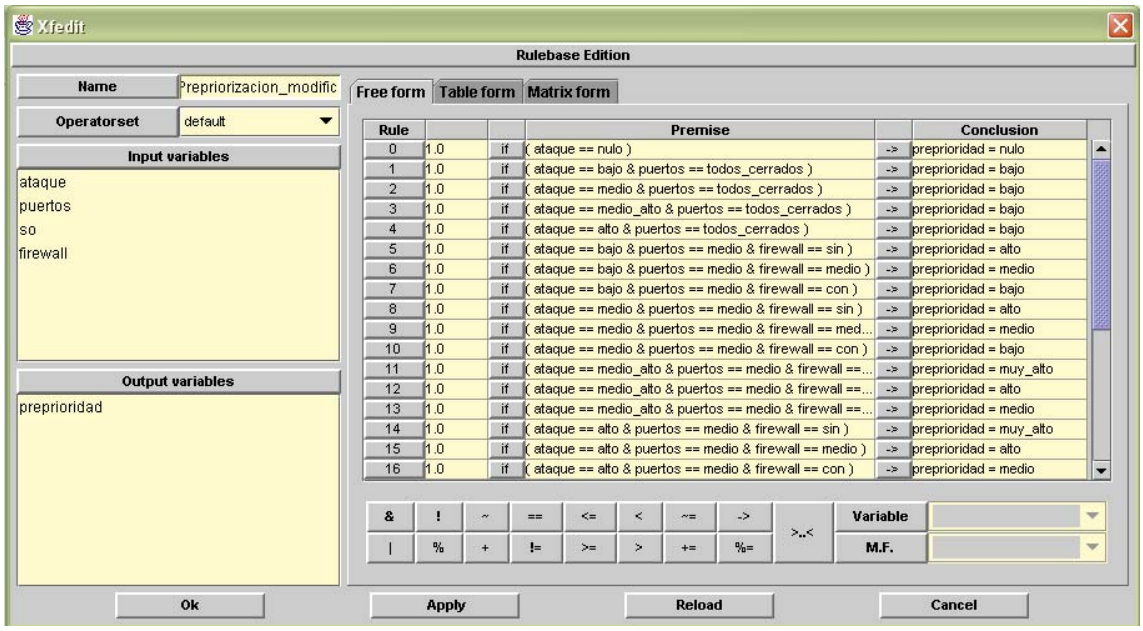
- *Preprioritización- Autentificación:*



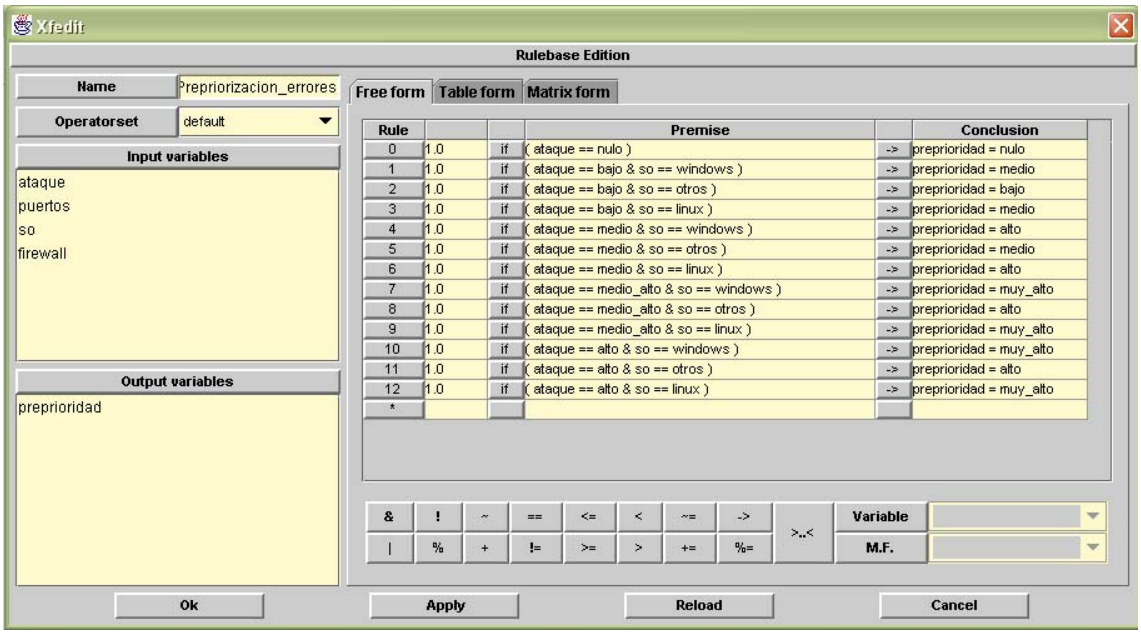
- *Prepriorización-Negación de Servicio:*



