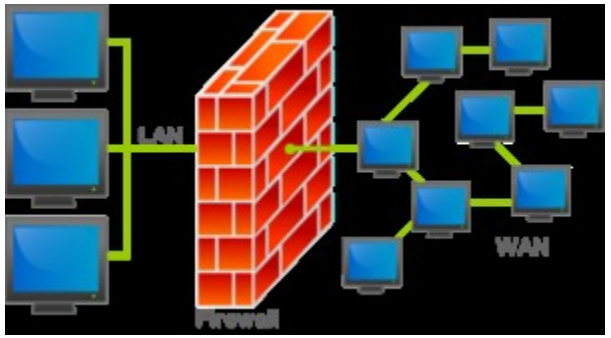
Seguridad y Privacidad en Redes

Firewalls



Firewalls

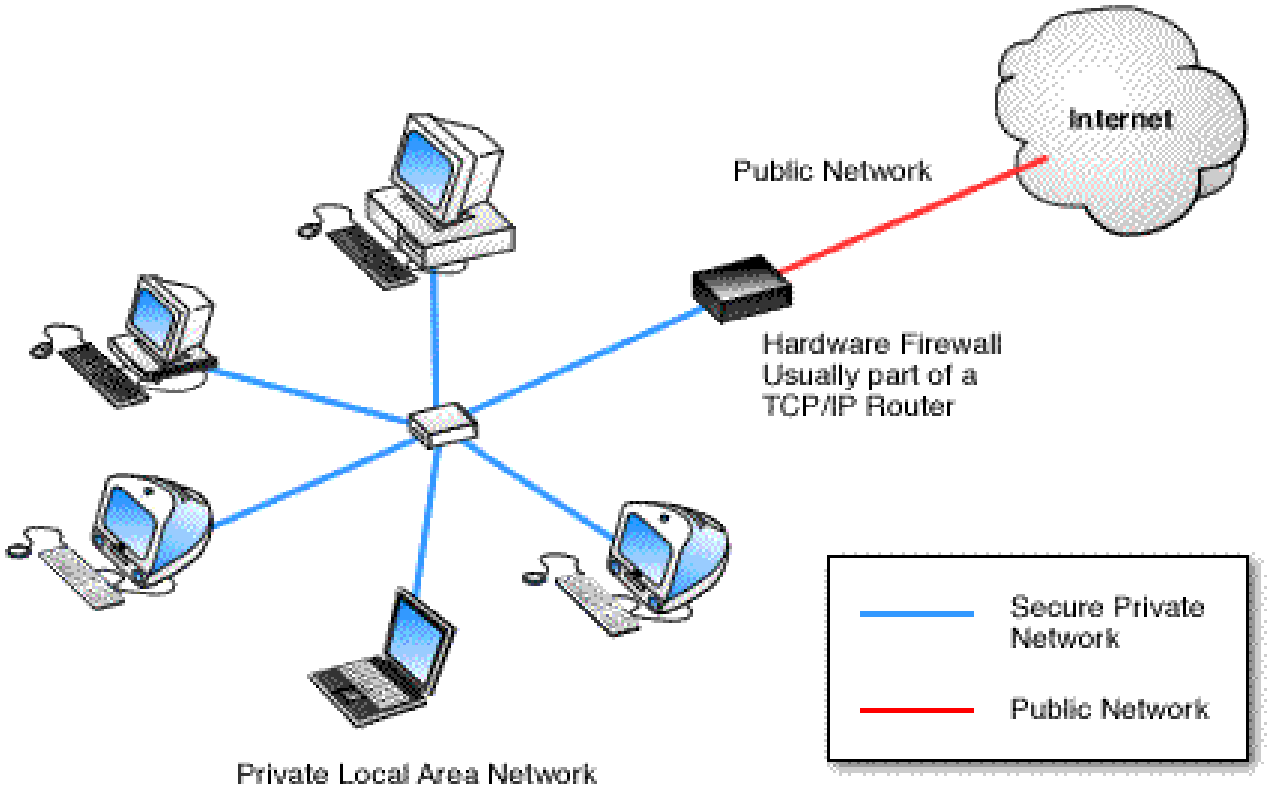
¿Qué es?

