Sistemas Informáticos
Curso 2003-04

RoccoFirewall:
Firewall OpenSource en Windows

Carlos BARDASANO GONZÁLEZ
Mª. Pilar BRAVO CONTRERAS
Jaime MARTÍNEZ MURADÁS

Dirigido por:
Prof. Ignacio Martín Llorente
Dpto. Arquitectura de Computadores y Automática

Facultad de Informática
Universidad Complutense de Madrid

Trautman: “How will you live, John?“
Rambo: “Day by day.”

INDICE

INTRODUCCION

1. Resumen

Cada día más ordenadores están conectados a algún tipo de red. Esto mejora mucho la calidad de vida y de comunicación de muchas personas, pero a la vez pone en peligro muchos datos e información privada. Normalmente cualquier documento de papel importante se guarda bien seguro en una caja fuerte, pero los documentos electrónicos se dejan en el ordenador sin protección ninguna, pensando “Esto nunca me va a pasar a mi”. O si a alguien un poco más precavido se le ocurre buscar, solo encuentra soluciones comerciales que no siempre cumplen con las expectativas o que cuestan mucho dinero.

El objetivo de RoccoFirewall es conseguir un sistema de filtrado de paquetes de red no deseados en base a una serie de reglas introducidas por el usuario que sea:

- Sencillo pero potente.
- Robusto pero flexible.
- Seguro pero eficiente.

En una palabra: **FIABLE**.

Para que además sea lo más útil posible al usuario, el código será abierto, es decir, cualquiera con conocimientos de C++ podrá modificarlo a su gusto para adaptarlo a sus necesidades.

2. Summary

Day by day, more computers are being connected to any net. This improves the quality of life and communication of many people, but also risks user’s data and private information. Usually, an important paper document, it is secured in a strongbox, but if an electronic document is important, it is stored into a computer without security. If a user, thinking in his privacy, search a security solution, only will find commercial solutions that don´t fit with his hope, or are very expensive.

The target of RoccoFirewall is to obtain a rule based non-desired net packet filter system, performed by the user, with these characteristics:

- Simple but powerful
- Robust but flexible
- Safe but efficient

In one word: **RELIABLE**.

RoccoFirewall also will be open source to allow user (with C++ language skills) to modify it in order to perform the operations to fit with his needs.

3. Palabras de referencia / Reference Words

Firewall, IDS, Filtrado, Paquete,Código abierto, Libre Distribución.

Firewall, IDS, Filtering, Packet, OpenSource, FreeWare.

Objetivo del documento

El objetivo de este documento es exponer y justificar el desarrollo del proyecto en base a unas necesidades previas observadas en la situación actual, así como fundamentar la viabilidad de éste mismo. Además se expondrá la planificación, líneas de desarrollo seguidas, definición a alto nivel de la arquitectura software obtenida, funcionamiento de la misma, comparativa respecto a otros productos y manual de instalación y uso.

No es objetivo de este documento el describir en profundidad los detalles técnicos, así como exponer el código obtenido. Para dicho fin se pone a disposición un sitio web que contendrá toda la información técnica necesaria.

4. Situación actual

La creciente proliferación de troyanos, virus y usuarios malintencionados, conlleva a los usuarios a preocuparse por la seguridad de sus entornos. Para los usuarios de Unix / Linux esta problemática no es nueva, pero aquellos que usan los sistemas operativos de Microsoft están empezando a mentalizarse de esta amenaza.

Algunos sistemas operativos como puede ser Windows XP, aportan un sistema rudimentario de seguridad que inhabilita el acceso desde el exterior por parte de cualquier otro sistema. Estas medidas totalitarias de protección causan problemas en ciertos aspectos como puede ser la compartición de datos a través de una red de área local, o el establecimiento de una VPN (Virtual Private Network).

Para una mejor gestión de la seguridad de las máquinas, existe software específico como Firewalls (cortafuegos) o IDS (Intrusion Detection System).

Los Firewalls son herramientas que filtran los paquetes que circulan por la tarjeta de red de la máquina y permiten o no su escalado hasta el nivel de aplicación.

Los IDS se encargan de lanzar alarmas y en algunos casos tomar medidas de contingencia ante situaciones que puedan ser potencialmente peligrosas para la integridad de la máquina. El reconocimiento de estas situaciones se realiza en base a patrones, por ejemplo, un exceso de peticiones a un mismo servicio puede desencadenar una alarma de intento de desbordamiento.

Cuando una máquina está conectada a una red, sobre todo a Internet, está siendo sometida a una serie de ataques casi de forma continua (se ha constatado que la media es de 1 ataque cada 2 minutos). Si a esto se le suman los ataques producidos de forma interna (troyanos), la maquina se encuentra en un peligro constante. Por supuesto, el usuario normal no se percata de dichos ataques debido a que, a menos que sean destructivos, no afectan al funcionamiento normal del sistema. Los ataques que no son destructivos, pueden ocasionar robo de datos como cuentas bancarias, tarjetas de crédito, o la manipulación remota de la máquina, pudiendo ser esta usada para fines poco propicios (spam, cracking, ataques DOS distribuidos...).

Partiendo de esta situación, se consultó a 2 empresas reales sobre dichos problemas:

- **Techrules.com**: empresa dedicada al desarrollo de software para la asesoría financiera, actualmente cuenta con unos 60 trabajadores.

- **GFI Informática**: empresa con más de 1000 trabajadores, dedicada a la prestación de servicios informáticos y al desarrollo de soluciones a medida.

La visión ofrecida por ambas empresas aumentó los focos de riesgo que se comentaban más arriba, añadiendo a estos el propio software que utilizaban, es decir, que en algunas ocasiones el comportamiento del software que se utiliza para el funcionamiento normal de la empresa, es similar al de un ataque. En concreto se expuso la problemática real surgida mediante el uso de un programa de comunicación interna, sufrida por una de estas 2 empresas.

La empresa que lo sufrió trabaja mediante una línea punto a punto con otra empresa de telecomunicaciones de reconocimiento internacional. En el final de esta línea punto a punto, por la parte de la empresa de telecomunicaciones, se encuentra ubicado un firewall con funcionalidad de IDS.

La problemática surge cuando de repente el IDS corta la conexión debido a la detección de un intento de DOS (Denial Of Service) al servicio de DNS de la empresa de telecomunicaciones, había detectado entorno a las 400 peticiones/segundo por parte de la misma IP (el gateway de la otra empresa). Esto se debía a que el programa de comunicación interna, que esta empresa usaba, intentaba descubrir cuántos clientes de ese mismo programa había en la subred, haciendo unas 30 peticiones DNS cada cierto tiempo. Si esto lo multiplicamos por 300 equipos obtenemos 9000 peticiones en 1 segundo en el peor de los casos.

Resulta que sabían las IP's que estaban haciendo las peticiones de DNS, por lo cual la máquina que era, hubiese sido tan fácil como cortar el trafico por el puerto DNS de esas máquinas, pero no tenían ninguna herramienta para hacerlo, ni encontraron ninguna que se pudiese descargar de Internet. Al no encontrar una forma de cortar dicho tráfico, tuvieron que desconectar los equipos de la red y los trabajadores dejar sus obligaciones hasta que se solucionase el problema, con la consecuente mala imagen que supone para el cliente y la pérdida de tiempo en los desarrollos.

Si hubieran tenido una herramienta para Windows que de forma sencilla permitiera filtrar el tráfico de red en base a reglas de protocolos, este problema se habría resuelto mucho más rápido.

Esto no es más que la punta del iceberg, ya que existen muchos más problemas que surgen cada día, tales como un nuevo fallo en un servicio de Windows que escucha en un puerto X que provoca el reiniciado del sistema, pues con prohibir las conexiones a ese puerto estaría solucionado el problema hasta que se publique el esperado parche.

Existen muchas aplicaciones profesionales que realizan la función de firewall o IDS, o las 2 a la vez, también es cierto que son extremadamente caras, no suelen incluir soporte y los parches se retrasan bastante.

Descendiendo un poco más en la escala, se encuentran las soluciones gratuitas, menos funcionales y de escasa utilidad para el usuario medio, la gran mayoría de ellas. Además, ninguna de las soluciones gratuitas puede utilizarse en entornos empresariales.

5. Requisitos

Tras las entrevistas con las empresas y la propia experiencia personal, se obtienen ciertas **características deseables** en aplicaciones de este tipo.

- **Facilidad de uso**: se valorará la simplicidad de su manejo y la curva de aprendizaje para sacar rendimiento a la aplicación.
- **Multiplataforma**: plataformas Windows en las que puede ser instalado.
- **Especialización**: se valorará el nivel de detalle hasta el que permite definir reglas de filtrado.
- **Información al usuario**: se valorará la claridad de información que retransmite al usuario sobre lo que está ocurriendo.
- **Adaptabilidad**: se valorará la capacidad de actualización y evolución ante nuevos problemas.
- **Utilización en entornos empresariales**: se valorará si puede ser utilizado en entornos de desarrollo empresarial, e incluso producción.
- **Rendimiento**

Además se considera necesario, debido a las largas esperas para la publicación de parches, la característica:

- **Código Abierto** (permite la evolución, como los proyectos GNU)

Y por último, la cual puede deducirse prácticamente de la anterior, sería que la aplicación fuese de libre distribución.

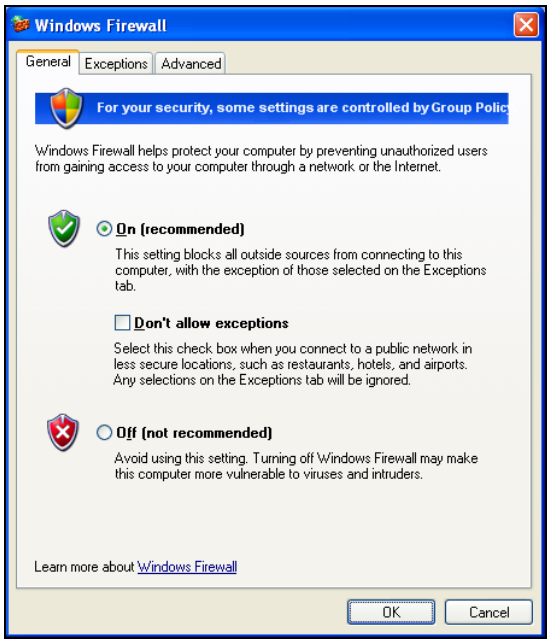
- **Libre distribución** (se mitigan los costes)

6. Productos en el mercado

Dado que el proyecto a desarrollar será de libre distribución, sólo se expondrán aquellas aplicaciones de libre distribución (más famosas) existentes en el mercado.

WINDOWS FIREWALL

Antes conocido como Internet Connection Firewall (ICF), se proporciona junto con Windows XP. Windows lo activa por defecto para todas las conexiones de red a partir del Service Pack 2.



Es un firewall basado en host que bloquea todo el tráfico entrante no solicitado que no corresponde a tráfico enviado como respuesta a una petición de la máquina. No bloquea tráfico de salida, excepto algunos mensajes ICMP (Internet Control Message Protocol)

Está activado globalmente por defecto. Esto quiere decir que todas las conexiones de un ordenador que corre bajo Windows XP SP2 tienen el firewall activado, incluyendo LAN's, módems y conexiones VPN (Virtual Private Network). Además las nuevas conexiones también tienen activado el firewall por defecto.

También se puede prohibir el tráfico de una aplicación concreta especificando el conjunto de puertos TCP/UDP que usa la aplicación en cuestión o bien el nombre de la aplicación.

Funcionalidades

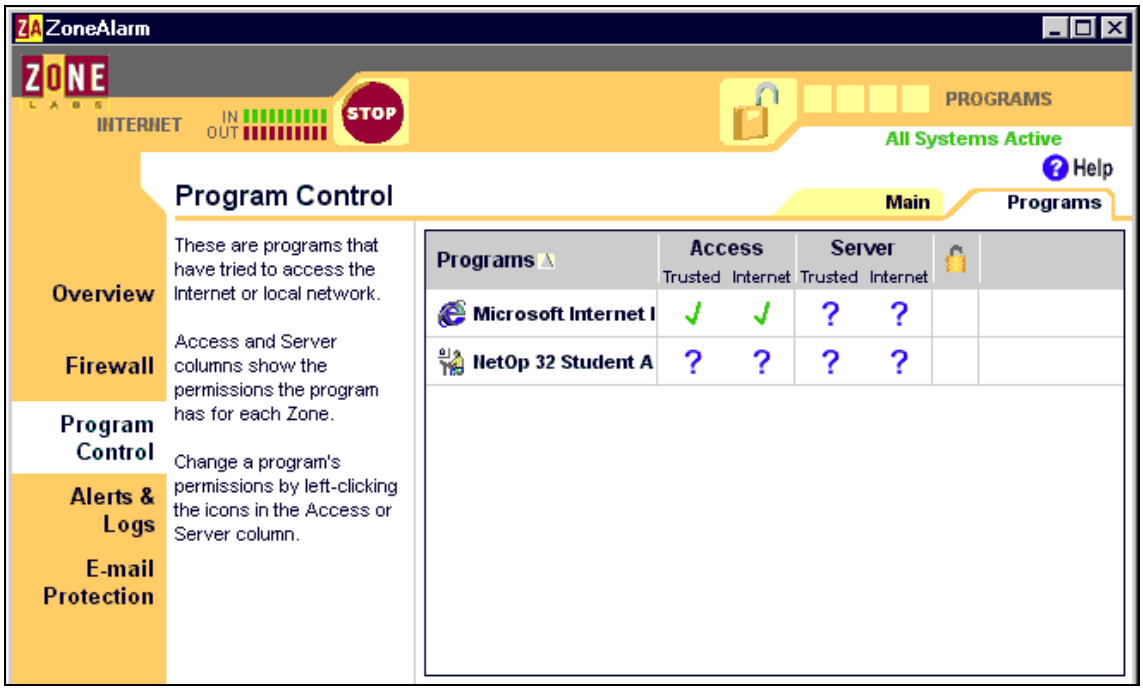
- Permite bloquear todo el tráfico no solicitado, sin excepción.

- Permite especificar programas que tienen permitido el tráfico.
- Especifica si se permite a los administradores locales configurar sus propias excepciones por programa.
- Permite especificar si se guarda la actividad del firewall en un log.
- Permite especificar una lista de puertos abiertos.
- Actualizaciones de frecuencia media.
- Se puede usar en entornos empresariales.
- Usa Drivers NDIS IM.
- Sólo disponible para Windows XP.

  http://www.microsoft.com

ZONE ALARM

Es un programa muy sencillo de usar, que no precisa de una configuración complicada para empezar a funcionar. Además no se necesita reiniciar el sistema una vez instalado. Después de la instalación se ejecuta un tutorial en el cual se enseña en siete pasos el funcionamiento del sistema. Una vez completado el tutorial, el firewall está listo para su ejecución.



El funcionamiento es sencillo pero tedioso, simplemente avisa si se recibe o envía tráfico perteneciente a una fuente aún no chequeada y pide permiso para dejarlo pasar. El usuario puede elegir si permite el tráfico o no, el cual no sabe muy bien cuando debe o no.

Toda esta actividad se guarda en un log de alertas.

Funcionalidades

- Especificar nivel de seguridad bajo, medio o alto.
- Permite especificar programas que tienen permitido el acceso a Internet.
- Se puede seleccionar si se desea que el programa avise cada vez que bloquea alguna comunicación.

- Guarda en un log toda la actividad de tráfico bloqueado.
- Bloqueo de emergencia bloquea todas las comunicaciones.
- Actualizaciones de frecuencia media.
- No se puede usar en entornos empresariales
- Usa Drivers TDI.
- Consume bastantes recursos del sistema.
- Compatible con todas las versiones de Windows a partir de Windows 98.