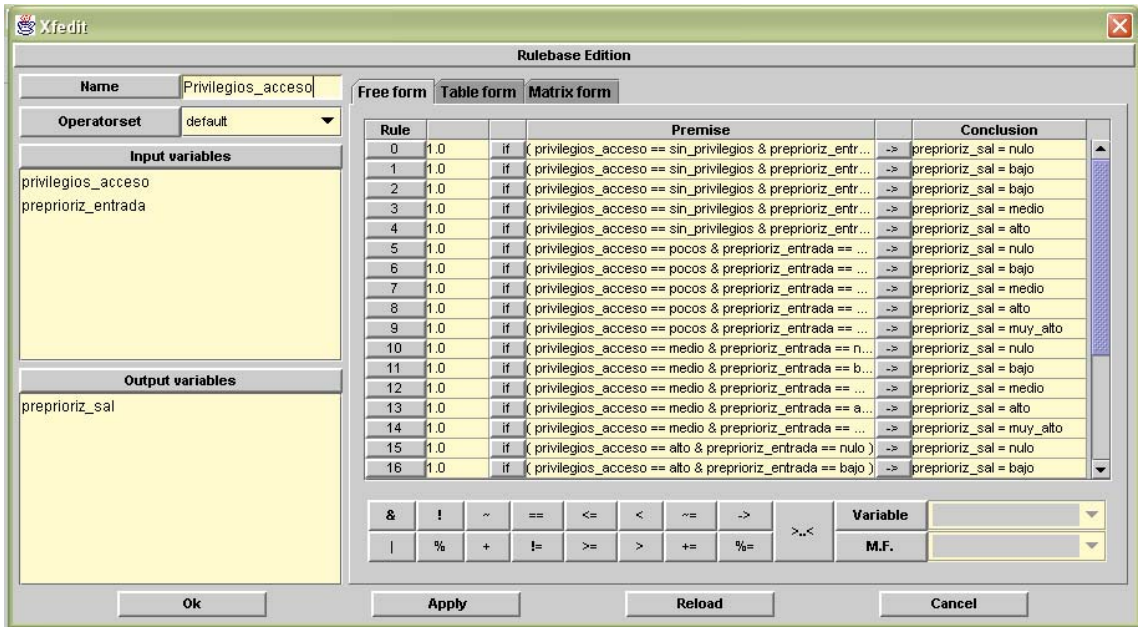
- Prepriorización-Modificación/Daño:



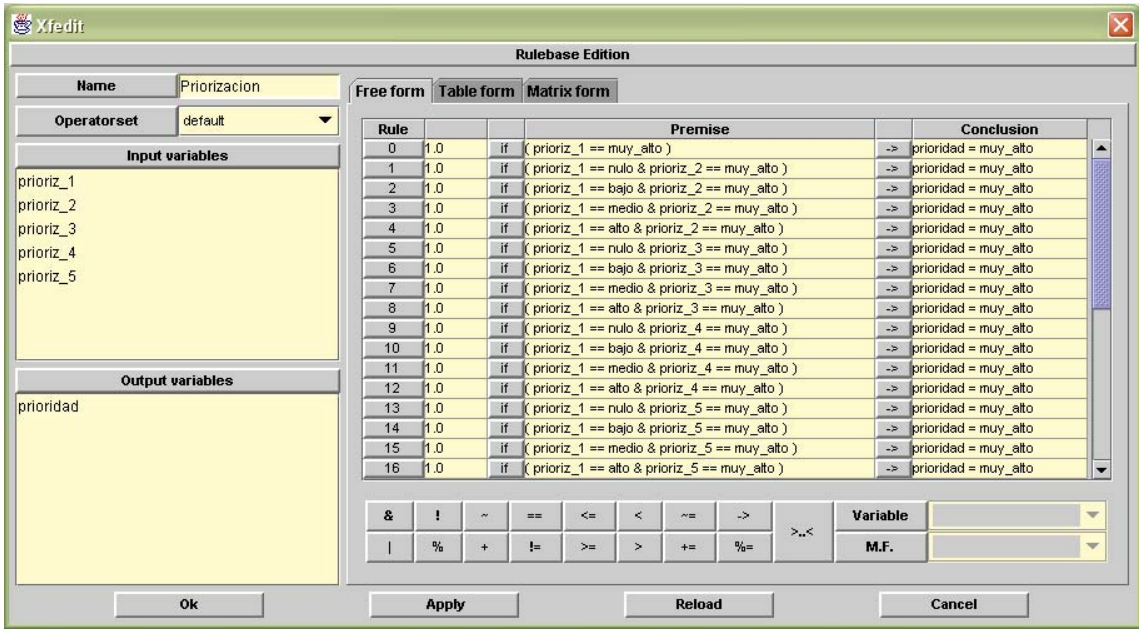
- *Prepriorización-Explotación de Errores:*



Las salidas serán las preprioridades (variables intermedias), que se evalúan junto con los privilegios de acceso (variable de entrada) en la base de reglas Privilegios_acceso que se detalla a continuación.