Es un mecanismo que ayuda a prevenir el acceso de programas o usuarios no autorizados a redes privadas o a computadoras personales.

¿Con qué objeto?

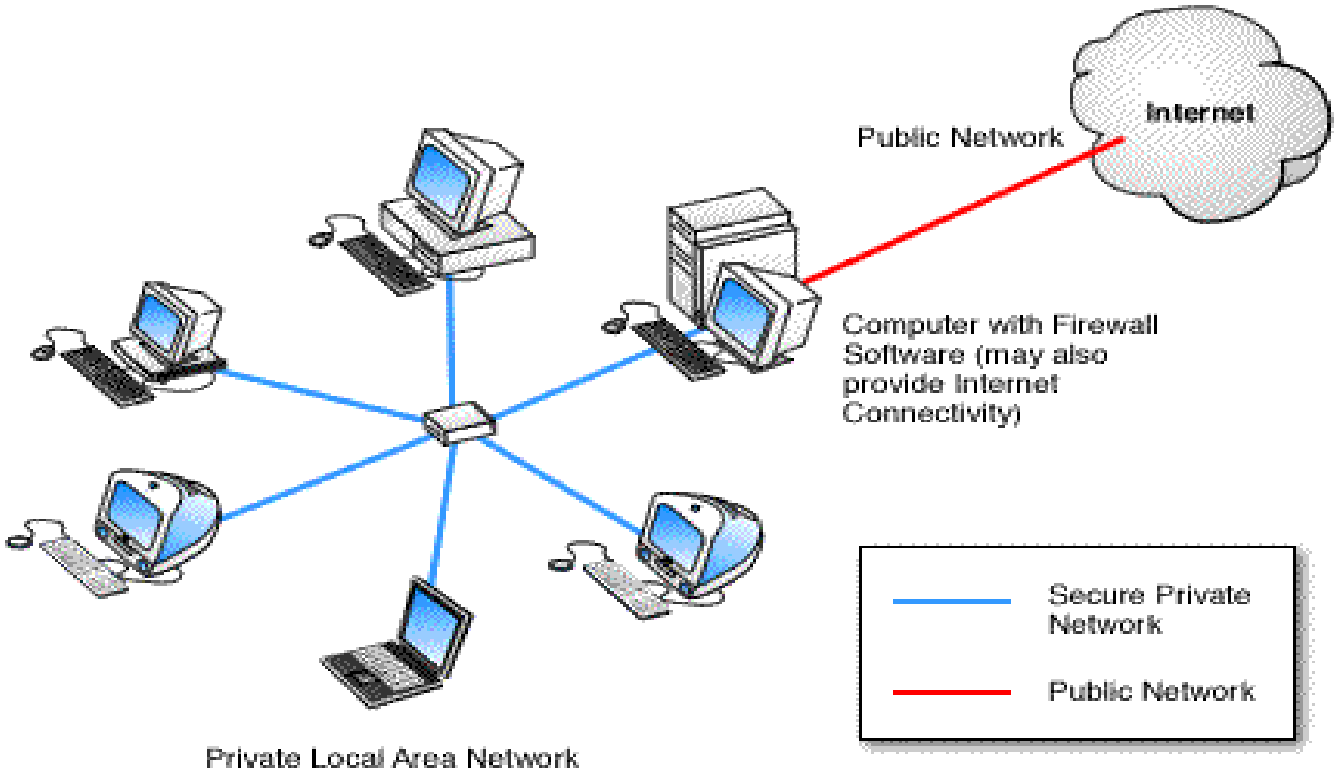
- Proteger una red
- Ocultar la estructura interna de una red
- Loguear el tráfico que entra y sale de una red
- Establecer políticas para mejorar la seguridad. Ej. evitar el spoofing, extrusion detection, etc.