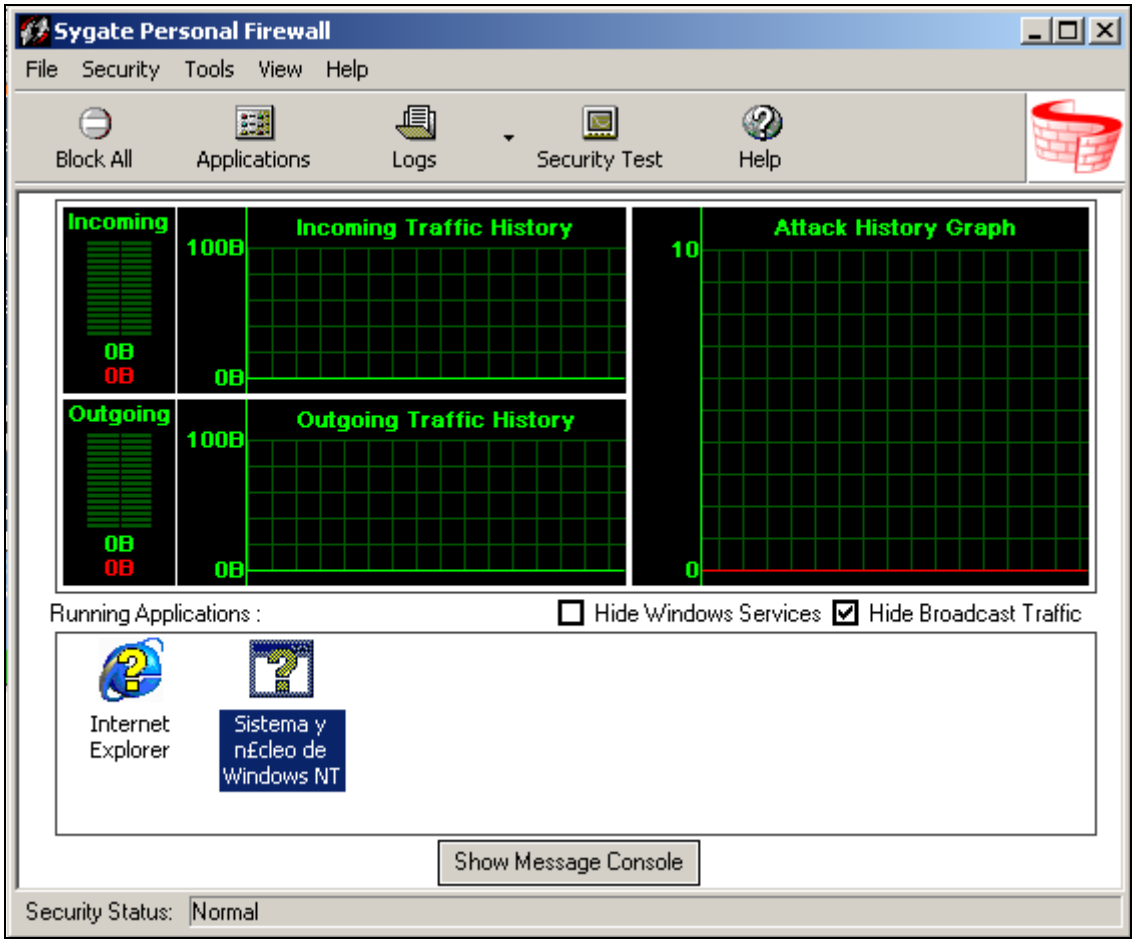
http://www.zonelabs.com

SYGATE PERSONAL FIREWALL

Posiblemente ésta sea la mejor herramienta gratuita de las mencionadas anteriormente. La última vez que se intentó acceder a está herramienta ya no estaba disponible como libre distribución.

El modo de funcionamiento es similar al mostrado por Zone Alarm, es decir, pregunta insistentemente sobre cada programa que quiere acceder a la red o desde la red. A pesar de este comportamiento, que puede ser confuso para el usuario medio y tedioso para el usuario avanzado, dispone de una configuración avanzada mediante reglas de filtrado que permiten definir con todo detalle las características del filtrado.

Muestra información constante por pantalla de todo lo que ocurre.



Funcionalidades

- Permite especificar programas que tienen permitido el acceso a Internet.

- Permite especificar reglas avanzadas.
- Se puede seleccionar si se desea que el programa avise cada vez que bloquea alguna comunicación.
- Muestra por pantalla y guarda en un log toda la actividad de tráfico bloqueado.
- Bloqueo de emergencia bloquea todas las comunicaciones.
- Actualizaciones periódicas.
- No se puede usar en entornos empresariales
- Usa Drivers TDI.
- El consumo de recursos de sistema es moderado.
- Compatible con todas las versiones de Windows a partir de Windows 98.

  http://www.sygate.com

Se pretende pues obtener una aplicación que reúna los siguientes requisitos:

- **Facilidad de uso**
- **Multiplataforma**
- **Especialización**
- **Información al usuario**
- **Adaptabilidad**
- **Utilización en entornos empresariales**
- **Rendimiento**
- **Código Abierto**
- **Libre distribución**

Dicha aplicación estará orientada al usuario de nivel medio/alto, el cual debe tener un conocimiento más profundo de lo que sucede en su máquina, así como ciertos conocimientos de protocolos de red.

7. Alternativas de implementación

Tras fijar los requisitos del desarrollo se investigó sobre las diferentes alternativas de implementación. Todas las alternativas comparten la misma estructura: una aplicación que ejecutará en “user-mode” para comunicarse con el usuario y un componente que dependiendo de la alternativa ejecutará en “kernel-mode” o en “user-mode”. Por lo cual, las alternativas se distinguirán en base al modo de ejecución del componente.

A. User-Mode

1. Usar las librerías **Winsock Layered Service Provider**: no se puede acceder al contenido de los paquetes de red.

2. Usar la librería **Windows 2000 Packet Filtering Interface**: librería ya desarrollada específicamente para acciones de filtrado de paquetes de red, puede especificar reglas basadas en IP origen e IP destino además de puertos TCP. No soporta otros parámetros de filtrado.

3. Usar **Replacement DLL’s**: consiste en reemplazar ciertas sublibrerías de la librería Winsock para añadir funcionalidad. Esta opción, aparte de no estar apenas documentada, sigue sin poder acceder a los paquetes de red.

B. Kernel-Mode

1. Implementar un **NDIS Transport Data Interface driver (TDI)**: el driver desarrollado tendría acceso completo a los paquetes de red, pero estaría ubicado encima de la capa TCP/IP del núcleo de Windows, pudiendo pues ser puenteado mediante la conjunción de un driver que estuviese por debajo y otro por encima.

2. Implementar un **NDIS Intermediate Driver (IM)**: el driver desarrollado tendría acceso completo a los paquetes de red y se ubicaría justo encima del driver de la tarjeta de red, por lo cual no podría ser puenteado. Su desarrollo en mas complicado ya que estos drivers necesitan por un lado comunicarse con el driver de la tarjeta de red (actuando como un TDI) y por otro comunicarse con los driver que componen la capa de transporte (actuando como un Miniport Driver). Aparte de la dificultad de desarrollo, el problema es que necesita ser firmado por Microsoft para poder ser instalado.

3. Usar el **Windows 2000 Filter Hook-Driver**: es un driver que implementa funciones de filtrado, sólo habría que añadir lógica. Su implementación sería la más sencilla pero sólo puede haber un driver de este tipo en cada sistema, por lo cual no se puede garantizar su instalación, ya que puede existir otra aplicación previamente instalada, que tenga un driver de este tipo.

4. Usar el **Windows 2000 Firewall Driver**: este driver está en versión Beta, sin documentación existente.

Todas las alternativas referentes al “User-Mode” quedan descartadas por su falta de funcionalidad y su escasez de rendimiento.

Dentro de las alternativas de “Kernel-Mode” se escogerá la número 2, el NDIS Intermediate Driver, ya que es el más seguro de todos y el que más puede aportar a la comunidad de libre distribución, ya que no existe absolutamente nada similar. Puede que dentro de unos años la opción más aconsejable sea el Windows Firewall Driver, pero queda mucho para que llegue ese día.

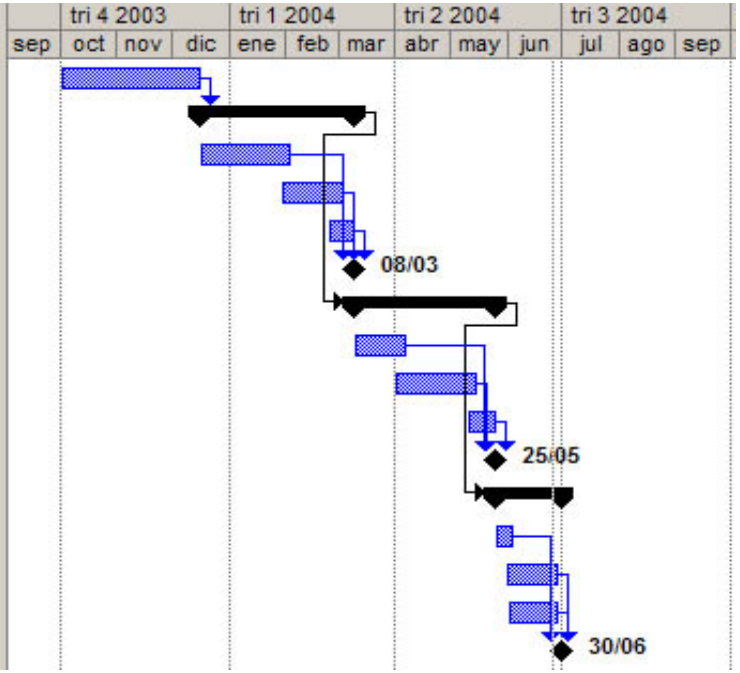
Más adelante en el documento se explicará con claridad y gráficos la estructura del núcleo de Windows y la zona de actuación del driver dentro de éste, así se podrá comprobar la justificación de la elección.

8. Planificación y riesgos

Desde el momento en el que se tomó la decisión de desarrollar el proyecto mediante un NDIS Intermediate Driver, se supo que la mayor parte de la planificación estaría dedica al estudio y documentación sobre el núcleo de Windows. Tras la fase inicial de instrucción, se decidió seguir un desarrollo de prototipos en varias fases, incrementando la funcionalidad de estos en cada entrega.

La planificación resultante se muestra a continuación:

		Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres
1		Fase inicial de instrucción	54 días	mié 01/10/03	lun 15/12/03		
2		**Iteración 1**	**60 días**	**mar 16/12/03**	**lun 08/03/04**	**1**	
3		Análisis y diseño	35 días	mar 16/12/03	lun 02/02/04		
4		Desarrollo	25 días	jue 29/01/04	mié 03/03/04		
5		Documentación	10 días	mar 24/02/04	lun 08/03/04		
6		Entrega de prototipo	0 días	lun 08/03/04	lun 08/03/04	3;4;5	
7		**Iteración 2**	**56 días**	**mar 09/03/04**	**mar 25/05/04**	**2**	
8		Análisis y diseño	20 días	mar 09/03/04	lun 05/04/04		
9		Desarrollo	32 días	jue 01/04/04	vie 14/05/04		
10		Documentación	12 días	lun 10/05/04	mar 25/05/04		
11		Entrega de prototipo	0 días	mar 25/05/04	mar 25/05/04	8;9;10	
12		**Iteración 3**	**26 días**	**mié 26/05/04**	**mié 30/06/04**	**7**	
13		Análisis y diseño	7 días	mié 26/05/04	jue 03/06/04		
14		Desarrollo	21 días	lun 31/05/04	lun 28/06/04		
15		Documentación	20 días	mar 01/06/04	lun 28/06/04		
16		Entrega final	0 días	mié 30/06/04	mié 30/06/04	13;14;15	



Como en todo proyecto, existen ciertos factores que pueden cuestionar el alcanzar las metas propuestas. El objetivo de la lista de riesgos que se muestra a continuación, es la de identificación y control de cualquier situación de éste tipo que pueda surgir a lo largo del desarrollo del proyecto y que pueda repercutir en el éxito de éste, proponer unas pautas para evitar dicha situación así como proponer una acción de contingencia en el caso de que ésta suceda.

Para lograr este objetivo, se identificarán tantos riesgos como sea posible, valorándolos por su criticidad, y se establecerá un plan de actuación para, en el caso de que finalmente ocurra algún acontecimiento de los aquí previstos, poder minimizar su efecto.

Riesgo	Impacto	Severidad	Probabilidad	Plan de actuación	Plan de Contingencia
Los desarrolladores desconocen el **funcionamiento interno del sistema operativo *Microsoft Windows***, sobre el que se realizará el FIREWALL	Retraso en las primeras entregas	Media	Alta	**Documentación** del equipo sobre el funcionamiento de Windows antes de comenzar con cualquier planificación o implementación	Replanteamiento del proyecto.
No se dispone del tiempo suficiente para la realización del proyecto, ya que la dedicación no es a tiempo completo.	Retraso en las entregas hasta la imposibilidad de llevar a buen término el proyecto	Alta	Media	Establecimiento de **objetivos mínimos realistas** y abordables	Eliminar requisitos no imprescindibles y aumentar número de horas de esfuerzo
Falta de coordinación entre los integrantes del grupo	Retraso en las entregas	Media	Media	El Jefe de Proyecto (profesor) debe establecer fechas de reunión y exigir informes periódicos.	Aumentar horas de esfuerzo hasta completar la entrega en curso
Abandono o pérdida de alguno de los integrantes del equipo	Retraso de las entregas hasta la imposibilidad de llevar a buen término el proyecto	Alta	Baja	Establecimiento de **objetivos mínimos realistas** y abordables	Eliminar requisitos no imprescindibles y aumentar número de horas de esfuerzo.
Pérdida de datos del desarrollo		Alta	Baja	Realizar copias de seguridad frecuentemente.	Retomar la última copia de seguridad y estimar si se puede alcanzar el estado que se perdió, si no es así, habría que replantear los objetivos.

9. Costes

Debido a que las horas de esfuerzo no van a ser compensadas económicamente, no se obtendrá absolutamente ningún beneficio económico de este desarrollo. Así pues, se expondrán los costes, los cuales están compuestos por los gastos de software, impresión y encuadernación.

Inicialmente, el equipo de desarrollo cuenta con un capital social de 180€, aportando pues 60€ cada integrante. Al final fue necesaria la aportación de una pequeña cantidad extra de capital, ya que el presupuesto no alcanzaba para cubrir los gastos.

A continuación se muestra un resumen de los gastos:

	Coste	**Aportado por**
Equipo informático	0,00 €	Equipo Desarrollo y UCM
Sistema Operativo		
Windows XP	0,00 €	UCM
Windows2000	0,00 €	UCM
Herramientas Ofimáticas		
Microsoft Office	0,00 €	UCM
Herramientas de Desarrollo comunes		
Microsoft VisualC++	0,00 €	UCM
Borland C++	0,00 €	UCM
Microsoft Source Safe 6.0	0,00 €	UCM
Herramientas Extra		
VMWare WorkStation 4	157,70 €	Equipo desarrollo
Microsoft Windows DDK	0,00 €	Microsoft
SourceGear SourceOffSite 3.5.3	0,00 €	SourceGear (Estudiante)
Gastos Reprografía		
Impresión	33,28 €	Equipo desarrollo
Encuadernación	72,42 €	
Total	263,40 €	

De este modo los gastos excedieron en 83,40 € al capital social aportado, debido a que no se habían estimado correctamente los gastos de reprografía, teniendo que realizar finalmente un desembolso de 87,80 € cada integrante del equipo.

DESARROLLO

1. Herramientas utilizadas

Durante el desarrollo de RoccoFirewall se han utilizado las siguientes herramientas y aplicaciones:

- Programación

  - **Microsoft Visual C++ 6.0**: Entorno de desarrollo de Microsoft que permite la creación de software en C y en C++. Utilizado para el desarrollo del controlador NDIS.
  - **Borland C++ Builder 5**: Entorno de desarrollo de Borland que permite la creación de software en C y en C++. Utilizado para el desarrollo del interfaz de usuario, por ser bastante más sencillo el diseño de formularios y ventanas en este entorno que en Visual C++.
  - **Microsoft Windows DDK** (Driver Development Kit): Son las librerías necesarias para desarrollar cualquier controlador para Windows.

- Control de versiones

  - **Microsoft SourceSafe 6.0**: Permite crear y gestionar repositorios con las diferentes versiones por las que va pasando el desarrollo, pudiendo revisar cambios entre versiones y recuperar cualquiera de ellas.
  - **Sourcegear SourceOffsite 3.5.3**: Permite conectar por Internet a un repositorio creado por SourceSafe.

- Pruebas y debug

  - **VMWare Workstation 4.5**: Esta aplicación es un emulador de un PC completo, al que se le puede instalar cualquier sistema operativo y funcionar con él. Esto es de gran utilidad a la hora de desarrollar un driver, ya que es bastante fácil (sobre todo sin tener experiencia) que haya algún error en la programación, algún desbordamiento de memoria o similar, que, en el caso de aplicaciones normales no es un problema muy grave, porque con el mismo debugger que viene incluido en el entorno de desarrollo y un poco de paciencia se puede encontrar donde está el fallo; sin embargo, en el caso de un controlador del sistema, cualquier fallo se traduce en un "cuelgue" del sistema, pudiendo dejarlo inutilizable. Con VMWare es posible hacer las pruebas ya que en el caso de que existan fallos, se puede recuperar un estado anterior sin que se vea afectado el resto del sistema y sin perder tiempo.