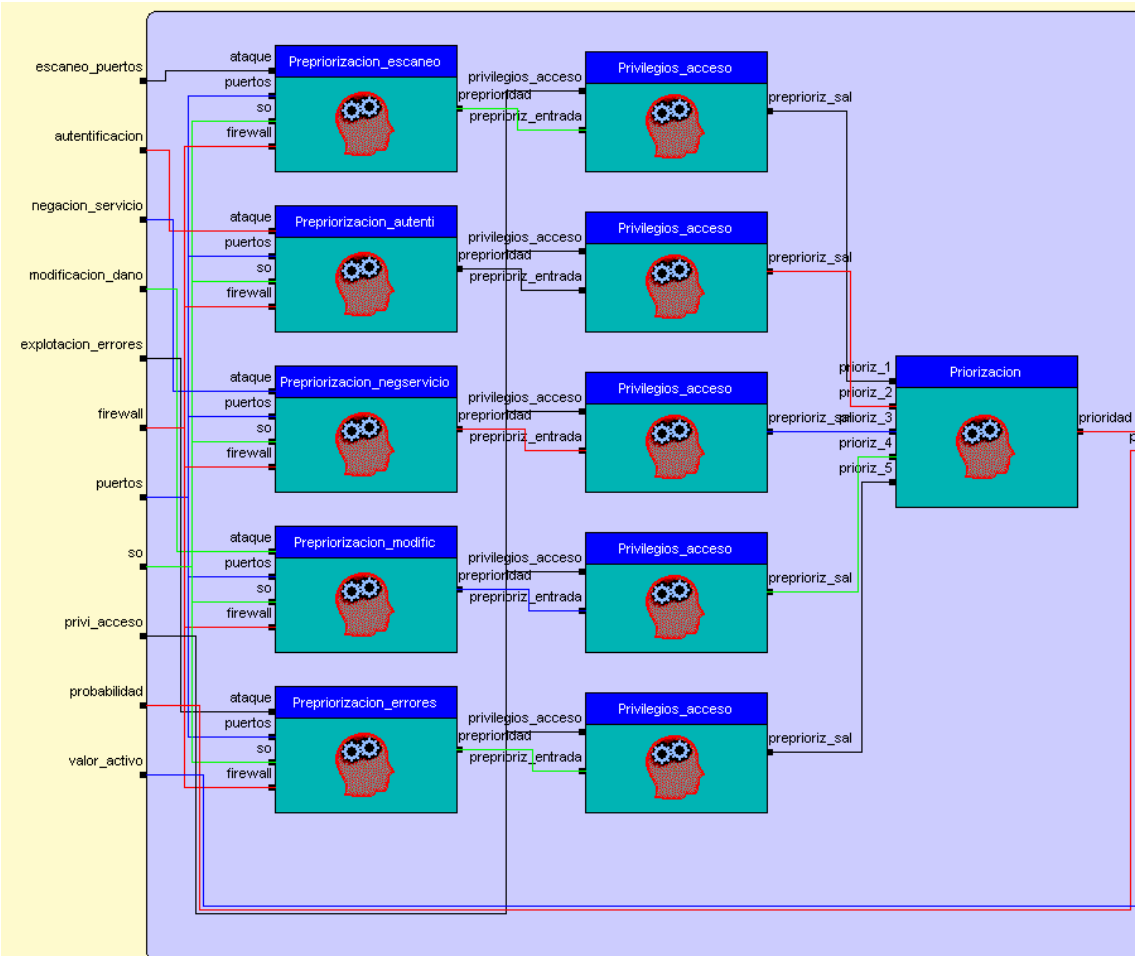


Como resultado tendremos 5 salidas de priorización intermedias que serán agrupadas en una base de reglas, Priorización, que de cómo salida el valor de la priorización final. Se detalla esta base de reglas del siguiente modo.


Xfedit
Rulebase Edition
Name
Priorizacion
Operatorset
default
Input variables
prioriz_1
prioriz_2
prioriz_3
prioriz_4
prioriz_5
Output variables
prioridad
Free form
Table form
Matrix form
Rule
Premise
Conclusion
0 1.0 if (prioriz_1 == muy_alto) -> prioridad = muy_alto
1 1.0 if (prioriz_1 == nulo & prioriz_2 == muy_alto) -> prioridad = muy_alto
2 1.0 if (prioriz_1 == bajo & prioriz_2 == muy_alto) -> prioridad = muy_alto
3 1.0 if (prioriz_1 == medio & prioriz_2 == muy_alto) -> prioridad = muy_alto
4 1.0 if (prioriz_1 == alto & prioriz_2 == muy_alto) -> prioridad = muy_alto
5 1.0 if (prioriz_1 == nulo & prioriz_3 == muy_alto) -> prioridad = muy_alto
6 1.0 if (prioriz_1 == bajo & prioriz_3 == muy_alto) -> prioridad = muy_alto
7 1.0 if (prioriz_1 == medio & prioriz_3 == muy_alto) -> prioridad = muy_alto
8 1.0 if (prioriz_1 == alto & prioriz_3 == muy_alto) -> prioridad = muy_alto
9 1.0 if (prioriz_1 == nulo & prioriz_4 == muy_alto) -> prioridad = muy_alto
10 1.0 if (prioriz_1 == bajo & prioriz_4 == muy_alto) -> prioridad = muy_alto
11 1.0 if (prioriz_1 == medio & prioriz_4 == muy_alto) -> prioridad = muy_alto
12 1.0 if (prioriz_1 == alto & prioriz_4 == muy_alto) -> prioridad = muy_alto
13 1.0 if (prioriz_1 == nulo & prioriz_5 == muy_alto) -> prioridad = muy_alto
14 1.0 if (prioriz_1 == bajo & prioriz_5 == muy_alto) -> prioridad = muy_alto
15 1.0 if (prioriz_1 == medio & prioriz_5 == muy_alto) -> prioridad = muy_alto
16 1.0 if (prioriz_1 == alto & prioriz_5 == muy_alto) -> prioridad = muy_alto
& ! ~ == <= < ~= ->
| % + != >= > += %=
>..<
Variable
M.F.
Ok
Apply
Reload
Cancel