Firewalls



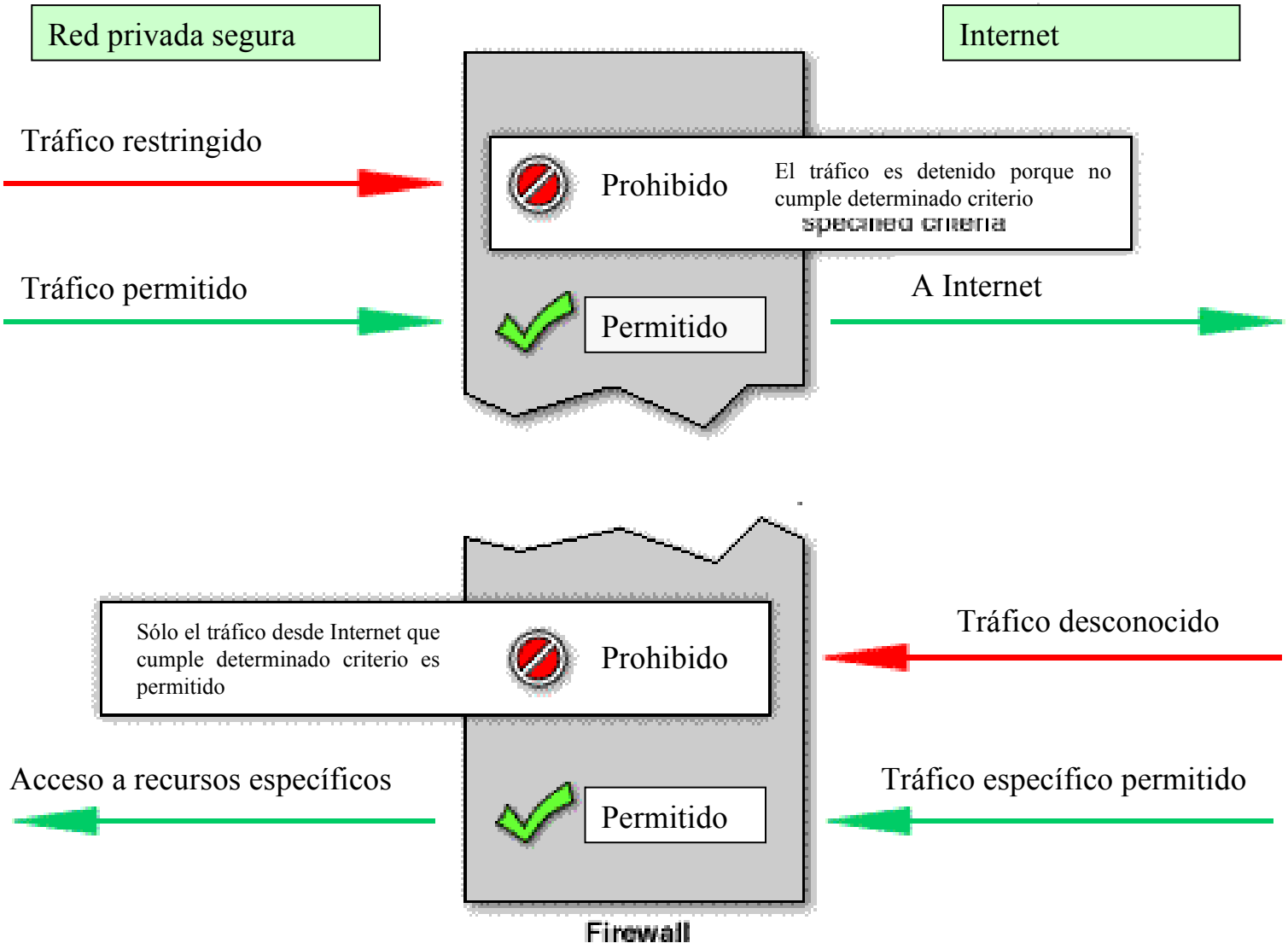
Firewall implementado en hardware

Firewalls



Firewall implementado en software

Firewalls


Red privada segura
Internet
Tráfico restringido
Prohibido
El tráfico es detenido porque no cumple determinado criterio
Tráfico permitido
Permitido
A Internet
Sólo el tráfico desde Internet que cumple determinado criterio es permitido
Prohibido
Tráfico desconocido
Acceso a recursos específicos
Permitido
Tráfico específico permitido
Firewall