2. Conceptos básicos de la arquitectura

A continuación se muestra un diagrama con la estructura interna de Windows:

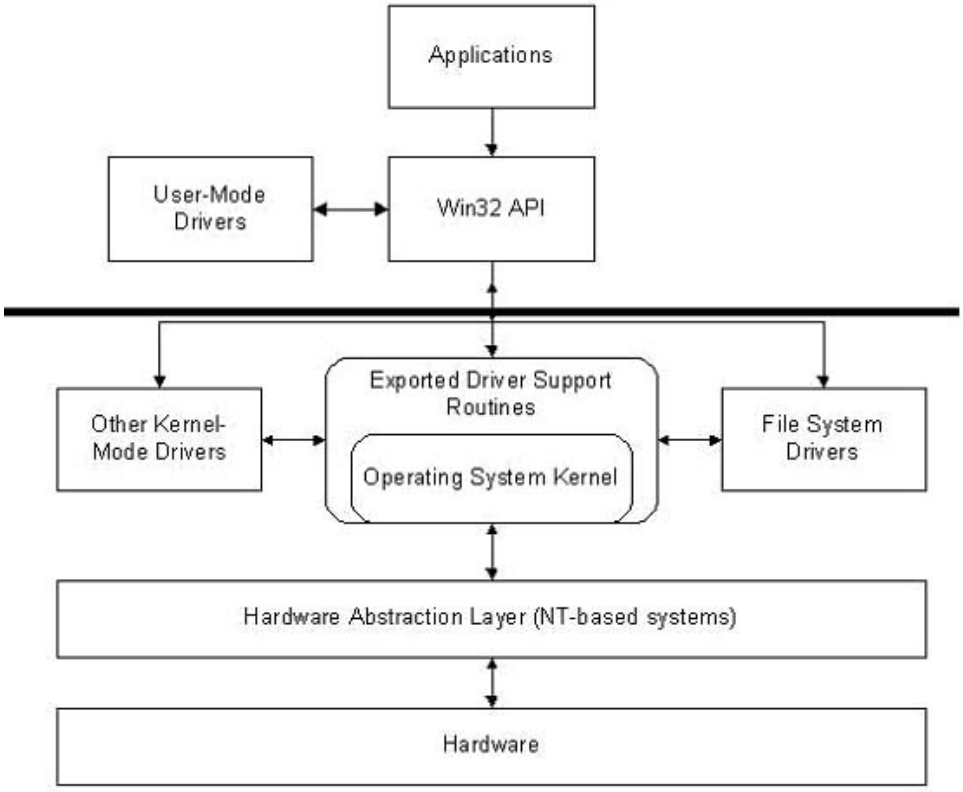


Fig A.

Como se puede apreciar, Windows dispone de una capa de abstracción (HAL) que impide el acceso directo al hardware por parte de los programadores (En realidad hay ciertos controladores que si necesitan el acceso al hardware por motivos de eficiencia o funcionalidad, sin embargo este no es el caso).

En Windows, para implementar los controladores de red se sigue el **NDIS**. El NDIS (Network Driver Interface Specification) se encarga del control de los adaptadores de red y de la comunicación entre los adaptadores de red y los drivers.

Existen 3 tipos de drivers de NDIS:

- **Miniport**: Son los drivers de más bajo nivel. Se relacionan directamente con las tarjetas de red y proporcionan un interfaz de más alto nivel para los demás drivers. Estos drivers los suelen realizar los fabricantes de tarjetas de red.

- **Protocolo**: Son los drivers de más alto nivel. Se relacionan con las aplicaciones.

- **Intermedio**: Como su propio nombre indica, estos drivers se sitúan entre los de protocolo y los Miniport, para añadir alguna funcionalidad o traducir el formato de los paquetes.

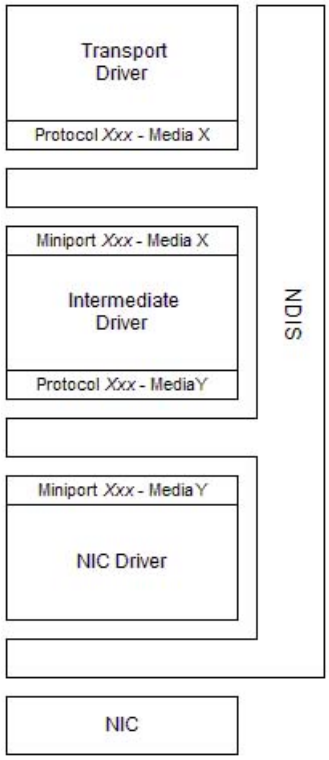


Fig. B

Dentro de los drivers intermedios hay dos subtipos:

- **Filtros**: Establecen una comunicación 1 a 1 con los adaptadores de red. Es decir, por cada adaptador de red real el driver expone un interfaz (adaptador de red virtual) para que los drivers superiores se comuniquen con él. Un ejemplo de este tipo de drivers es el "Programador de paquetes QoS" incluido en Windows 2000 y XP.

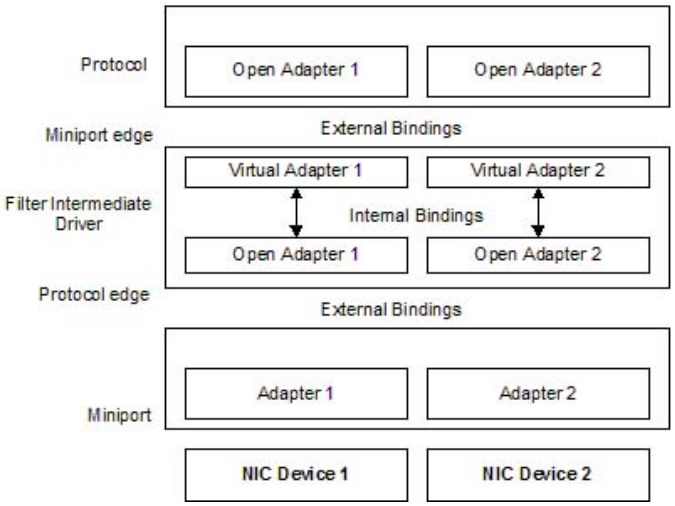


Fig. C

- **Multiplexores**: Establecen cualquier tipo de relación con los adaptadores de red reales. Los controladores de Balanceo de carga son un ejemplo de estos drivers.

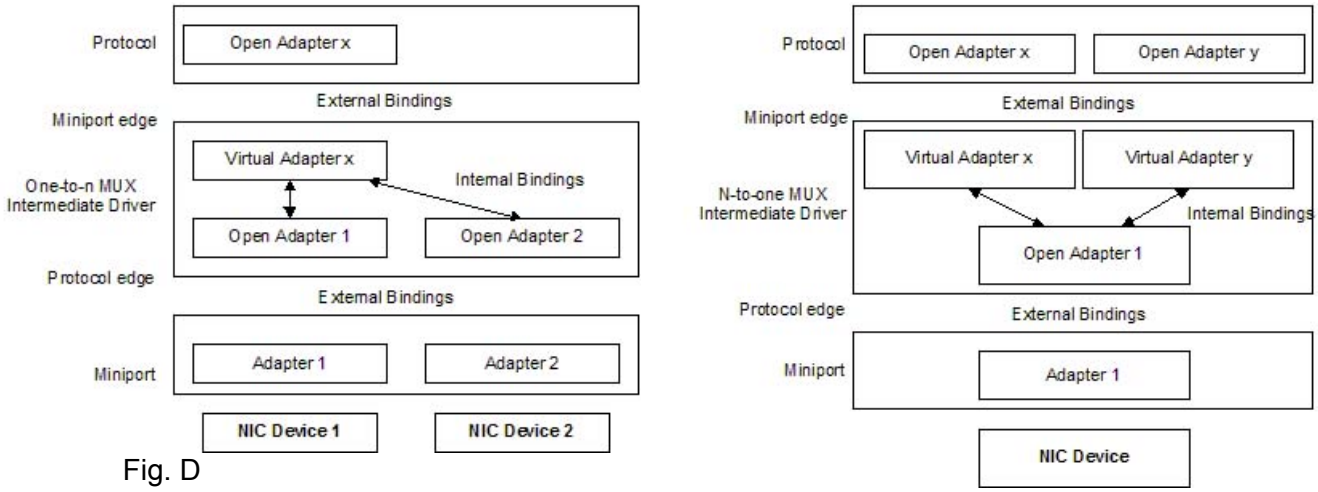


Fig. D

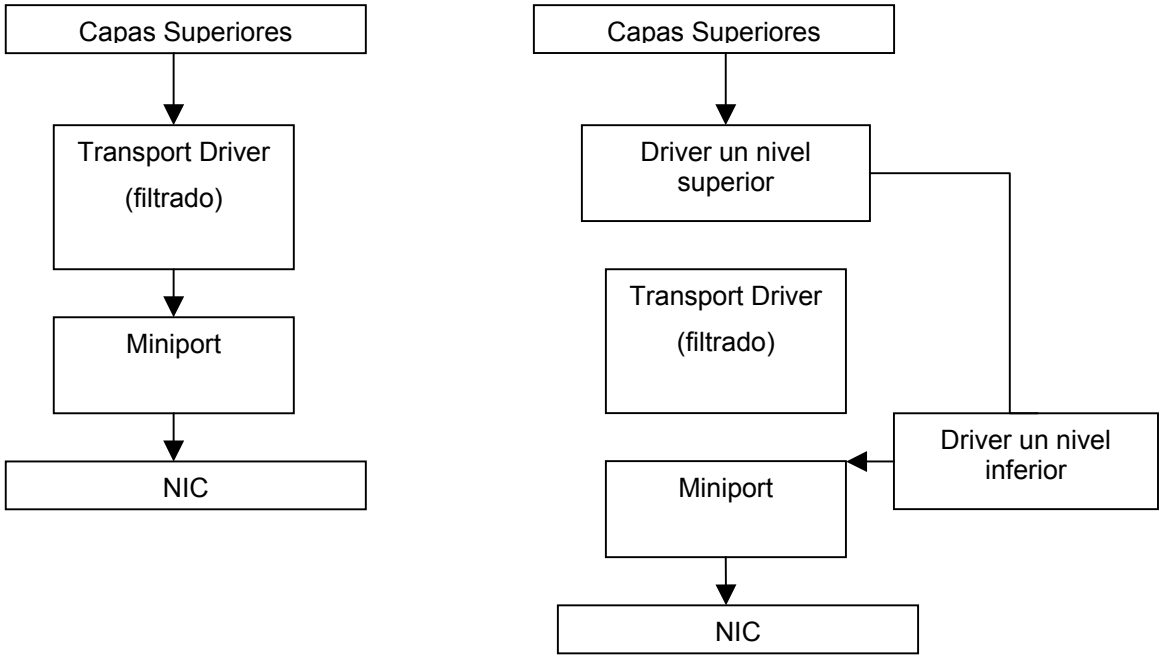
Como lo que se va a hacer es añadir la funcionalidad de "Filtrado de paquetes", se implementará un driver Intermedio de tipo Filtro.

A grandes rasgos, las características que debe tener un driver que filtre paquetes deben ser:

- **Analizar todos los paquetes**: Un sistema de seguridad como este no puede permitirse el dejar pasar un paquete sin analizarlo y tampoco puede perder ningún paquete recibido. Todos y cada uno de los paquetes que lleguen al interfaz de red tienen que pasar por el filtro.

- **Rendimiento óptimo**: No se puede afectar al rendimiento de la conexión ni tampoco al del sistema por analizar una gran cantidad de paquetes. El análisis debe ser lo más rápido posible.

Observando la figura B, se puede justificar la elección de la implementación de un driver Intermedio, ya que si se realiza un driver TCP, éste puede ser puenteado de la siguiente forma:



Funcionamiento esperado

Funcionamiento puenteado

La técnica de puenteado anteriormente mostrada es una técnica complicada pero factible, sin embargo un Intermediate Driver no puede ser puenteado ya que está en el nivel más bajo posible.

Todas las operaciones de Entrada/Salida de Windows se realizan a través de un componente del núcleo llamado IOManager. Gracias a esto, es posible definir un estándar de comunicación y además evita el tener conocimiento de otros componentes instalados en el sistema: si algún componente necesita ser notificado de algún evento, el IOManager se encarga.

En cuanto a la implementación, normalmente los controladores de Windows se desarrollan en C en lugar de C++, por dos motivos principalmente:

- El núcleo de Windows está escrito en C.
- C tiene una menor sobrecarga en tiempo de ejecución que C++.

Sin embargo, se ha optado por realizar la implementación en C++ por dos motivos:

- Para realizar el filtrado de paquetes de una forma eficiente y elegante, era necesario disponer de las STL

- La estructura de objetos desarrollada puede servir de plataforma para realizar otro tipo de drivers como pueden ser los de balanceo de carga.

- El equipo de desarrollo tiene más experiencia en la programación en C++

Además, la implementación en C++ tiene otras ventajas como son la comprobación de sintaxis más restrictiva y la orientación a objetos.

Microsoft no tiene soporte para los drivers escritos en C++, aunque es de esperar que esto cambie en poco tiempo puesto que este tipo de drivers va aumentando en número. Mientras esto sucede, es necesario utilizar un ***wrapper*** para poder utilizar C++.

El *wrapper* proporciona las siguientes funcionalidades:

- Invocar constructores para las instancias globales de las clases.
- Invocar destructores al salir para las instancias globales.

  Estas dos funcionalidades son necesarias porque el compilador puede situar los constructores y los destructores en zonas del código que en el momento de la ejecución no estén disponibles. Por ejemplo, los destructores pueden estar generados en la zona INIT del código, que deja de estar disponible una vez está inicializado el driver; si después de inicializar se llamase al destructor, el driver fallaría.

- Redefinir **new** y **delete** para su uso en el kernel.

  En el modo kernel, normalmente se usa memoria no paginada, por ello se redefine el constructor *new* para que reserve la memoria en un espacio de este tipo como comportamiento por defecto. Sin embargo, en algunos casos puede ser que el desarrollador quiera usar memoria paginada, por esto es necesario incluir una definición de *new* que permita reservar memoria paginada.

Con estas funcionalidades básicas, ya es posible el uso de C++ para crear un driver, con todo lo que conlleva, aunque en este caso lo más importante era el poder usar las STL de C++.

3. Descripción del funcionamiento

Windows dispone de una pila de controladores por la que van circulando los paquetes desde la red a la aplicación y viceversa. Es el propio Windows el que se encarga de poner el driver en la parte que le corresponda de la pila.

Cuando la tarjeta de red recibe un paquete, lo notifica mediante una interrupción al sistema operativo, éste a través del IOManager (Mediador del sistema operativo Windows que se encarga de todas las operaciones de Entrada/Salida) va enviando el paquete a los drivers por orden. Cada driver hace las operaciones necesarias con el paquete y reenvía el paquete al IOManager para que siga realizando sus operaciones.

En el caso del firewall pueden darse 2 opciones:

- El paquete cumple las restricciones impuestas por el firewall:
  Si, según las reglas impuestas, el paquete no es bloqueado debe seguir su curso, sin sufrir ninguna modificación.

- El paquete NO cumple las restricciones impuestas por el firewall:
  En el caso de que alguna regla impida el paso del paquete, éste será desechado. Si no existiese una comunicación entre el driver y una aplicación que mostrase las acciones del driver, el usuario nunca tendría conocimiento de que ese paquete ha llegado al sistema.

El filtrado de paquetes, implica el almacenamiento de una serie de reglas. Dicho almacenamiento debe cumplir 3 condiciones:

- **Ocupar poca memoria**: En el núcleo de Windows no se puede trabajar con memoria paginada y hay un límite de memoria que los drivers deben respetar.

- **Acceso rápido**: Para saber si un paquete puede pasar o no, el firewall no puede ralentizar ni la red ni el sistema.

- **Flexibilidad**: Es necesario poder tener distintos tipos de reglas con la mayor capacidad de configuración posible.

Para cumplir estas 3 condiciones se optó por usar tablas hash para almacenar las reglas de la siguiente forma:

Como el filtrado se realiza por 4 protocolos distintos, cada protocolo tendrá sus tablas de reglas.

Como se puede elegir entre bloquear los paquetes entrantes y los salientes habrá tablas distintas para cada uno.

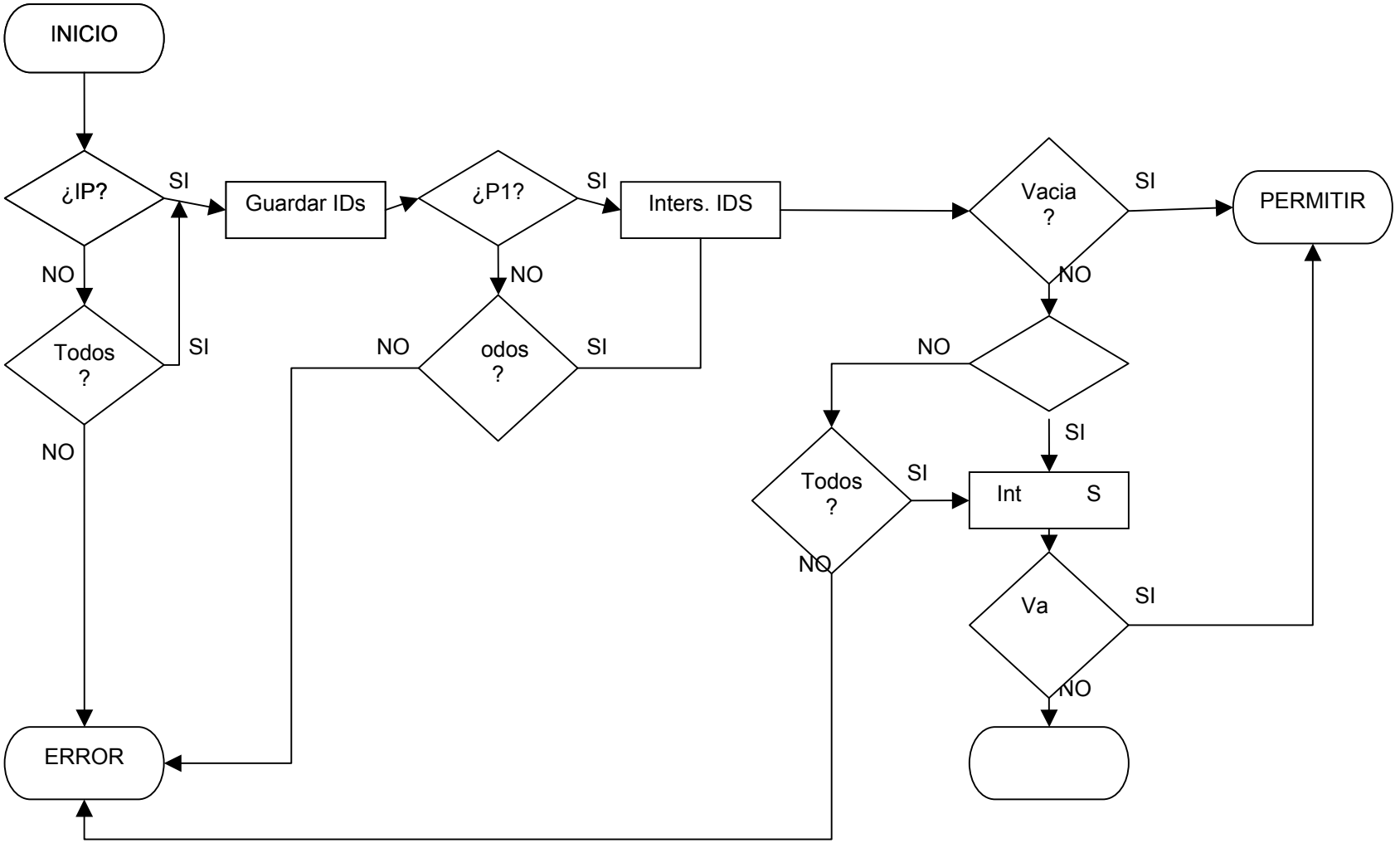
Según el protocolo de las reglas, las tablas son:

- **Protocolo IP**:
  - Dirección IP entrante: Tabla indexada por la dirección IP y que almacena un identificador de la regla.
  - Dirección IP saliente: Tabla indexada por la dirección IP y que almacena un identificador de la regla.
  - Tipo IP entrante: Tabla indexada por el tipo de protocolo IP y que almacena un identificador de la regla.
  - Tipo IP saliente: Tabla indexada por el tipo de protocolo IP y que almacena un identificador de la regla.