MAPA CONCEPTUAL – PRIORIZACIÓN

Aquí tenemos el esquema final correspondiente a los procesos y variables anteriores.



Subsistema de Valoración de Riesgos

El riesgo que supone un evento depende de la amenaza de dicho evento y del valor del activo que es atacado. La amenaza se calculará en función de la probabilidad de que el evento ocurra y la prioridad obtenida del subsistema anterior para dicho evento, Por tanto, tendremos las siguientes variables de entrada:

- **Prioridad**
- **Probabilidad**
- Valor del **Activo**

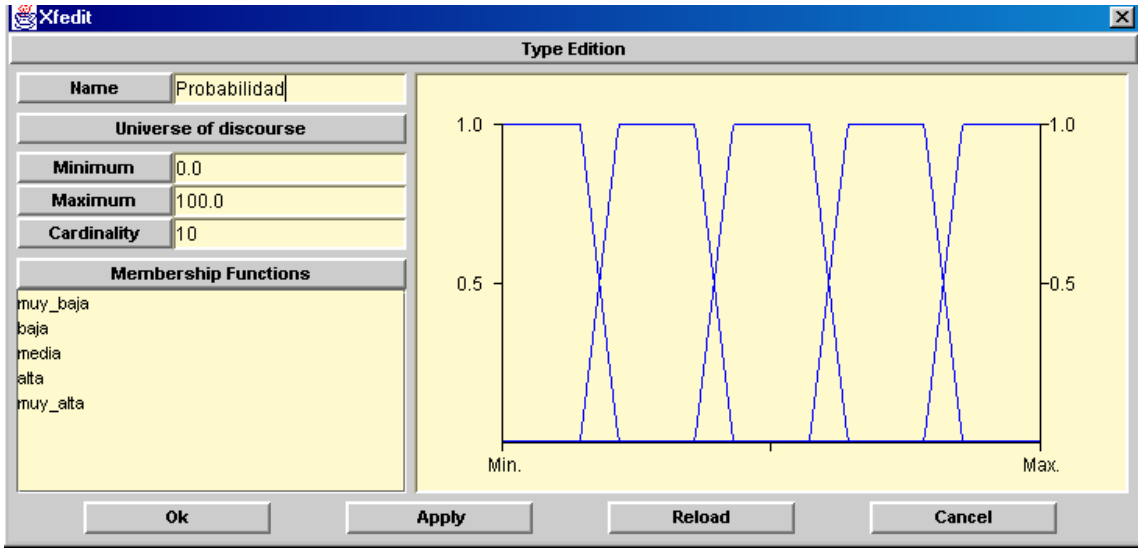
Prioridad

La prioridad de un evento ha sido calculada por el subsistema anterior y no necesita de más explicación. Solo decir que es una variable interna del sistema borroso y ha sido definida a partir del tipo riesgo.

Probabilidad

Se trata de la probabilidad de que este ocurriendo en ese momento el evento, y se refiere a la fiabilidad que tenemos en los sensores encargados de detectar los ataques.

Este tipo se define como se muestra a continuación.