Políticas de Firewall

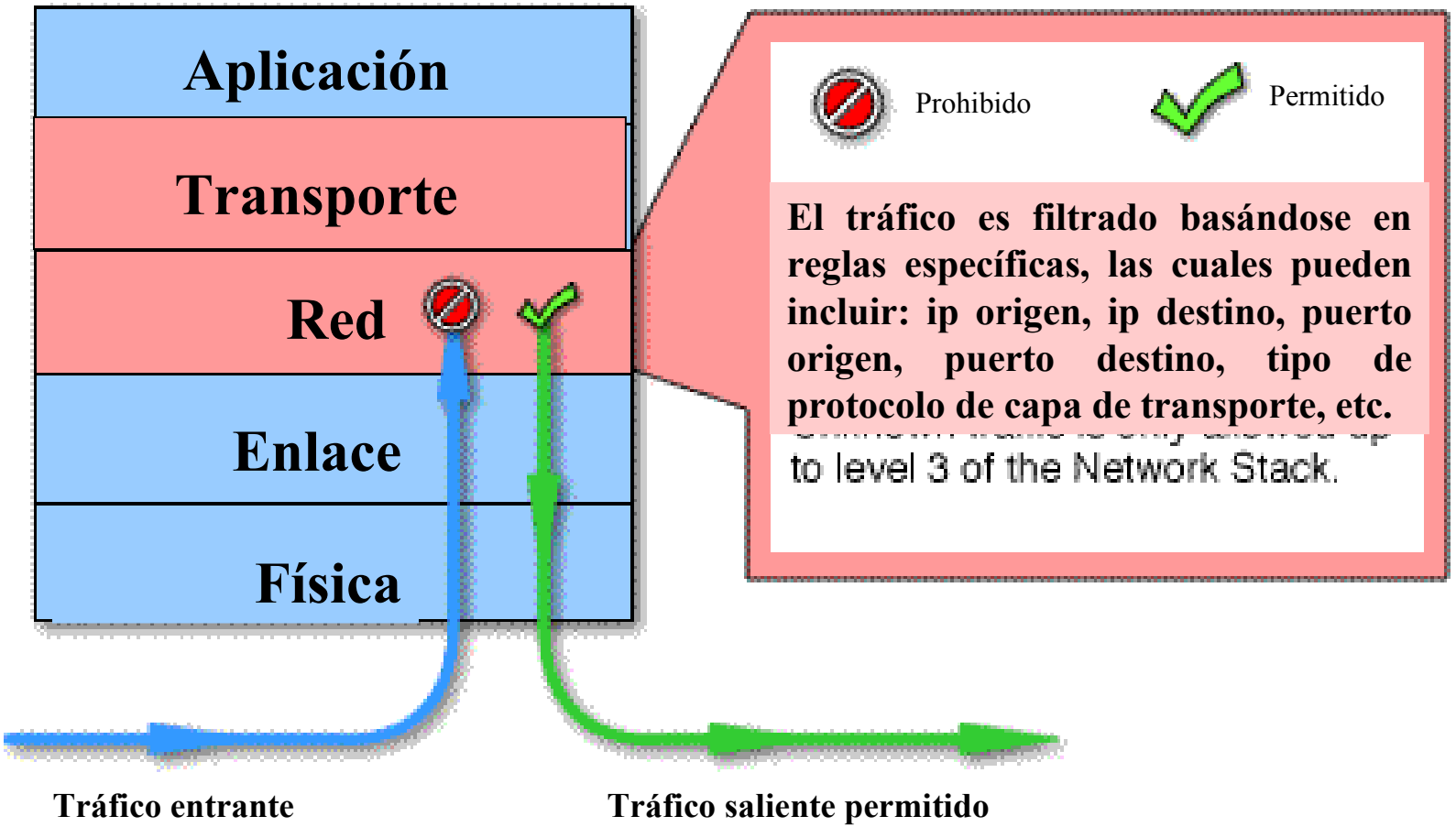
· **Restrictiva**:
Lo que no está expresamente permitido está prohibido. En modelos como éste, se identifica aquellos servicios que deben ser permitidos y las medidas de seguridad que serán aplicadas. El resto de servicios serán bloqueados por defecto.

· **Permisiva**:
Lo que no está expresamente prohibido está permitido. Se trata de encontrar qué tipo de servicios no deben estar activos por creer que presentan riesgos en su funcionamiento y por lo tanto podrían poner en una situación de compromiso al sistema.