- **Protocolos TCP/UDP**:
  - Dirección IP entrante: Tabla indexada por la dirección IP y que almacena un identificador de la regla.
  - Dirección IP saliente: Tabla indexada por la dirección IP y que almacena un identificador de la regla.
  - Puerto Local entrante: Tabla indexada por el puerto local y que almacena un identificador de la regla.
  - Puerto Local saliente: Tabla indexada por el puerto local y que almacena un identificador de la regla.
  - Puerto Remoto entrante: Tabla indexada por el puerto remoto y que almacena un identificador de la regla.
  - Puerto Remoto saliente: Tabla indexada por el puerto remoto y que almacena un identificador de la regla.

- **Protocolo ICMP**:
  - Dirección IP entrante: Tabla indexada por la dirección IP y que almacena un identificador de la regla.
  - Dirección IP saliente: Tabla indexada por la dirección IP y que almacena un identificador de la regla.
  - Tipo ICMP entrante: Tabla indexada por el tipo petición ICMP y que almacena un identificador de la regla.
  - Tipo ICMP saliente: Tabla indexada por el tipo petición ICMP y que almacena un identificador de la regla.

Una vez enumeradas las tablas, se puede explicar el algoritmo de filtrado. Este es un diagrama de flujo para filtrar un paquete TCP o UDP, los otros son similares, pero con menos comprobaciones ya que no se comprueban puertos.


INICIO
¿IP?
SI
Guardar IDs
¿P1?
SI
Inters. IDS
Vacia ?
SI
PERMITIR
NO
NO
NO
Todos ?
SI
NO
odos ?
SI
NO
SI
Todos ?
SI
Int S
NO
Va
SI
NO
ERROR

1. Se comprueba si la dirección IP del paquete está en la tabla correspondiente.

   1.1. Si está, se guardan los identificadores de las reglas que filtran esa IP.
   1.2. Si no está, se guardan los identificadores de las reglas que filtran todas las IP.

2. Se comprueba si el puerto local del paquete está en la tabla correspondiente.

   2.1. Si está, se realiza la intersección de los identificadores de reglas que filtran ese puerto con los que había guardados.
   2.2. Si no está, se realiza la intersección de los identificadores de reglas que filtran todos los puertos con los que había guardados.

3. Se comprueba la intersección.

   3.1. Si la intersección es vacía quiere decir que no hay ninguna regla que impida el paso de ese paquete. **FIN**.
   3.2. Si la intersección no es vacía quiere decir que es posible que exista una regla que impida el paso de ese paquete y hay que seguir comprobando con el puerto remoto.

4. Se comprueba si el puerto remoto del paquete está en la tabla correspondiente.

   4.1. Si está, se realiza la intersección de los identificadores de reglas que filtran ese puerto con los que había guardados.
   4.2. Si no está, se realiza la intersección de los identificadores de reglas que filtran todos los puertos con los que había guardados.

5. Se comprueba la intersección.

   5.1. Si la intersección es vacía quiere decir que no hay ninguna regla que impida el paso de ese paquete. **FIN.**
   5.2. Si la intersección no es vacía quiere decir que existe una regla que impide el paso de ese paquete. **FIN**.

   Con este algoritmo se consiguen los objetivos propuestos:

   - Eficiencia: la consulta a las tablas hash se realiza en tiempo constante, y dentro de cada consulta, se realiza un bucle para encontrar todos los identificadores de reglas (lineal en el número de reglas almacenadas para esa entrada de la tabla)
   - Poco espacio: las tablas solo almacenan enteros, por lo tanto no pueden ocupar demasiado espacio en memoria. Además, al usar las STL, se consigue que ocupen el mínimo espacio posible, redimensionándose las tablas en tiempo de ejecución en caso de ser necesario.
   - Flexibilidad: con las tablas definidas se puede definir cualquier regla para filtrar por los campos más comunes de los paquetes, pero en el caso de que se quiera incluir algún otro campo, basta con incluir las tablas pertinentes.

4. Comunicación

Puesto que el driver es un desarrollo separado de la aplicación la cual permite su configuración, es necesario establecer algún canal de comunicación entre ellos.

Windows dispone de un método para comunicarse con los drivers llamado **IRPs** (I/O Request Packet). Los IRPs son peticiones de Entrada/Salida que se realizan al driver de un dispositivo. Estas peticiones puede que impliquen el uso del dispositivo o puede que el driver pueda completarlas sin necesidad de ocupar el dispositivo. Gracias a esta característica, es posible la comunicación con el driver.

Desde la aplicación no se crean directamente los IRPs sino que se envían unas estructuras llamadas **IOCTLs** (I/O Control) junto con un buffer de entrada con los datos que se quieren enviar al dispositivo y un buffer de salida donde puede escribir el dispositivo. Estos datos los recibe el IOManager, que es quien crea y maneja los IRPs.

Cuando el driver recibe un IRP, comprueba el código del IOCTL que lleva dentro y realiza las operaciones correspondientes.

En el caso de RoccoFirewall las posibles comunicaciones entre la aplicación y el driver son:

- **IOCTL_PTUSERIO_OPEN_ADAPTER**: Es necesaria para comprobar que el driver está disponible y poder establecer cualquier otra comunicación.
- **IOCTL_PTUSERIO_SEND_RULES**: Envía las reglas de filtrado al driver. Este borrará las antiguas y añadirá las nuevas.
- **IOCTL_PTUSERIO_SEND_TYPE**: Configura el modo de funcionamiento del driver (Normal, Permitir todo, Bloquear todo).
- **IOCTL_PTUSERIO_LOGGING_STATUS**: Configura si el driver debe guardar un informe con los paquetes que va bloqueando o no.
- **IOCTL_PTUSERIO_SEND_LOG**: Solicita al driver el informe con los paquetes que haya bloqueado.

Las peticiones en RoccoFirewall se realizan a distintos tiempos:

- El envío de las reglas se realiza al abrir la aplicación y al modificar, añadir o borrar alguna regla.
- El envío del modo de funcionamiento se realiza al iniciar la aplicación (entra en el modo de funcionamiento normal), al cambiar el modo de funcionamiento mediante el menú y al cerrar la aplicación (indica al driver que debe dejar pasar todos los paquetes)
- El envío del estado del logging se realiza al iniciar la aplicación (se activa) y al cambiar la opción en el menú.

- La petición del log de bloqueados se realiza periódicamente (cada segundo). En un principio se intentó que la aplicación no tuviese que pedir al driver el log, sino que fuese éste quien se lo mandase asíncronamente, pero esta tentativa fracasó, ya que las formas que se encontraron (Eventos, Llamadas asíncronas a procedimientos…) o bien no estaban suficientemente documentadas como para saber usarlas o no cumplían con la funcionalidad necesaria. En cualquier caso, y aunque pueda parecer lo contrario, estas peticiones no producen apenas sobrecarga.

- La petición para abrir el adaptador se realiza justo antes de cualquier otra llamada, para comprobar que está disponible.

Para cualquier otra información acerca del funcionamiento de la aplicación o cualquier otro dato sobre el desarrollo, se puede consultar la página web

http://roccosoft.sytes.net

MANUAL DE USUARIO

1. Instalación.

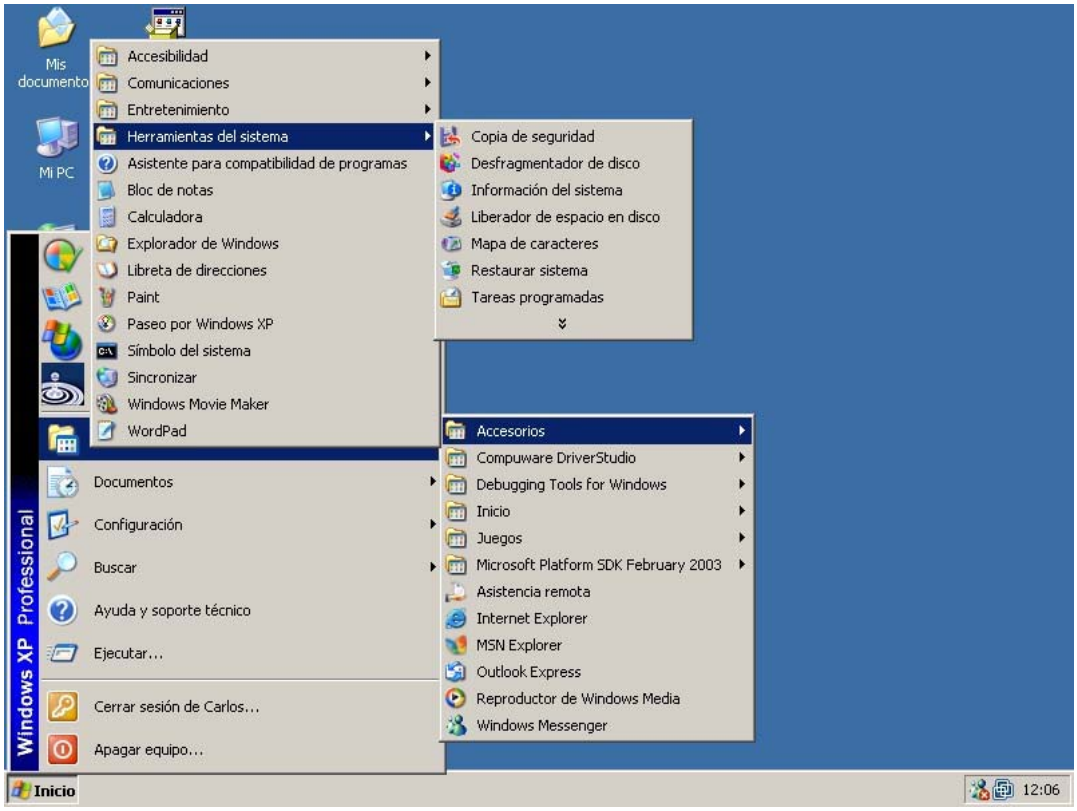
Establecer un punto de restauración del sistema (Sólo Windows XP).

Esto no es estrictamente necesario, pero puesto que RoccoFirewall está en sus primeras versiones y no ha podido ser comprobado en todas las configuraciones posibles, siempre es recomendable realizarlo.

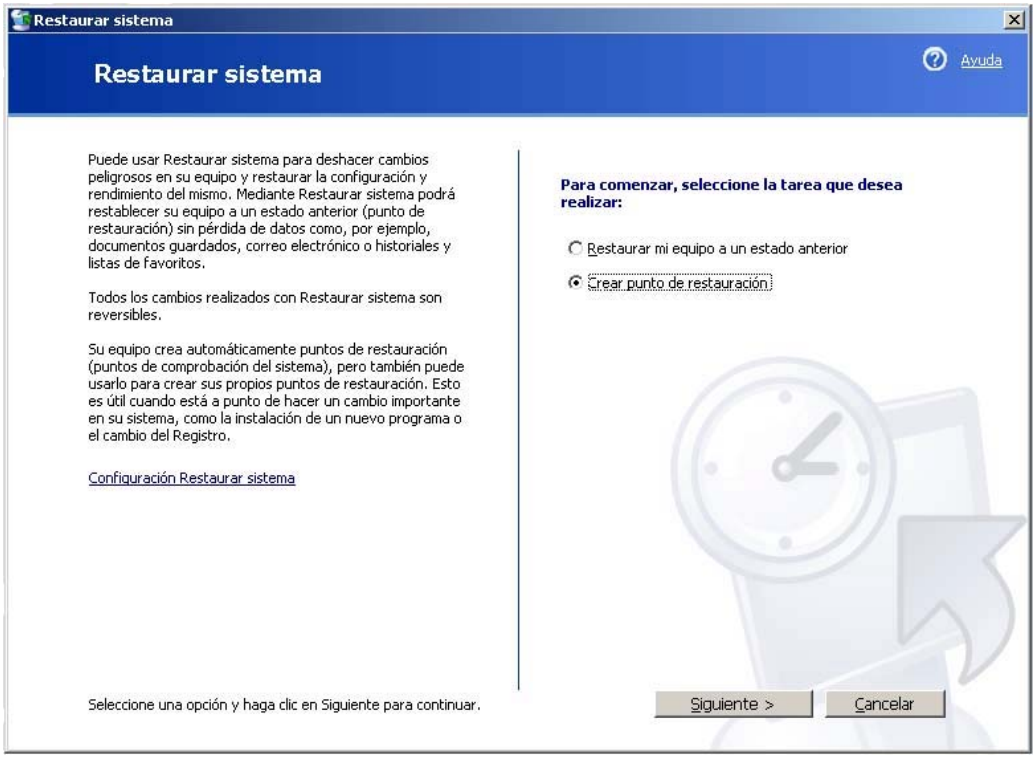
A continuación se muestra un resumen de como se debe realizar:

- Iniciar la utilidad de restauración del sistema. Para ello pulsar:

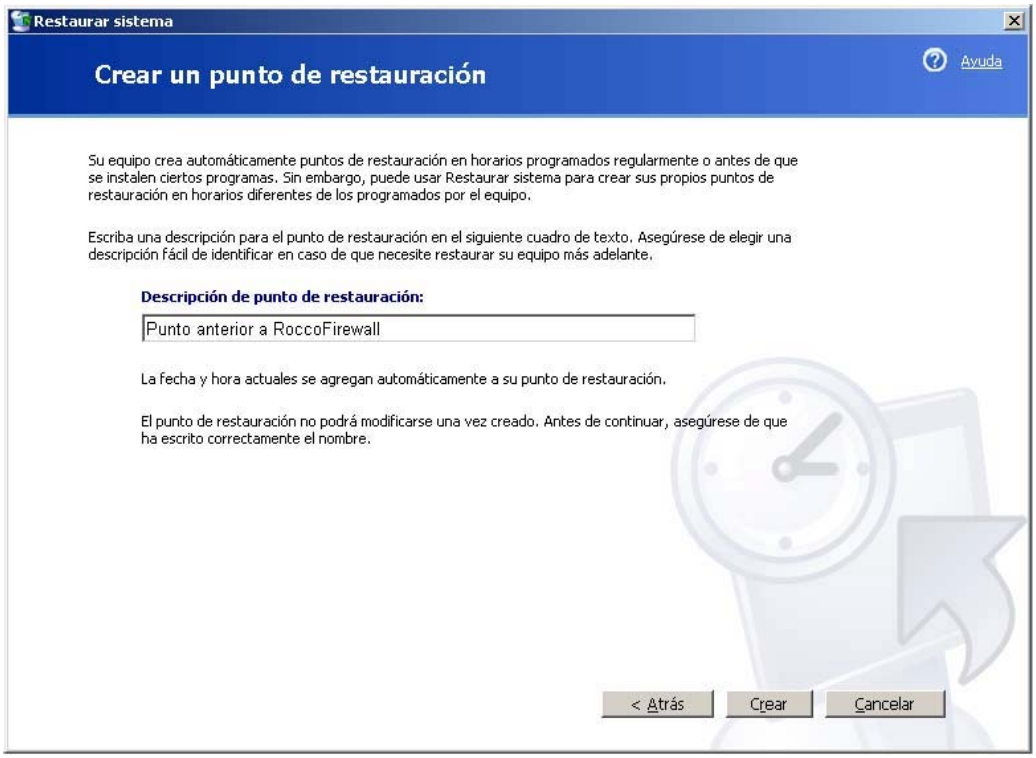
Inicio → Programas → Accesorios → Herr. del sistema → Restaurar Sistema



- Seleccionar la opción "Crear punto de restauración" y pulsar "Siguiente".