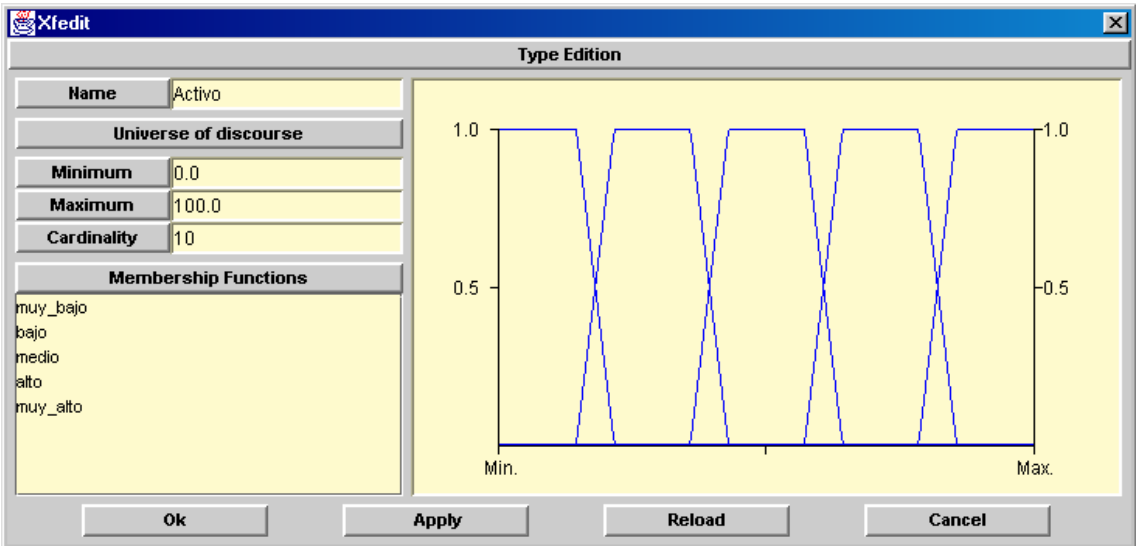
Valor del Activo

Los activos de la organización son los elementos que posee dicha organización para desarrollar su actividad, no se pretende ser demasiado rigurosos en la definición.

Aplicados a los sistemas informáticos, los activos serían todos los elementos que componen las diferentes redes y sistemas de la organización y que, además, son susceptibles de ser atacados por los intrusos. Como ejemplo podríamos nombrar a las diversas bases de datos de la organización, a los servidores, las estaciones de trabajo de los empleados,...

Como es obvio, no todos los activos tendrán el mismo valor, habrá elementos críticos para la organización que tendrán un valor muy alto. Cualquier ataque hacia ellos debe ser examinado cuidadosamente. También existirán activos de valor menor que exigirán menos atención.

Definimos el tipo Activo de la siguiente manera.



Amenaza

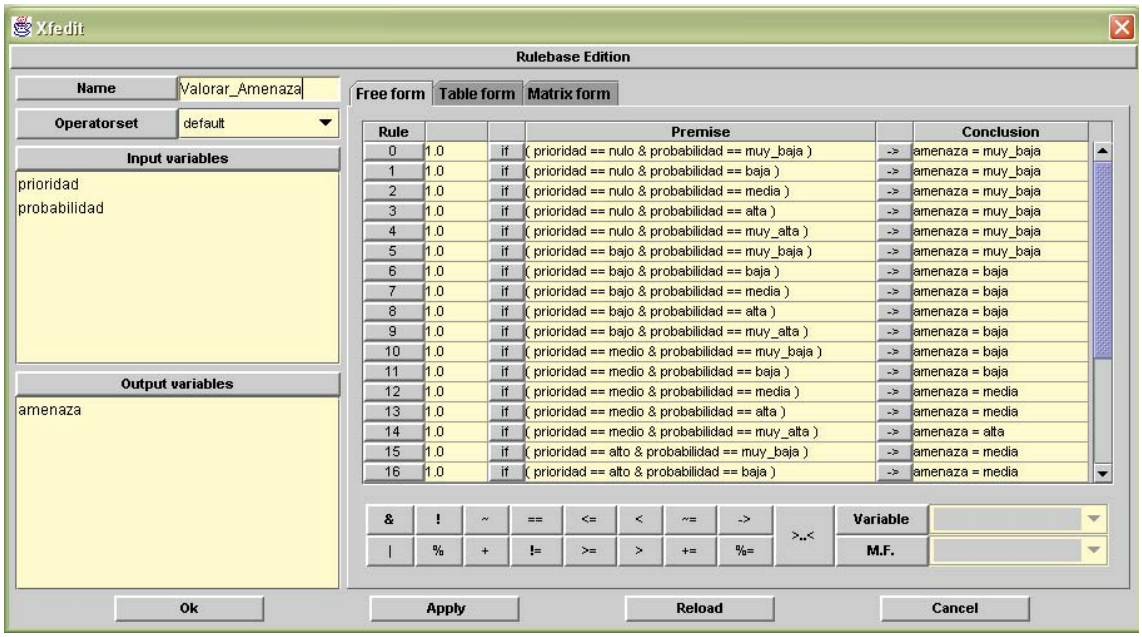
La amenaza es una variable intermedia del sistema y se calcula a partir de la probabilidad del evento y la prioridad calculada en el subsistema de priorización. Es decir, la amenaza de un evento depende de la prioridad que tiene dicho evento y la probabilidad de que este ocurriendo en este momento.

La amenaza esta definida según el tipo probabilidad, ya que comparten un universo del discurso similar.