- Introducir una descripción para el punto y hacer clic en "Crear".



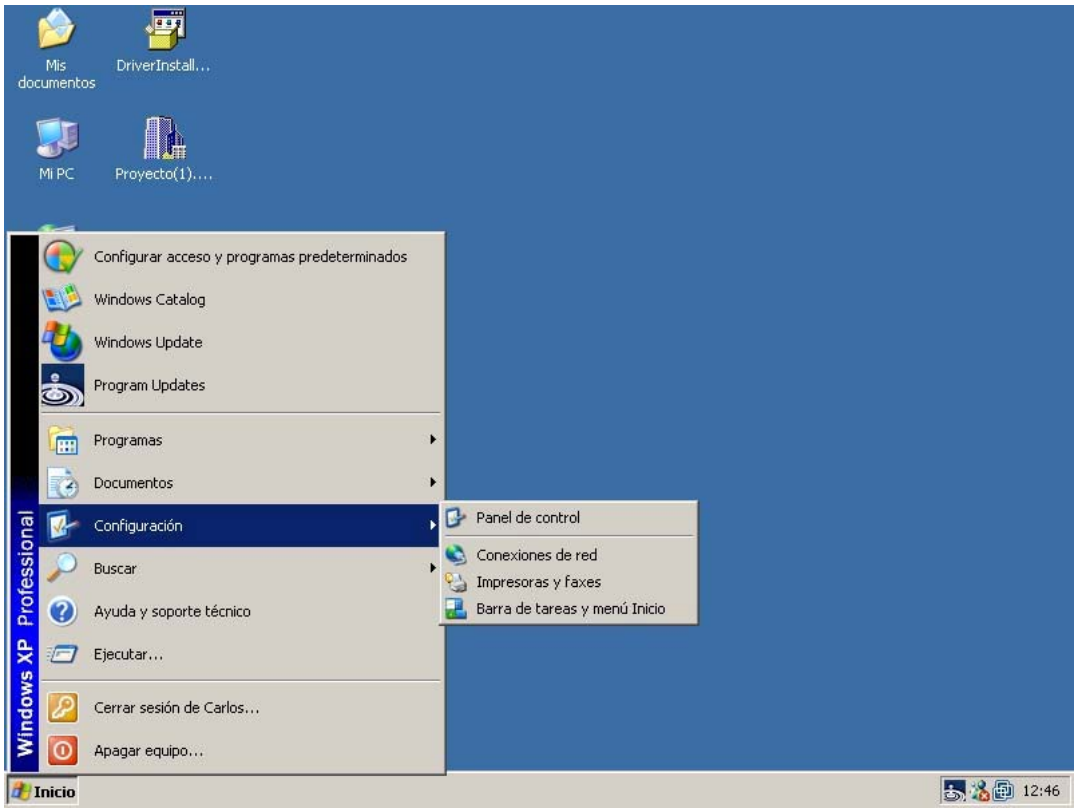
Con esto estará creado el Punto de Restauración. Así en el caso de que exista algún problema de incompatibilidad con el driver, se podrá volver al estado anterior.

Instalación del Servicio de Red

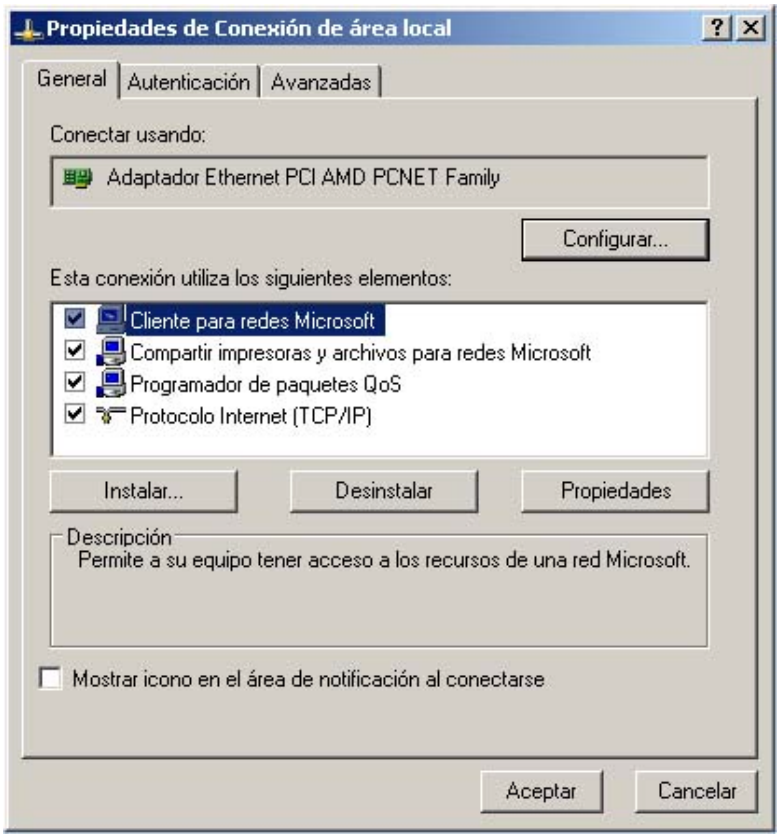
Esto se puede considerar como el "núcleo" de RoccoFirewall. Sin realizar esto es imposible que el firewall funcione.

- Abrir las "Conexiones de Red".

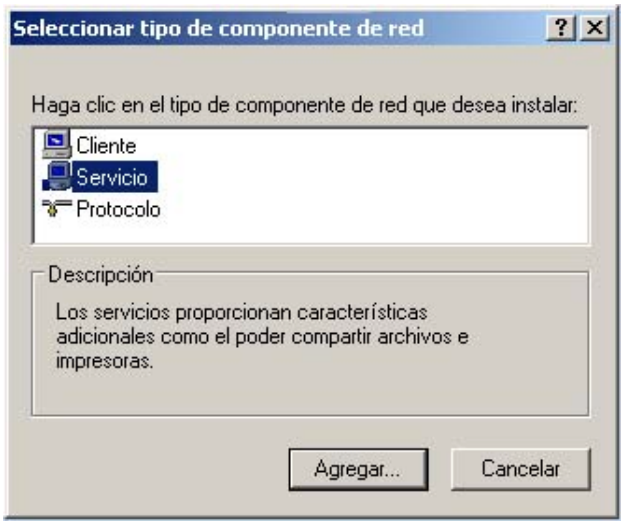
Inicio → Configuración → Conexiones de red



- En la ventana que se abre, pulsar con el botón derecho en la conexión de la tarjeta de red sobre la que se quiera aplicar el firewall, y a continuación seleccionar "Propiedades".

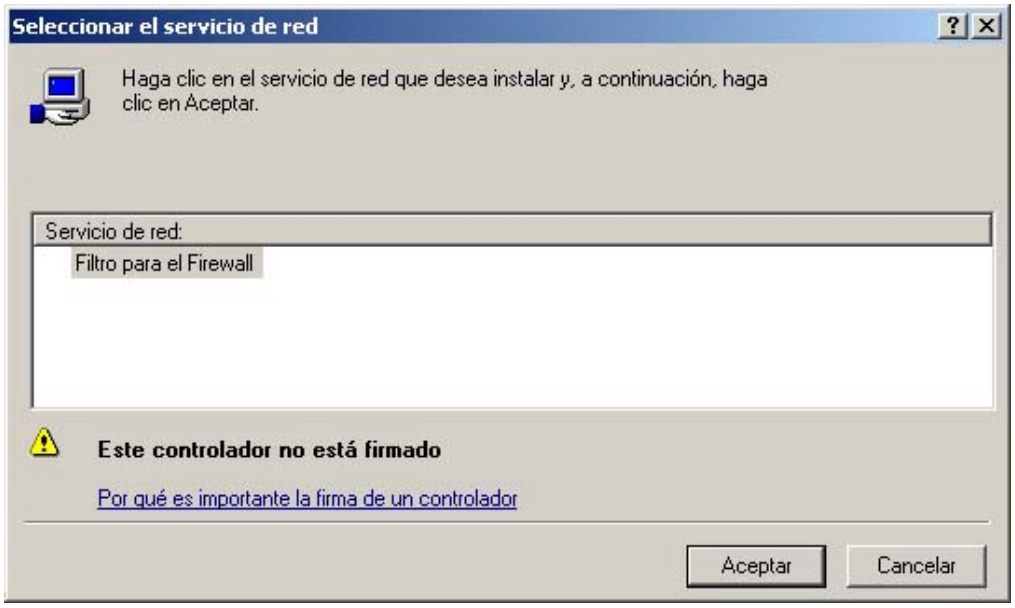


- Pulsar el botón "Instalar", seleccionar "Servicio" y pulsar "Agregar"



- Cuando aparezca el diálogo pidiendo el servicio de red, pulsar sobre el botón "Utilizar disco", seleccionar la ruta donde están los ficheros de RoccoFirewall y pulsar "Aceptar".

- Aparecerá otra ventana para que seleccionar el Servicio. Esta vez seleccionar "Filtro para el Firewall" y pulsar "Aceptar"



- A continuación, Windows instalará el controlador, dando al menos una vez el siguiente aviso:

Pulsar "Continuar" tantas veces como aparezca el mensaje.

Siguiendo estos pasos, se habrá completado la instalación del controlador.

Instalación de la aplicación

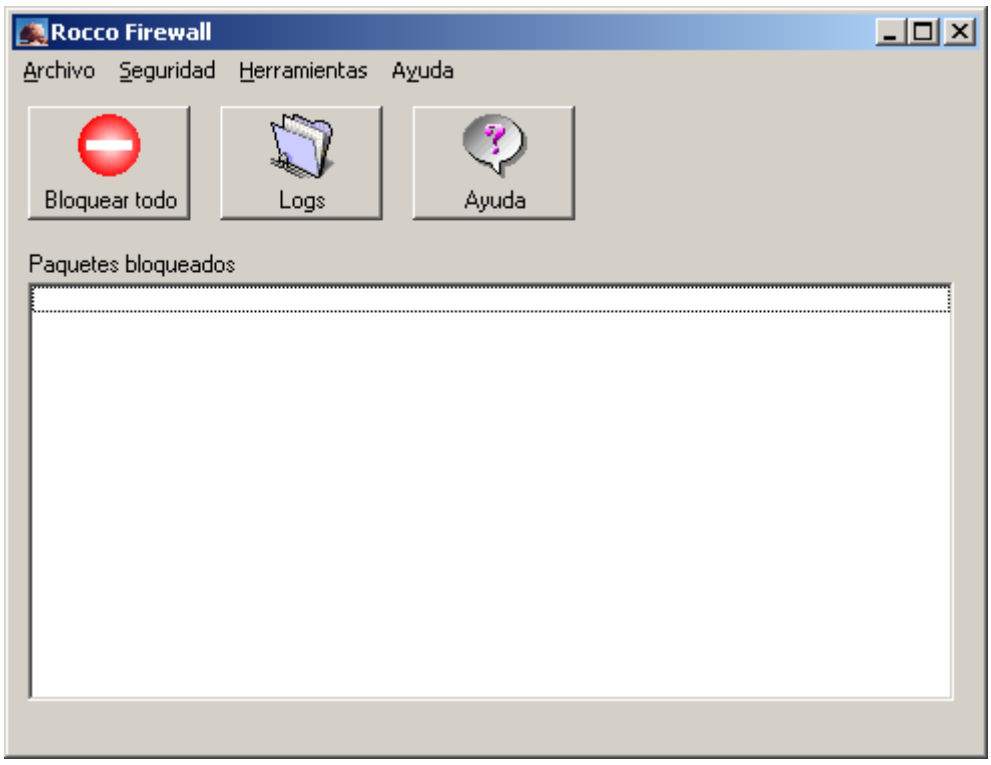
El ejecutable que sirve para configurar RoccoFirewall no necesita ninguna instalación adicional. Simplemente hay que copiarlo en una carpeta y ejecutarlo.

2. Uso de RoccoFirewall

A continuación se muestra la pantalla principal de la aplicación. Se puede observar que hay un menú con varias opciones disponibles (Archivo, Seguridad, Herramientas y Ayuda)

Bajo el menú se ubican unos botones que permiten acceder a algunas de las funcionalidades del firewall de manera más rápida.

Se examinarán todas con detenimiento.



Las opciones disponibles en el menú son las siguientes:

1. *Archivo*: Se muestra una opción *Salir* que sale de la aplicación. De esta forma se desactiva el firewall.

2. *Seguridad*: Esta opción permite configurar el modo de funcionamiento del firewall. Tiene tres modos posibles:

   - *Bloquear todo*: bloquea todo el tráfico de red, tanto entrante como saliente.

   - *Normal*: es el modo de funcionamiento activo por defecto. Cuando está activo, el comportamiento del firewall se rige por las reglas especificadas en el apartado de reglas avanzadas (este apartado se tratará más adelante en el manual).

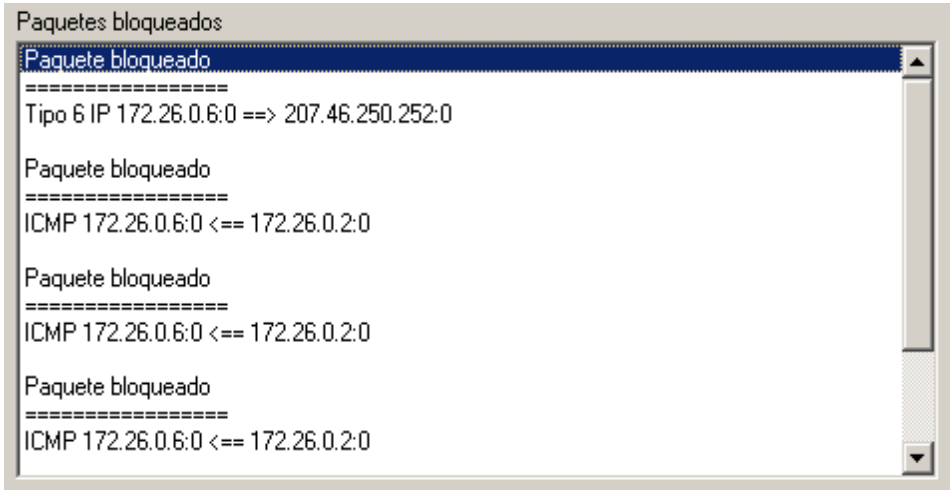
   - *Permitir todo*: este modo permite todo el tráfico de red, ignorando las reglas especificadas.

3. *Herramientas*: Tiene dos opciones:

   - *Logs*: Para gestionar los logs de la aplicación. Al pulsar esta opción se despliega un submenú mediante el cual se puede:

     - *Activar logs:* Para activar el guardado de los logs en un fichero.
     - *Desactivar logs*: Para dejar de guardar los logs en el fichero.
     - *Ver logs*: Para ver todos los logs almacenados hasta el momento.

   Si están activados los logs, en el cuadro central de la aplicación, aparecerá la lista con los paquetes bloqueados de la siguiente forma:



   La información que se muestra es:
   Regla por la que se ha bloqueado: IP, TCP, UDP o ICMP
   Dirección y puerto locales.
   Sentido del envío.
   Dirección y puerto remotos.

   Además, si se ha bloqueado por Tipo IP o ICMP aparecerá dicho tipo. (Ver apéndices A y B).

- *Reglas avanzadas*: esta opción permite configurar las reglas de bloqueo de paquetes. Se explicará en la siguiente sección.

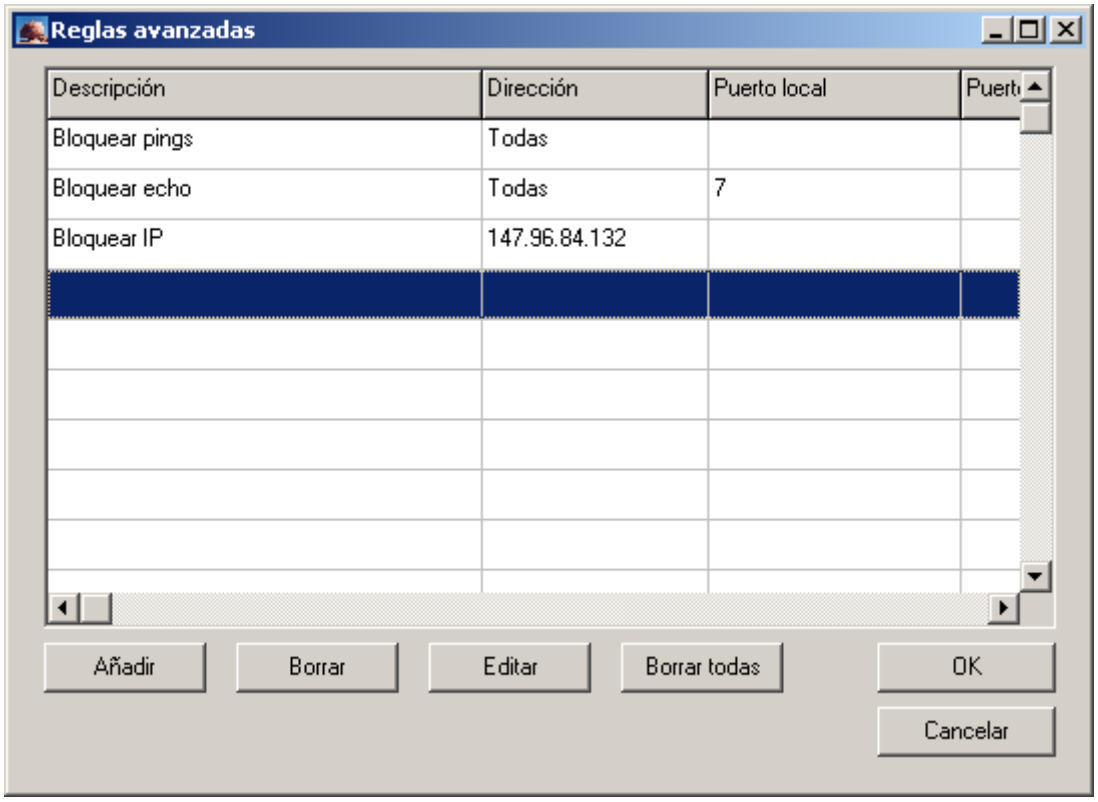
4. *Ayuda*: Muestra una pequeña ayuda para configurar las reglas avanzadas.

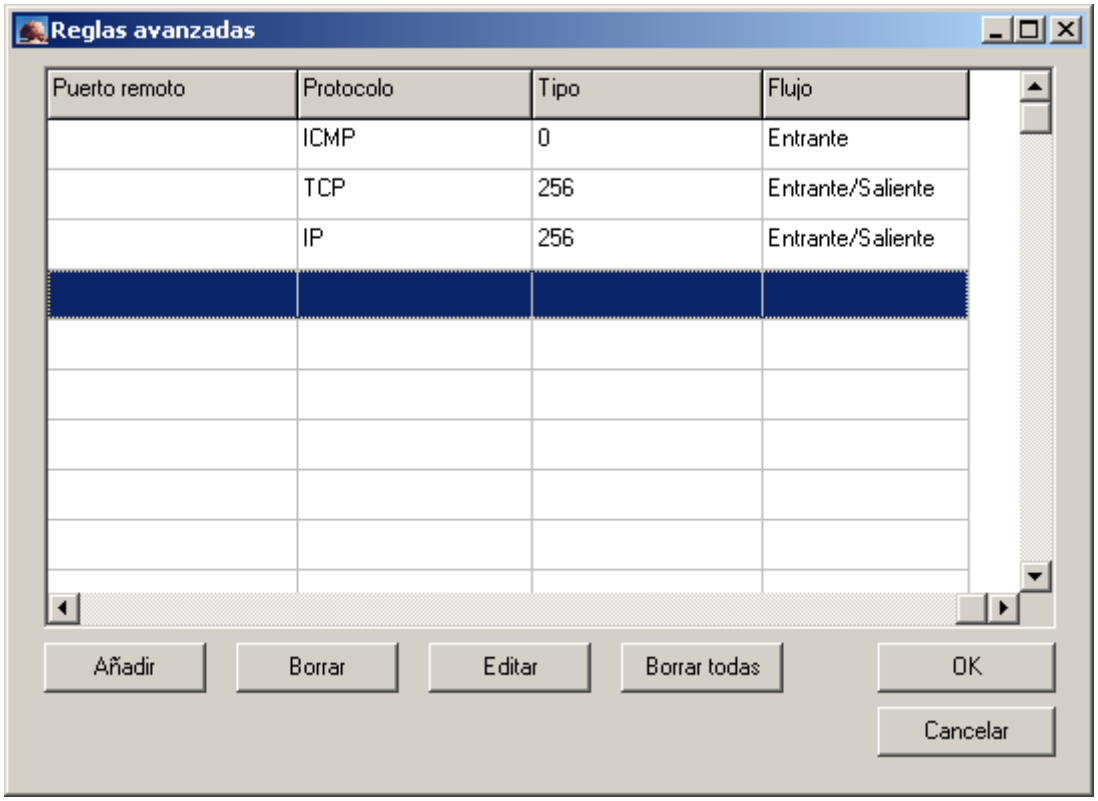
Las otras tres opciones disponibles mediante los botones permiten *bloquear todo*, ver los *logs* u obtener *ayuda* sin necesidad de utilizar el menú.

Gestión de las reglas avanzadas

Mediante el uso de las reglas avanzadas se puede tener un control total sobre el comportamiento del firewall.

Al pulsar en la opción *Herramientas -> Reglas avanzadas* se muestra la siguiente ventana:





Es una tabla en la que se muestran todas las reglas añadidas hasta el momento. Cada campo de la tabla es un campo de la regla correspondiente. Los campos son los siguientes:

- *Descripción*: Una breve descripción de cada regla.
- *Dirección*: La dirección a bloquear. Puede ser una IP concreta o todas las direcciones
- *Puerto local* y *puerto remoto*: Los puertos locales y/o remotos a bloquear (en el caso de protocolos TCP o UDP se pueden especificar determinados puertos para ser bloqueados).
- *Protocolo*: El protocolo de los paquetes a bloquear. Los protocolos disponibles son TCP, IP, UDP e ICMP.
- *Tipo*: El tipo del protocolo seleccionado (en el caso de protocolos IP e ICMP se puede especificar un tipo).
- *Flujo*: El sentido del flujo que se bloquea (paquetes entrantes, salientes o ambos).

Mediante las opciones mostradas en la parte inferior de la ventana se puede:

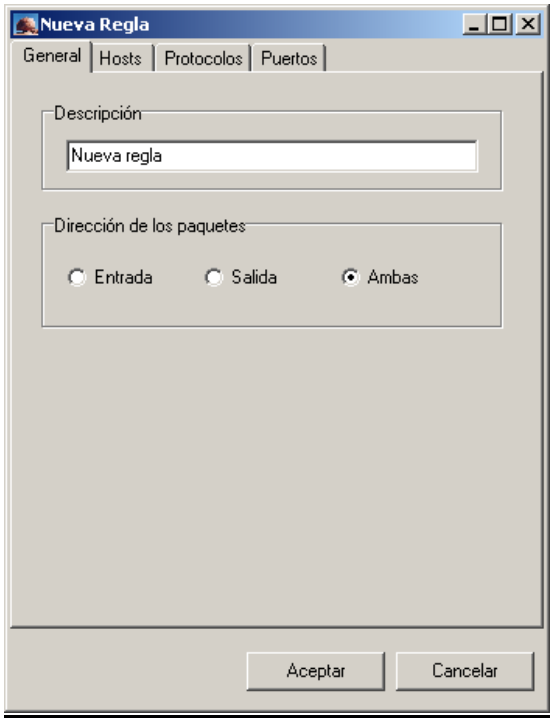
- *Añadir* una regla nueva a la lista.

- *Borrar* la regla seleccionada.
- *Editar* la regla seleccionada para cambiar alguno o todos los campos.
- *Borrar todas* las reglas de la lista. De esta forma, el firewall funcionará en un modo equivalente a *permitir todo* hasta que se especifiquen nuevas reglas.

Para salir de esta ventana, solo hay que pulsar el botón *OK* si se desean guardar los cambios o *Cancelar* si se desean ignorar.

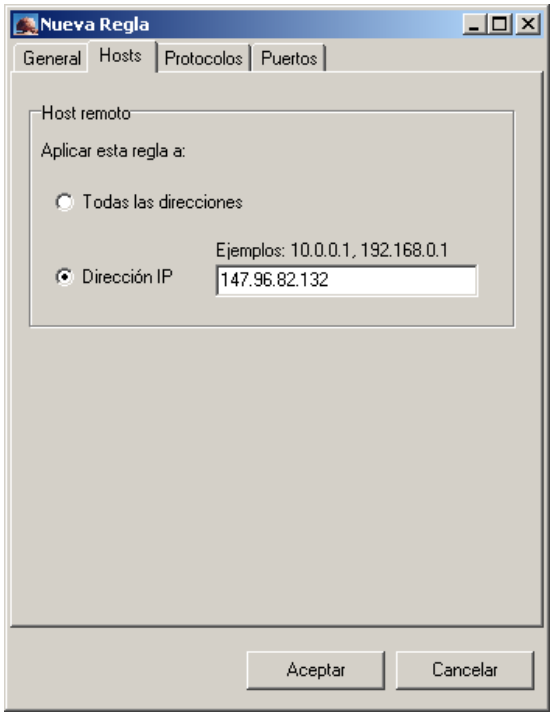
Configuración de una nueva regla

Al pulsar el botón *Añadir* en la ventana de Reglas avanzadas se muestra lo siguiente:

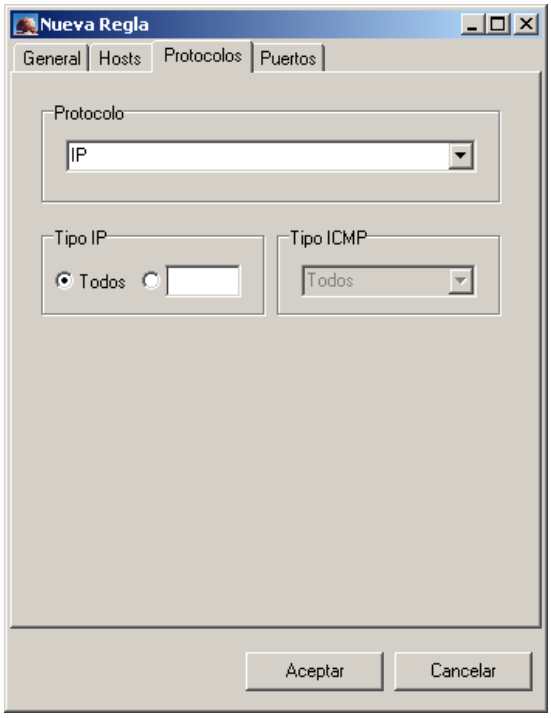


En cada una de las secciones se especifican distintos aspectos de la regla que se va a introducir.

La primera sección (mostrada arriba) es *General*. En ella se solicita una pequeña descripción de la regla y el sentido de los paquetes a bloquear. Se pueden bloquear paquetes de entrada, de salida o de entrada y salida.



En la sección *Host remoto* se puede especificar si se desea bloquear una dirección IP específica o si por el contrario se desean bloquear todas las direcciones. Por defecto se bloquearán todas las direcciones.



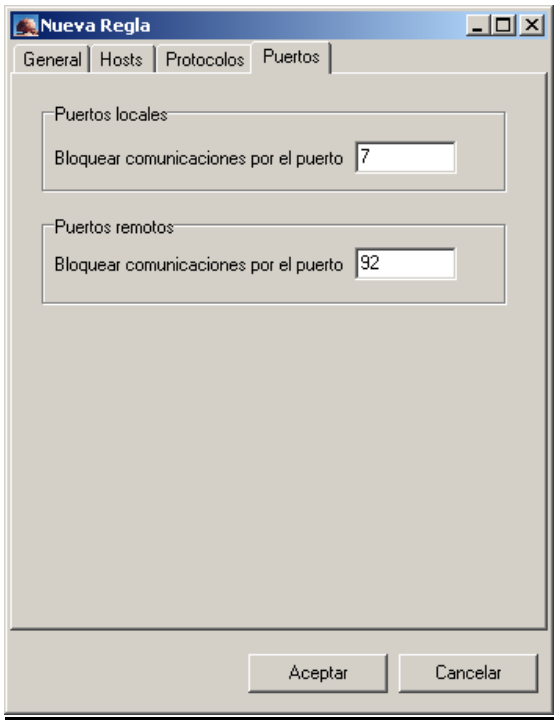
En la sección *Protocolos* se solicita el protocolo de los paquetes a bloquear. Según el protocolo seleccionado hay opciones adicionales:

- En el caso de los protocolos *TCP* y *UDP*, se pueden especificar puertos locales o remotos para filtrarlos. Para especificar los puertos se debe seleccionar primero el protocolo deseado (TCP o UDP) y después acceder a la sección *Puertos* de esta misma ventana (la opción *Puertos* en protocolos IP e ICMP está deshabilitada). En el caso de que no se especifiquen puertos, se entenderá que se quiere impedir la conexión desde/a todos los puertos.

- En el caso del protocolo *IP*, se podrá elegir un tipo a bloquear. Este tipo es un número entre 0 y 255 que especifica el protocolo del paquete. También se pueden bloquear todos los tipos. Al seleccionar el protocolo IP, si no se especifica un número, por defecto se bloquearán todos los tipos IP.

- En el caso del protocolo *ICMP* también se puede especificar un tipo a bloquear. Este tipo es un número entre 0 y 18 que especifica el tipo de petición ICMP. Los posibles tipos ICMP están enumerados en el apéndice A.

Si se elige bloquear todos los protocolos (opción por defecto) se entenderá que de desea bloquear cualquier puerto de los protocolos que disponen de puertos y cualquier tipo de los protocolos que dispongan de tipos.



En la sección *Puertos* se especifican los puertos locales y remotos a bloquear en el caso de que se haya seleccionado protocolo TCP o UDP. En el apéndice B se dan algunos consejos de seguridad relacionados con puertos TCP/UDP.

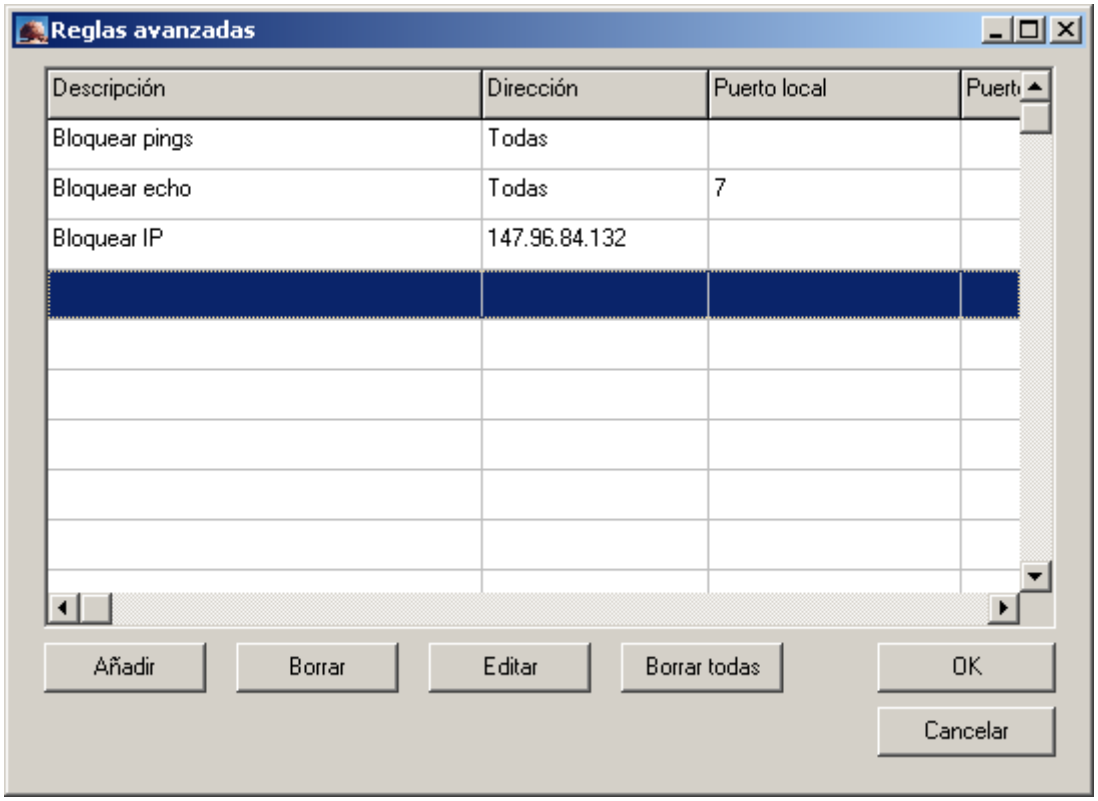
Cuando se haya terminado de configurar la regla, se puede salir de la ventana pulsando el botón *Aceptar* si se quiere añadir la nueva regla, o *Cancelar* si no se desea añadirla.

Ejemplo práctico: bloqueo de chargen

A continuación se explica cómo configurar una regla para bloquear el puerto *chargen*, utilizado frecuentemente para provocar bucles en la red. El puerto correspondiente es el número 19, en TCP y en UDP.