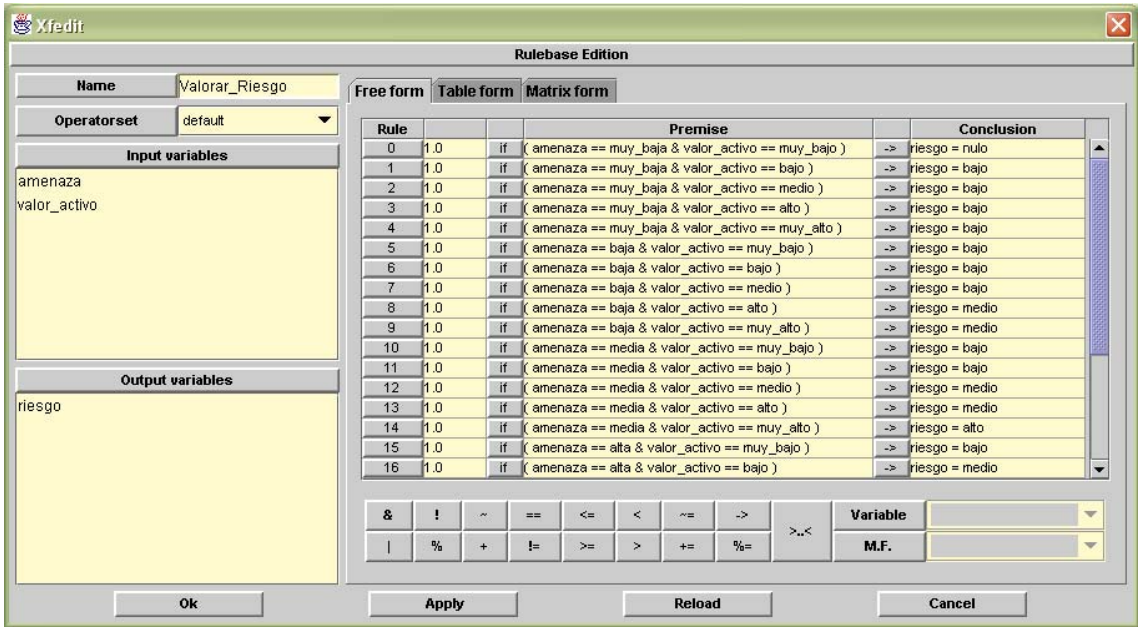
VARIABLES INTERMEDIAS Y FUNCIONAMIENTO DEL SUBSISTEMA

Se Incluyen aquí las bases de reglas usadas para el cálculo del riesgo.