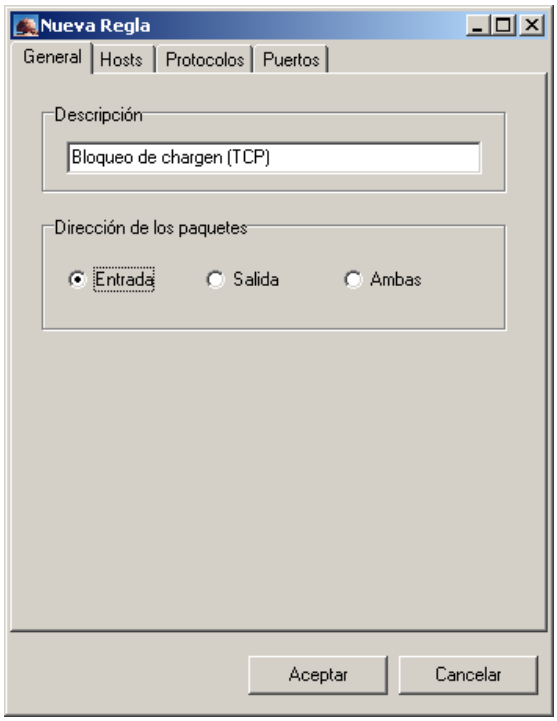
Para bloquearlo se especificarán dos reglas, una para el protocolo TCP y otra para en protocolo UDP.

En primer lugar se introduce la regla correspondiente para TCP. Se hace clic en el menú *Herramientas → Reglas avanzadas.* Una vez hecho esto aparece lo siguiente:



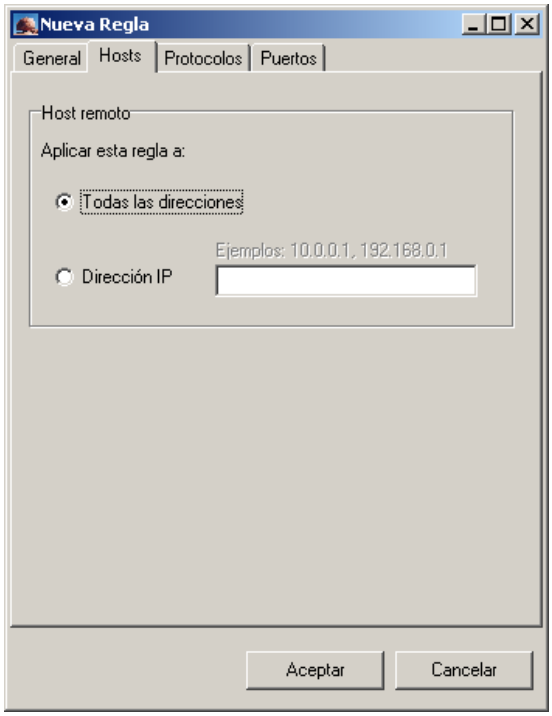
En esta tabla aparecerán las reglas que se hayan definido hasta el momento. Si es la primera vez que se ejecuta la aplicación, la tabla aparecerá vacía.

Para añadir una nueva regla se pulsa el botón *Añadir*. Una vez hecho esto, aparece la siguiente ventana:



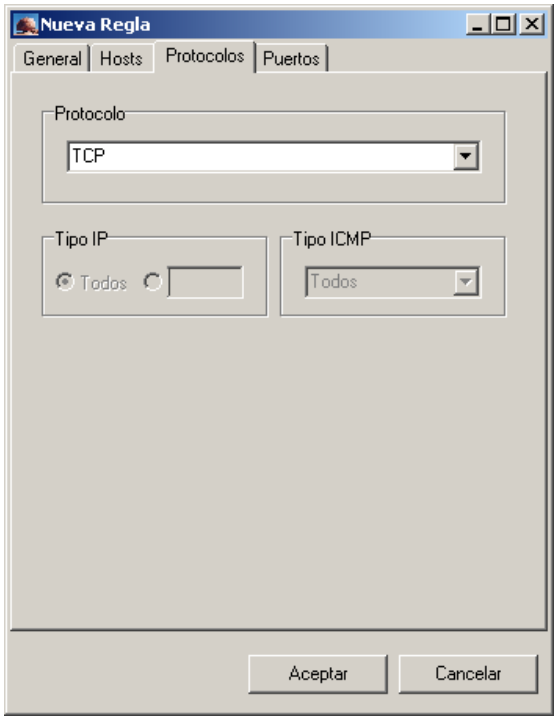
En el campo *Descripción* se introduce una breve descripción de la regla que se vaya a añadir, en este caso *Bloqueo de chargen (TCP).* Como en este caso los paquetes que se desea bloquear son paquetes de entrada, en la sección *Dirección de los paquetes* hay que seleccionar *Entrada.*

A continuación se hace clic en la pestaña *Hosts*:

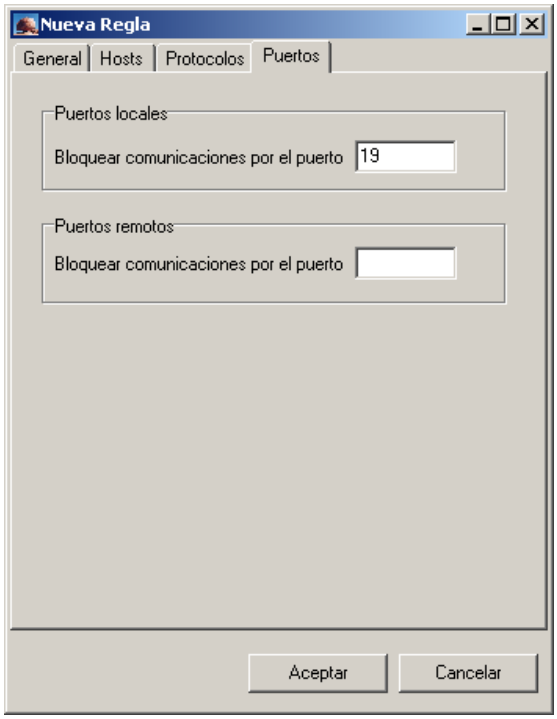


En este caso no interesa bloquear una dirección específica, sino bloquear todas las direcciones (ya que a priori se desconoce la fuente de un posible ataque). Por eso hay que seleccionar *Aplicar la regla a todas las direcciones*.

Una vez hecho esto, se pulsa en la pestaña *Protocolos*:

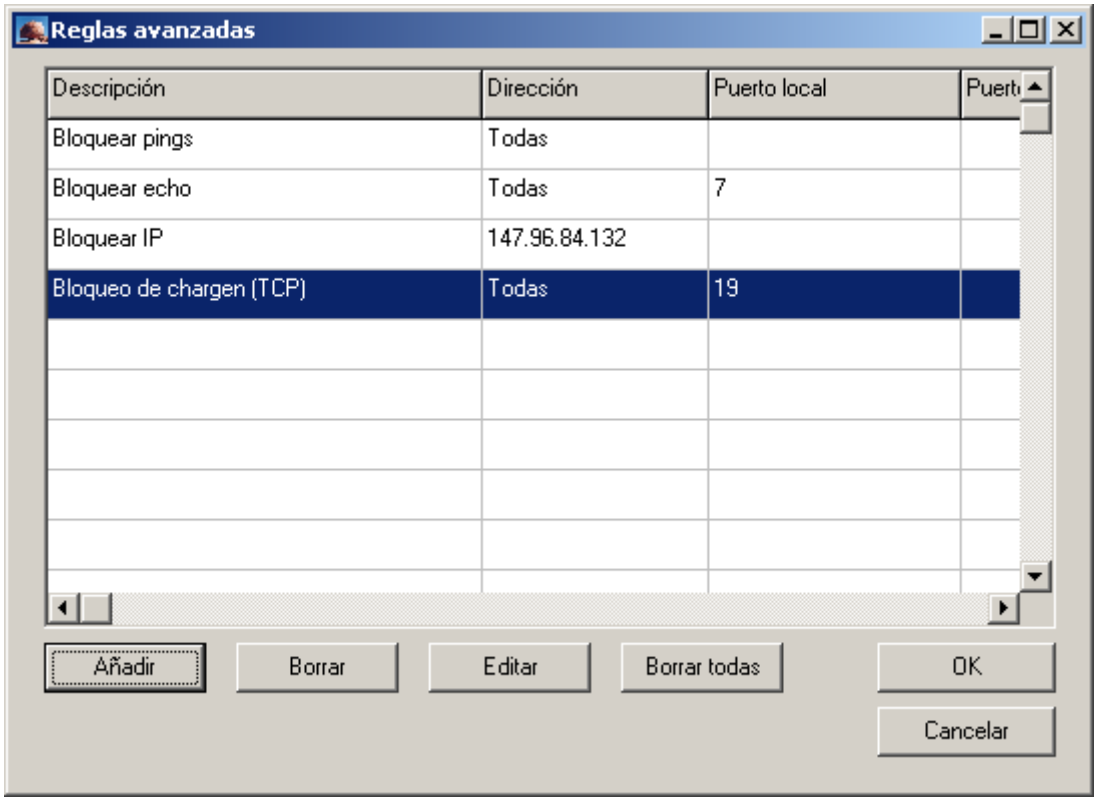


Hay que seleccionar *Protocolo* TCP en la lista desplegable que aparece en la parte superior. Ya que este protocolo permite especificar puertos, el siguiente paso es hacer clic en la pestaña *Puertos*:



En el caso que se está tratando, el puerto a bloquear es el puerto 19 en sentido entrante.

Una vez hecho esto, se pulsa *Aceptar* y el resultado debe ser el siguiente:



El proceso para añadir la regla para el protocolo UDP es análoga, cambiando solo el protocolo TCP por UDP.

Apéndice A: Códigos de los protocolos

A continuación se enumeran los protocolos que puede encapsular un paquete IP:

```
  0      HOPOPT       IPv6 Hop-by-Hop Option
  1      ICMP         Internet Control Message
  2      IGMP         Internet Group Management
  3      GGP          Gateway-to-Gateway
  4      IP           IP in IP (encapsulation)
  5      ST           Stream
  6      TCP          Transmission Control
  7      CBT          CBT
  8      EGP          Exterior Gateway Protocol
  9      IGP          any private interior gateway
                      (used by Cisco for their IGRP)
 10      BBN-RCC-MON  BBN RCC Monitoring
 11      NVP-II       Network Voice Protocol
 12      PUP          PUP
 13      ARGUS        ARGUS
 14      EMCON        EMCON
 15      XNET         Cross Net Debugger
 16      CHAOS        Chaos
 17      UDP          User Datagram
 18      MUX          Multiplexing
```

19	DCN-MEAS	DCN Measurement Subsystems
20	HMP	Host Monitoring
21	PRM	Packet Radio Measurement
22	XNS-IDP	XEROX NS IDP
23	TRUNK-1	Trunk-1
24	TRUNK-2	Trunk-2
25	LEAF-1	Leaf-1
26	LEAF-2	Leaf-2
27	RDP	Reliable Data Protocol
28	IRTP	Internet Reliable Transaction
29	ISO-TP4	ISO Transport Protocol Class 4
30	NETBLT	Bulk Data Transfer Protocol
31	MFE-NSP	MFE Network Services Protocol
32	MERIT-INP	MERIT Internodal Protocol
33	SEP	Sequential Exchange Protocol
34	3PC	Third Party Connect Protocol
35	IDPR	Inter-Domain Policy Routing Protocol
36	XTP	XTP
37	DDP	Datagram Delivery Protocol
38	IDPR-CMTP	IDPR Control Message Transport Proto
39	TP++	TP++ Transport Protocol
40	IL	IL Transport Protocol
41	IPv6	Ipv6
42	SDRP	Source Demand Routing Protocol
43	IPv6-Route	Routing Header for IPv6
44	IPv6-Frag	Fragment Header for IPv6
45	IDRP	Inter-Domain Routing Protocol
46	RSVP	Reservation Protocol
47	GRE	General Routing Encapsulation
48	MHRP	Mobile Host Routing Protocol
49	BNA	BNA
50	ESP	Encap Security Payload for IPv6
51	AH	Authentication Header for IPv6
52	I-NLSP	Integrated Net Layer Security TUBA
53	SWIPE	IP with Encryption
54	NARP	NBMA Address Resolution Protocol
55	MOBILE	IP Mobility
56	TLSP	Transport Layer Security Protocol using Kryptonet key management
57	SKIP	SKIP
58	IPv6-ICMP	ICMP for IPv6
59	IPv6-NoNxt	No Next Header for IPv6
60	IPv6-Opts	Destination Options for IPv6
61		any host internal protocol
62	CFTP	CFTP
63		any local network
64	SAT-EXPAK	SATNET and Backroom EXPAK
65	KRYPTOLAN	Kryptolan
66	RVD	MIT Remote Virtual Disk Protocol
67	IPPC	Internet Pluribus Packet Core
68		any distributed file system
69	SAT-MON	SATNET Monitoring
70	VISA	VISA Protocol
71	IPCV	Internet Packet Core Utility
72	CPNX	Computer Protocol Network Executive
73	CPHB	Computer Protocol Heart Beat

74	WSN	Wang Span Network
75	PVP	Packet Video Protocol
76	BR-SAT-MON	Backroom SATNET Monitoring
77	SUN-ND	SUN ND PROTOCOL-Temporary
78	WB-MON	WIDEBAND Monitoring
79	WB-EXPAK	WIDEBAND EXPAK
80	ISO-IP	ISO Internet Protocol
81	VMTP	VMTP
82	SECURE-VMTP	SECURE-VMTP
83	VINES	VINES
84	TTP	TTP
85	NSFNET-IGP	NSFNET-IGP
86	DGP	Dissimilar Gateway Protocol
87	TCF	TCF
88	EIGRP	EIGRP
89	OSPFIGP	OSPFIGP
90	Sprite-RPC	Sprite RPC Protocol
91	LARP	Locus Address Resolution Protocol
92	MTP	Multicast Transport Protocol
93	AX.25	AX.25 Frames
94	IPIP	IP-within-IP Encapsulation Protocol
95	MICP	Mobile Internetworking Control Pro.
96	SCC-SP	Semaphore Communications Sec. Pro.
97	ETHERIP	Ethernet-within-IP Encapsulation
98	ENCAP	Encapsulation Header
99		any private encryption scheme
100	GMTP	GMTP
101	IFMP	Ipsilon Flow Management Protocol
102	PNNI	PNNI over IP
103	PIM	Protocol Independent Multicast
104	ARIS	ARIS
105	SCPS	SCPS
106	QNX	QNX
107	A/N	Active Networks
108	IPComp	IP Payload Compression Protocol
109	SNP	Sitara Networks Protocol
110	Compaq-Peer	Compaq Peer Protocol
111	IPX-in-IP	IPX in IP
112	VRRP	Virtual Router Redundancy Protocol
113	PGM	PGM Reliable Transport Protocol
114		any 0-hop protocol
115	L2TP	Layer Two Tunneling Protocol
116	DDX	D-II Data Exchange (DDX)
117	IATP	Interactive Agent Transfer Protocol
118	STP	Schedule Transfer Protocol
119	SRP	SpectraLink Radio Protocol
120	UTI	UTI
121	SMP	Simple Message Protocol
122	SM	SM
123	PTP	Performance Transparency Protocol
124	ISIS over IPv4	
125	FIRE	
126	CRTP	Combat Radio Transport Protocol
127	CRUDP	Combat Radio User Datagram
128	SSCOPMCE	
129	IPLT	

```
130     SPS          Secure Packet Shield
131     PIPE         Private IP Encapsulation within IP
132     SCTP         Stream Control Transmission Protocol
133     FC           Fibre Channel
134     RSVP-E2E-IGNORE
135     Mobility Header
136     UDPLite
137-252 Unassigned
253     Use for experimentation and testing
254     Use for experimentation and testing
255                  Reserved
```

Apéndice B: Consejos de seguridad sobre ICMP

0 → Echo replay

El uso del ICMP más familiar es el ping. Mediante el ping se envía un ICMP echo request (tipo 8) y se espera un echo replay (tipo 0). El ping puede ser utilizado para escaneo de puertos, por esta razón muchos firewalls bloquean echo requests entrantes.

3 → Destination unreachable

Es importante permitir que estos mensajes pasen a través del firewall cuando sean entrantes. Si se bloquean los navegadores no reciben el mensaje destination unreachable y esperarán hasta que se produzca timeout. Permitir estos paquetes cuando son salientes es otra cuestión. Mediante estos mensajes se puede hacer un mapeo “inverso” de la red.

4 → Packet lost, slow down

El tipo 4 se puede usar para denegación de servicio. Este mensaje se manda al emisor del tráfico para decirle que mande los paquetes al receptor más despacio. Es conveniente permitir estos mensajes en servidores públicos, pero no tiene sentido permitirlos en redes internas.

5 → Shorter route

El tipo 5 se usa para ajustar tablas de enrutamiento. Dependiendo de la información disponible del router puede mandar uno de los cuatro códigos, por ejemplo un host redirect (código 0) o un network redirect (código 1). Si un atacante puede enviar un ICMP redirects, puede ajustar la tabla de enrutamiento, provocando una denegación de servicio. Este tipo nunca debería pasar el firewall.

8 → Echo service

9 → Router advertisement

10 → Router solicitation

11 → Time exceeded

Este tipo se puede usar para mapear redes. También se puede usar para mapeo de puertos, aunque esta técnica depende de los puertos accesibles a través del firewall.

12 → IP header bad

13 → Timestamp request

14 → Timestamp reply

El timestamp (tipo 13) se usa para escanear puertos, pero solo en sistemas Unix. Windows 2000 y XP responden a peticiones timestamp. Por lo tanto, mediante peticiones ICMP se pueden identificar tanto a las pilas IP de Windows como a las de Unix.