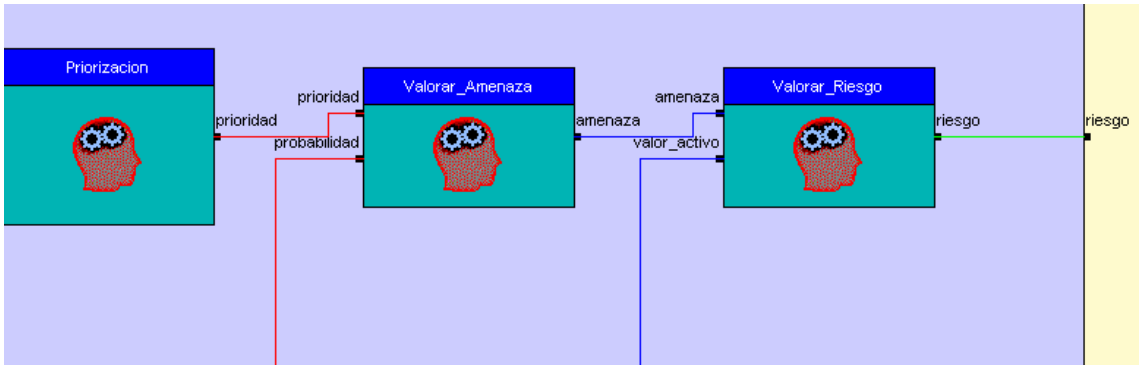
- *Valorar amenaza:*



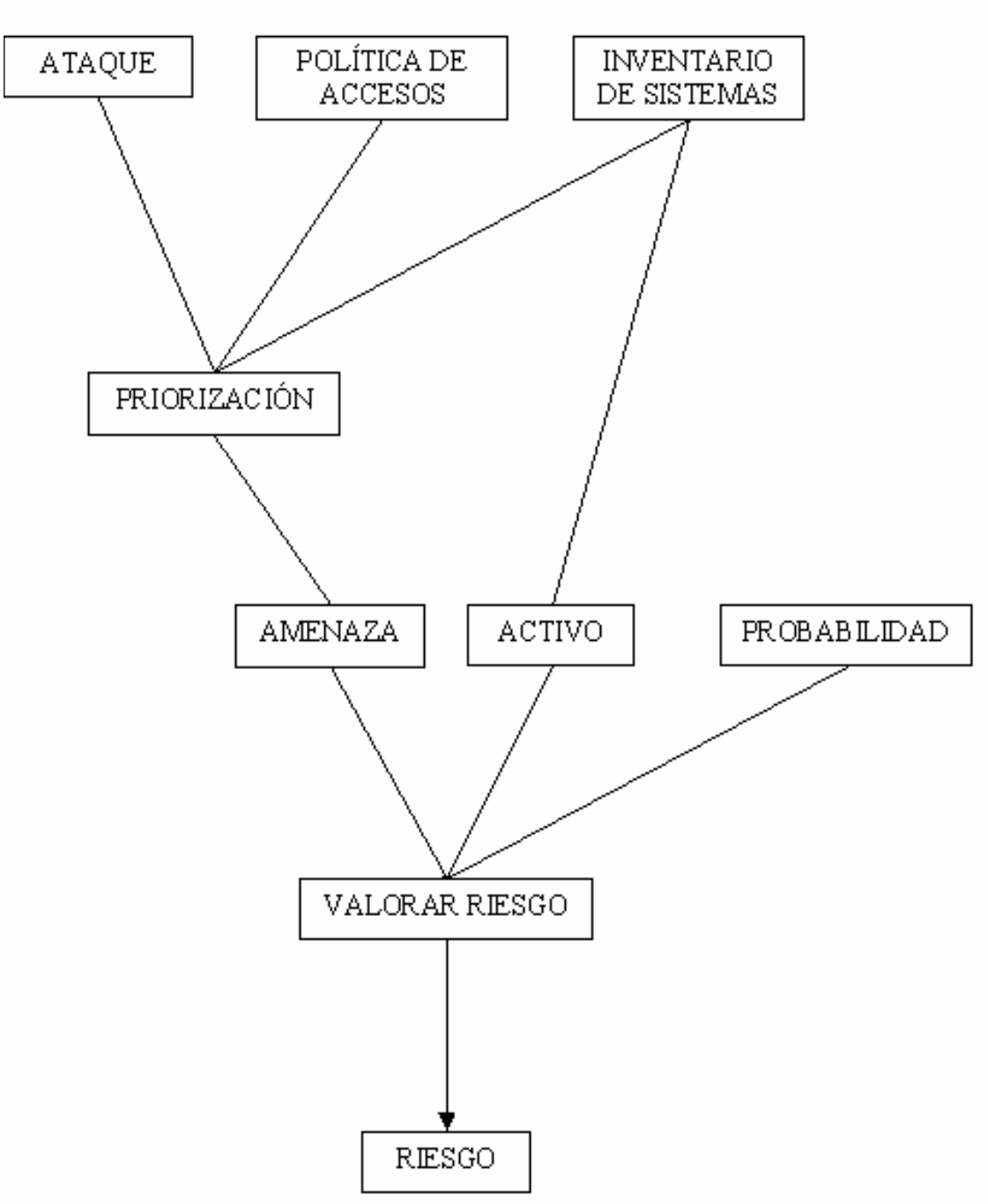
- *Valorar Riesgo:*



MAPA CONCEPTUAL – VALORACIÓN DE RIESGOS



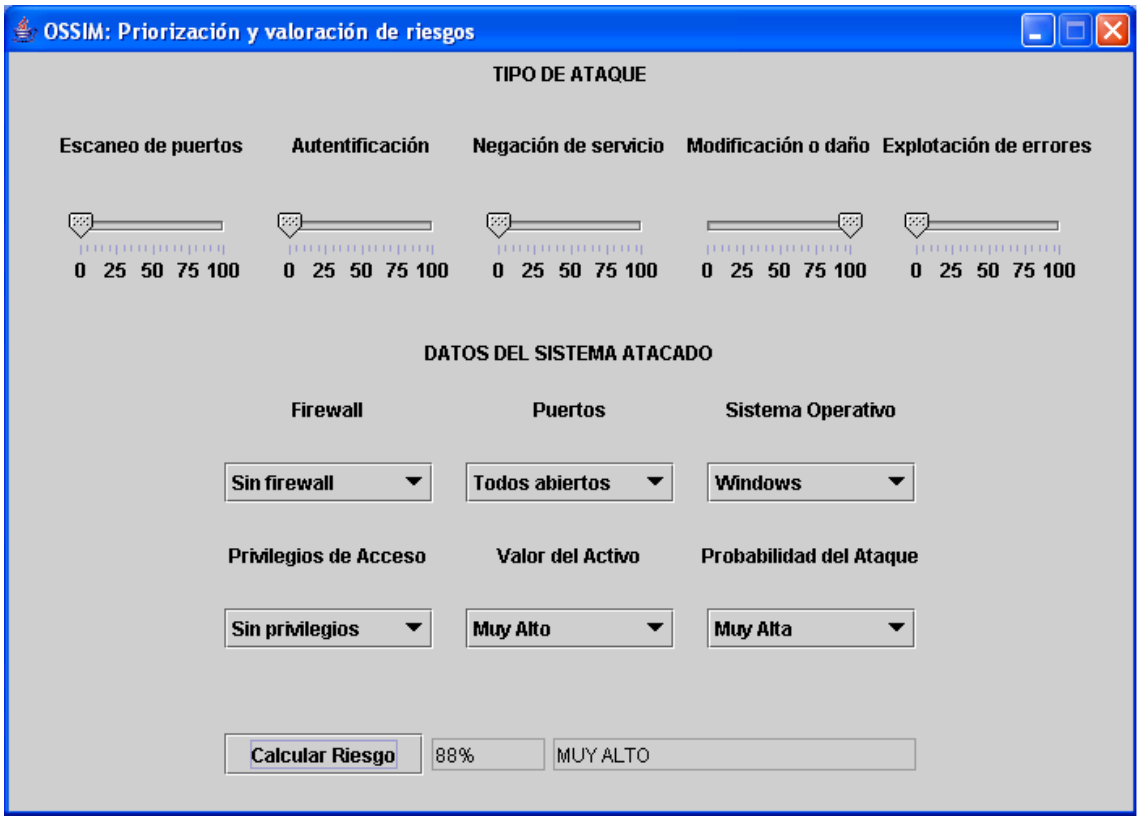
Mapa Conceptual - Roadmap



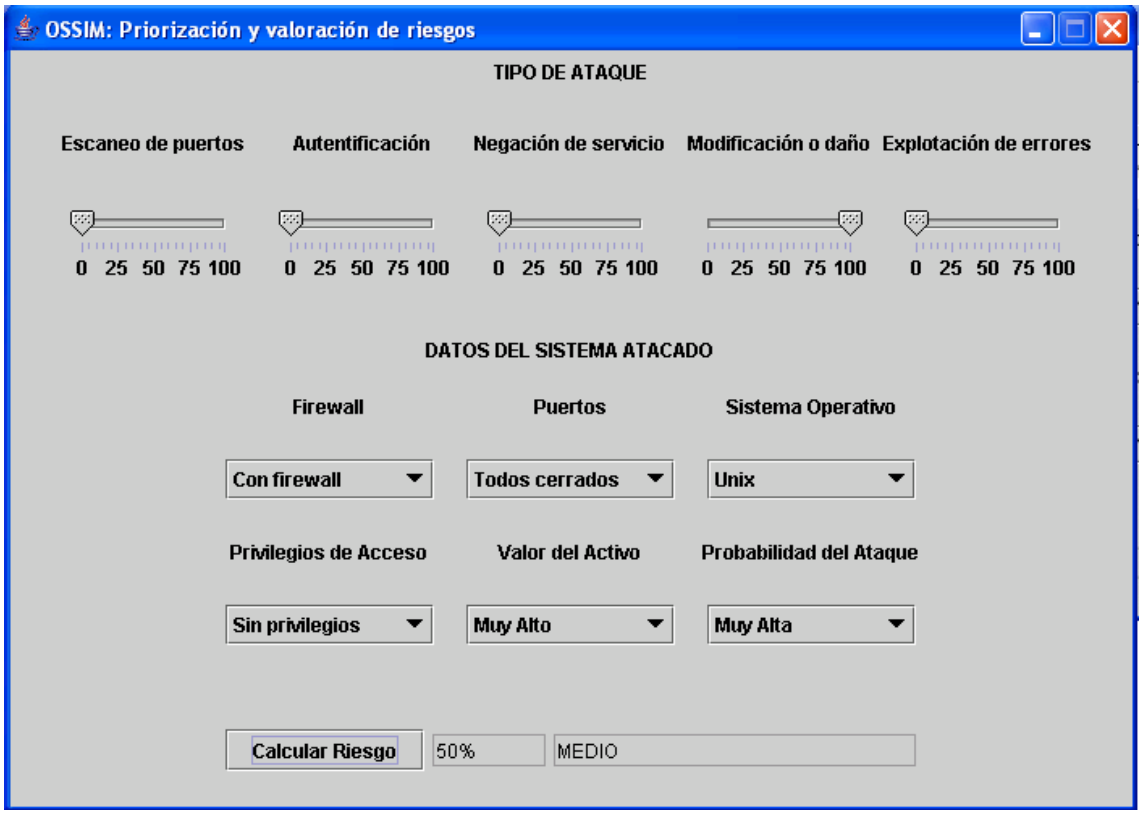
Caso de ejemplo

Se van a mostrar dos casos de ejemplo con el mismo ataque, prioridad y valor del activo, pero en uno de ellos el ordenador esta muy protegido y en el otro no dispone de ningún medio de seguridad.

1. En este caso el ordenador que recibe el ataque no dispone de ningún medio de defensa, bien sea firewall o los puertos cerrados.



2. Por el contrario, en este otro caso el ordenador cuenta con firewall y con los puertos cerrados pero el ataque es el más fuerte posible.



Fuentes Consultadas

- Proyecto OSSIM:

  - Descripción General del Sistema

  - Implementación OSSIM

  - Página web: www.ossim.net.

- First International Workshop: Intelligent Systems for intrusión detection and facing malware. Universidad de León.

- Sistemas Operativos. William Stallings

- La información referente a los tipos de ataque ha sido extraída de la pagina: http://webs.ono.com/usr016/Agika/index.html
  Esta información ha sido contrastada en diversas páginas webs de seguridad informática y prácticamente no ha sido modificada, sólo actualizada cuando se ha requerido.