15 → Information request

16 → Information reply

El information reply debería ser respondido solo por los routers, con un address mask reply (tipo 16). El address mask request no solo se usa para identificar routers, sino también para obtener información de subred, muy útil para mapear redes. Es un buen candidato para bloquear por medio de un firewall.

17 → Address mask request

18 → Address mask reply

Apéndice C: Consejos de seguridad sobre TCP/UDP

echo (7 TCP, UDP)
Se utiliza únicamente para depuración. Sin embargo un atacante puede realizar "labores de depuración" creando bucles en la red a partir de este puerto. BLOQUEAR.

systat (11/TCP/UDP)
Muestra información acerca del host, como usuarios conectados, carga del sistema… BLOQUEAR.

chargen (19/TCP, UDP)
Se utiliza únicamente para depuración. Basta con enviar un paquete a este puerto aparentemente originado desde el puerto de echo para provocar un bucle en la red. BLOQUEAR.

telnet (23/TCP, UDP)
Vulnerable a "toma de sesiones". Es preferible utilizar otras soluciones como SSH. BLOQUEAR.

smtp (25/TCP, UDP)
Históricamente la mayoría de las entradas en hosts han venido por este puerto. Se debe mantener siempre la última versión estable conocida de cualquier programa de correo, especialmente si trabajamos con sendmail.

time (37/TCP, UDP)
Devuelve la hora del sistema en un formato legible por la máquina. Puede ser accedido tras un ataque vía ntp.

nameserver (42/TCP, UDP)
Si se dispone de una red privada se debe instalar un servidor de nombres para ella. Hay que bloquear el acceso a dicho servidor desde el exterior y utilizar siempre la última versión de BIND para resolver nombres. En este caso se puede cortar el acceso al DNS desde UDP.

tftp (69/TCP, UDP)
Falta de autentificación. Bloquear si no se dispone de máquina con arranque remoto.

private dialout (75/TCP, UDP)
Si el los logs aparece una traza de este puerto, en el mejor de los casos estamos siendo analizados por un scanner de puertos. BLOQUEAR.

finger (79/TCP, UDP)
Puede obtenerse información acerca de usuarios concretos, información que puede utilizarse para adivinar claves de acceso. Bloquear o sustituir por una política coherente de asignaciones de correo (Juan Fernández → juan.fernandez@host.com) y un mensaje advirtiendo de dicha política.

npp (92/TCP, UDP) (Network Printing Protocol)
Nadie quiere imprimir documentos ajenos, ¿verdad?

objcall (94/TCP, UDP) (Tivoli Object Dispatcher)
Utilizado por la herramienta de gestión de redes Tivoli. Si utilizamos Tivoli aplicar las mismas precauciones que con SNMP.

sunrpc (111/TCP, UDP)
Especialmente peligroso sobre UDP. No autentifica fuentes, y es la base para otros servicios como NFS.

auth (113/TCP, UDP)
No debería permitirse obtener información acerca de puertos privilegiados (puede utilizarse para realizar un portscan). No se utiliza más que en Unix.

ntp (123/TCP, UDP) (Network Time Protocol)
Se utiliza para sincronizar los relojes de las máquinas de una subred. Un ejemplo de ataque clásico consiste en enviar paquetes a este puerto para distorsionar los logs de la máquina.

netbios (137, 138, 139/TCP, UDP)

No dispone de suficiente autenticación. Afortunadamente según los RFC 2001 y 2002 NetBIOS es capaz de funcionar correctamente a pesar de que se estén enviando bloques de datos con información errónea o corrompida.

snmp (161/TCP, UDP)
¿Quién puede querer administrar nuestra red desde el exterior? Se puede obtener mucha información a través de este servicio, como por ejemplo el estado de los interfaces de red, conexiones concurrentes en la máquina, etc… BLOQUEAR.

snmp-trap (162/TCP, UDP)
Traps de SNMP. A través de este puerto se realizan solicitudes que pueden cambiar la configuración del host. BLOQUEAR.

irc (194/TCP, UDP)
No es peligroso en sí. Sin embargo sus usuarios suelen divertirse atacando los hosts de otras personas con el fin de echarlos cuando no pueden hacer uso de la orden kick. Generalmente conviene bloquear los puertos 6666, 6667 y 6668, que son a los que generalmente se enganchan los servidores irc.

exec (512/TCP, UDP)
Ejecuta órdenes en estaciones remotas. No se realiza más autentificación que la basada en dirección IP y usuario remoto. MUY PELIGROSO. BLOQUEAR.

biff (512/UDP)
Notifica la llegada de correo. Buen candidato para posibles desbordamientos de buffer, o simplemente para obligar a abandonar la sesión a un usuario debido a la llegada masiva de mensajes de correo. BLOQUEAR.

login (513/TCP)
BLOQUEAR.

who (513/UDP)
Muestra quién está utilizando el host remoto. Se puede obtener información bastante detallada acerca de quién utiliza una máquina y desde qué terminal, carga de la máquina… BLOQUEAR.

cmd (514/TCP)
Similar a exec, mismas precauciones. BLOQUEAR.

syslog (514/UDP)
BLOQUEAR a menos que existan suficientes razones para mantenerlo. Suele utilizarse para corromper los logs del sistema con entradas falsas.

router (513/TCP, UDP)
BLOQUEAR.

ingreslock (1524/TCP)
En la mayoría de los Unix se puede encontrar esta entrada en /etc/services. Ya que está dado de alta y es un puerto no privilegiado es un buen lugar para una puerta trasera. (No sería la primera vez que ocurre)

CONCLUSION

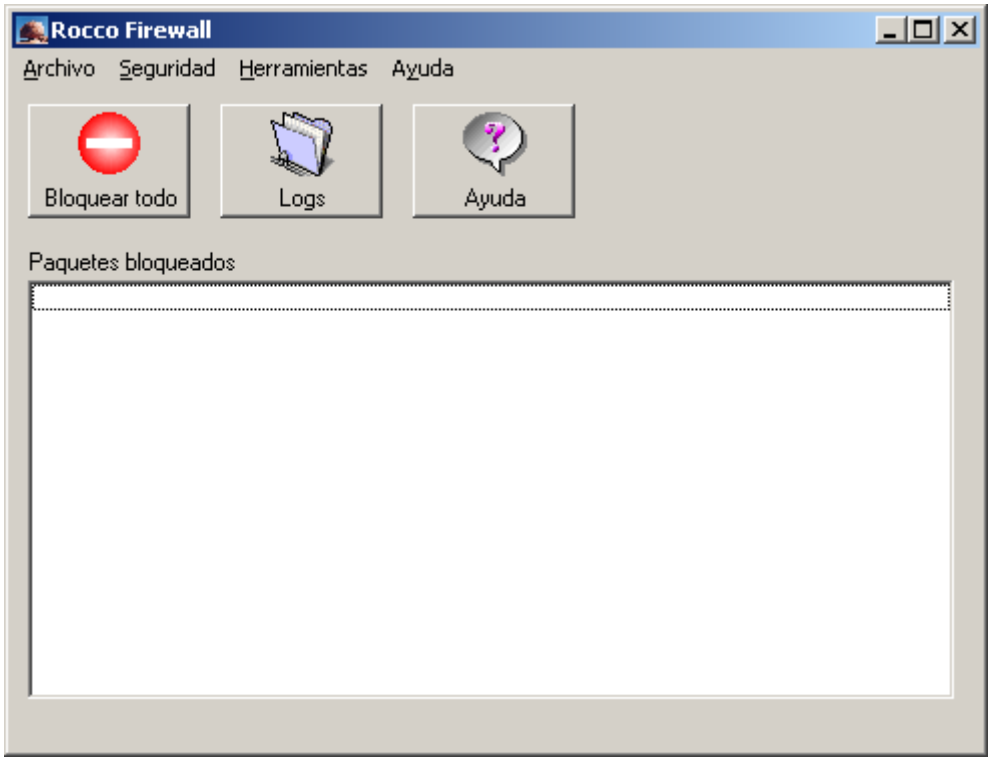
1. Producto obtenido

Es un firewall que consta de una aplicación que funciona en modo usuario y un driver que funciona en modo kernel. La aplicación permite al usuario introducir reglas de filtrado de una forma fácil e intuitiva, que son inmediatamente transmitidas al driver, el realiza el filtrado de paquetes.

Las reglas avanzadas ofrecen la posibilidad de filtrar por puertos, por direcciones y por protocolos, especificando si los paquetes a filtrar son de entrada, de salida o ambos.

Mediante un fichero de configuración estas reglas son guardadas de una ejecución a otra del programa.

La principal ventaja de este firewall, aparte del hecho de ser gratuito, es que su código es libre. Tanto el código como toda la documentación pertinente están disponibles en la red.



Funcionalidades

- Bloqueo de emergencia que bloquea todas las comunicaciones.

- Posibilidad de introducir reglas avanzadas de filtrado de paquetes.
- Muestra toda actividad por pantalla y la almacena en un fichero de log.
- Se puede usar en entornos empresariales
- Se espera que goce de actualizaciones muy frecuentes.
- Usa Drivers NDIS IM
- Consume escasos recursos de sistema

2. Valoración

El producto obtenido es una aplicación completamente funcional formada por 2 componentes: un driver y un programa ejecutable.

A continuación se mostrarán los requisitos que se han logrado cumplir y la valoración de éstos en base a una puntuación del 1 al 5, siendo 1 el mínimo y 5 el máximo, según el grado de satisfacción de cada uno de los requisitos.

Requisito	Puntuación	Explicación
Facilidad de uso	3	La aplicación de usuario es muy sencilla, pero la instalación del driver no es un proceso transparente.
Multiplataforma	3	La aplicación es compatible con Windows 2000, XP y 2003
Especialización	4	Permite definir reglas muy precisas
Información al usuario	3	Muestra las reglas definidas y los paquetes que se bloquean. Se podría haber mejorado mostrando la aplicación que genera los paquetes y gráficas sobre el tráfico.
Rendimiento	5	Al funcionar en el nivel más bajo del núcleo de Windows, el rendimiento es máximo.
Código Abierto	5	El código se dejará en Internet.
Libre distribución	5	La aplicación estará compilada y se podrá descargar de Internet.
Utilización en entornos empresariales	5	Debido a los 2 requisitos anteriores, no se impone ninguna traba para el uso en entornos empresariales.
Adaptabilidad	4	Al ser de código abierto y libre distribución, la aplicación podrá evolucionar con rapidez. No se otorga la puntuación máxima ya que puede que no tenga la aceptación esperada entre el público.

3. Comparativa final

A continuación se expondrá una tabla valorando el producto obtenido con los ya existentes comentados en la introducción:

	Facilidad de uso	Multiplataforma	Especialización	Información al usuario	Rendimiento	Código Abierto	Libre distribución	Utilización en entornos empresariales	Adaptabilidad	Total
RoccoFirewall	3	3	4	3	5	5	5	5	4	**37**
Windows Firewall	3	1	2	2	5	0	2	5	3	**23**
Zone Alarm	3	5	3	3	3	0	5	0	3	**25**
Sygate Personal Firewall	4	5	5	5	4	0	5	0	3	**31**

Como se puede observar, el producto obtenido pierde en algunos aspectos técnicos respecto a algunos de los productos, pero aún así resulta muy competitivo y teniendo en cuenta de que es la primera versión, el desarrollo se considera exitoso.

4. Posibles mejoras

Como se comentaba en la parte introductoria del documento, lo ideal hubiese sido introducir funcionalidad de detección de intrusión (IDS) pero lamentablemente este aspecto no ha sido abordable dada la escasez de personal y tiempo.

Otros objetivos alcanzables a más corto plazo, hubiesen sido

- Poder certificar el driver mediante Microsoft, para así poder desarrollar un instalable que introdujese el driver en el sistema de forma automática. Este apartado no se pudo llevar a cabo por falta de tiempo y recursos económicos. Si esto pudiese haberse realizado, se podría haber aumentado la puntuación en el requisito de facilidad de uso ya que la instalación del driver sería transparente al usuario.

- Mostrar gráficas sobre el tráfico entrante, saliente y bloqueado, este apartado no ha sido abordable por falta de tiempo.

- Permitir filtrar por aplicaciones, al igual que en el apartado anterior, el tiempo ha sido el detractor de esta implementación.

Así pues, se espera que otro equipo tome el relevo de este desarrollo, y pueda cumplir los objetivos anteriormente mencionados.

GLOSARIO

A

- Ataques DOS: Tipo de ataque contra un ordenador que impiden que éste pueda ofrecer sus servicios normalmente.

B

C

- Código abierto: El código fuente está disponible para que otros desarrolladores puedan modificarlo a su antojo.
- Controlador: Código que, manejado por el Sistema operativo, permite a éste manejar un dispositivo.
- Cracking: Rotura del sistema de seguridad de un programa o dispositivo para fines maliciosos.

D

- Dirección IP: Identificador de un ordenador para el protocolo IP.
- DNS: (Domain Naming Service) Permite el uso de nombres fáciles de recordar para los humanos en lugar de usar direcciones IP.
- DOS: (Denial of Service) Ver “Ataques DOS”.
- Driver: Ver “Controlador”.

E

F

- Firewall: Programa o dispositivo que, mediante unas reglas, impide el acceso por red a un ordenador o a una red.

G

- Gateway: Maquina que sirve de conexión hacia una red externa.

H

I

- ICMP: (Internet Control Message Protocol) Protocolo de Internet que permite el control y monitorización de algunos aspectos de la red.
- IDS: (Intrusion Detection System) Programa que detecta ataques contra un ordenador o una red.
- IOCTL: (I/O Control) Estructura proporcionada por Windows para poder hacer peticiones a un driver.

- IOManager: Componente del núcleo de Windows que gestiona todas las Entradas/Salidas del sistema.
- IP: (Internet Protocol) Protocolo de comunicaciones por red en el que se basa Internet.
- IRP: (I/O Request Packet) Estructura de Windows para cualquier petición de Entrada/Salida.

J

K

- Kernel: Núcleo del sistema operativo.
- Kernel-Mode: Modo de funcionamiento del sistema operativo en el que las instrucciones son ejecutadas por el procesador en modo privilegiado.

L

- LAN: Red de área local.
- Linux: Sistema operativo de código abierto.
- Log: Registro en el que se guardan los sucesos de una aplicación.

M

N

- NDIS: (Network Driver Interface Specification) Interfaz para el desarrollo de drivers en Windows.
- NDIS IM: Driver NDIS Intermedio, añade funcionalidades o realiza traducciones de paquetes.
- NDIS Miniport: Driver NDIS de bajo nivel, controla los adaptadores de red.
- NDIS TDI: Driver NDIS de la capa de Transporte.

O

- Open Source: Código abierto.

P

- Paquete: Contenedor de la información recibida desde una red.
- Puerto: Punto de entrada a un ordenador para el protocolo TCP o UDP.

Q

R

S

- Sistema Operativo: Programa que controla los dispositivos del ordenador, permitiendo al usuario el manejo de estos.

- Spam: Correo electrónico no deseado.
- STL: Librerías de plantillas estándar de C++. Facilitan el uso de estructuras como listas, vectores, tablas hash…

T

- TCP: (Transmission Control Protocol) Protocolo orientado a conexión para la transmisión de datos por Internet.
- Troyano: programa que se instala en un ordenador y permite la ejecución de código o envía información privada sin conocimiento ni consentimiento del usuario legitimo del ordenador.

U

- UDP: (User Datagram Protocol) Protocolo no orientado a conexión para la transmisión de datos por internet.
- User-Mode: Modo de funcionamiento normal de cualquier aplicación.

V

- Virus: Programa que una vez instalado en el ordenador, destruye datos o degrada el rendimiento.
- VPN: Red Privada Virtual.

W

- Windows: Familia de sistemas operativos de Microsoft. Son los más utilizados en el mundo.
- Winsock: Librerías de Windows para el trabajo con redes.
- Wrapper: Traductor. En este caso permite utilizar C++ cuando se debería usar C.

X
Y
Z

REFERENCIAS

[1] Art Baker y Jerry Lozano, The Windows 2000 Device Driver Book, A guide for programmers (2ª Ed.), Prentice Hall

[2] Documentación de Microsoft Windows XP DDK, Microsoft Corporation

[3] Microsoft MSDN Library, Microsoft Corporation

[4] Handling IRPs: What Every Driver Writer Needs to Know, Microsoft Corporation

[5] Thomas F. Divine, Extending The Microsoft PassThru NDIS Intermediate Driver, Printing Communications Assoc., Inc. (PCAUSA)

[6] C++ in the Kernel, Compuware Corporation

[7] C++ Run Time Support for NT/WDM Kernel Mode Drivers, Compuware Corporation

[8] Implementing a C++ framework for NDIS driver development, Compuware Corporation

[9] Recursos de Microsoft para desarrolladores, http://www.microsoft.com/whdc

[10] Herramientas para el desarrollo de drivers, http://www.sysinternals.com/ntw2k/utilities.shtml

[11] FAQ's sobre el desarrollo de drivers, http://www.cmkrnl.com/faq.html

[12] FAQ's explícito sobre NDIS, http://www.ndis.com

[13] FAQ's y WhitePapers sobre el desarrollo de drivers NDIS, http://www.pcausa.com/resources/ndisfaq.htm

[14] TCP/IP Frame Format, http://www.zytrax.com/tech/protocols/tcp.html