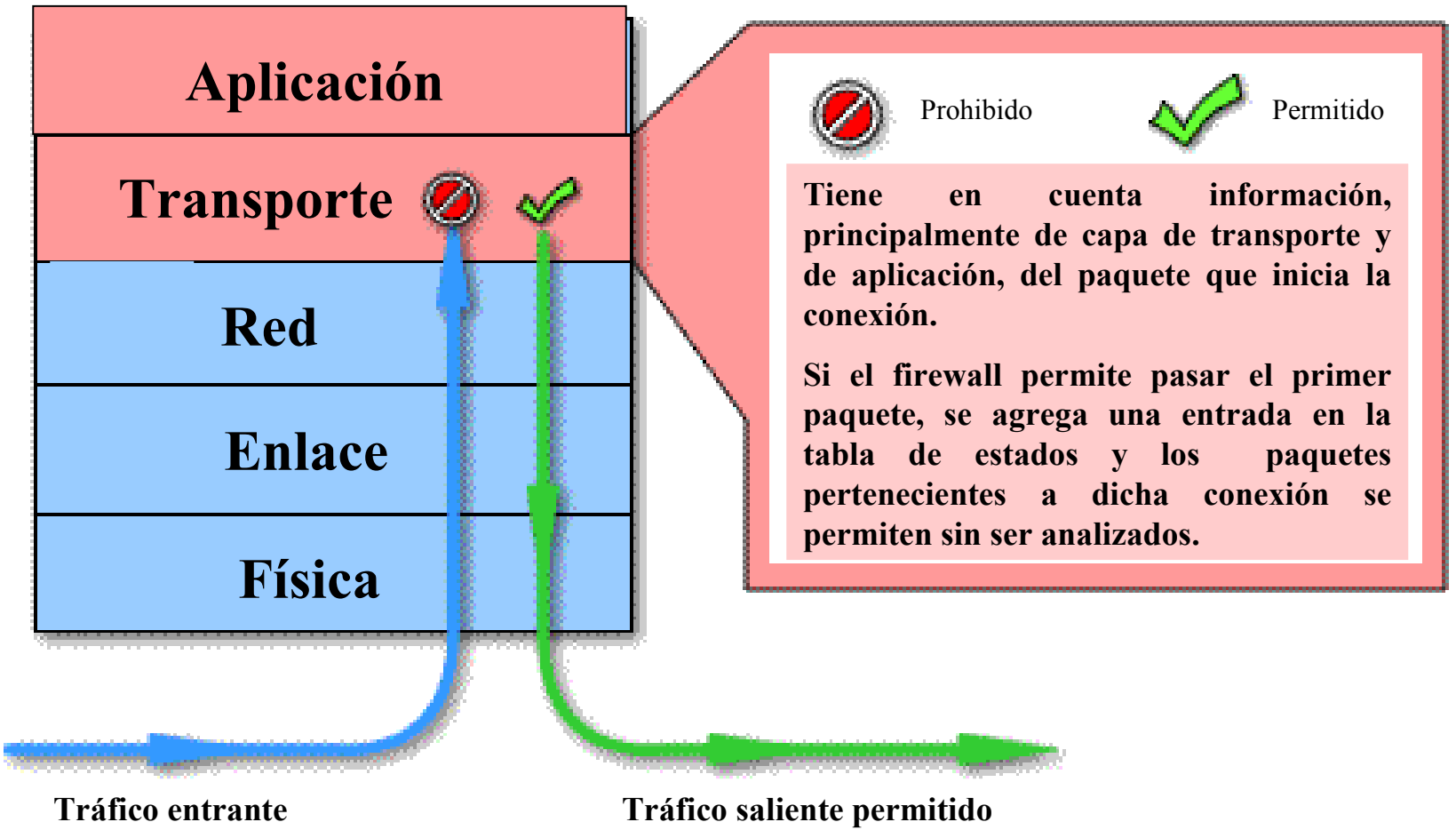
Firewalls - Tipos

- **Packet filtering**
- **Stateful Firewall**
- **Proxy Firewall**
- **Firewalls personales**

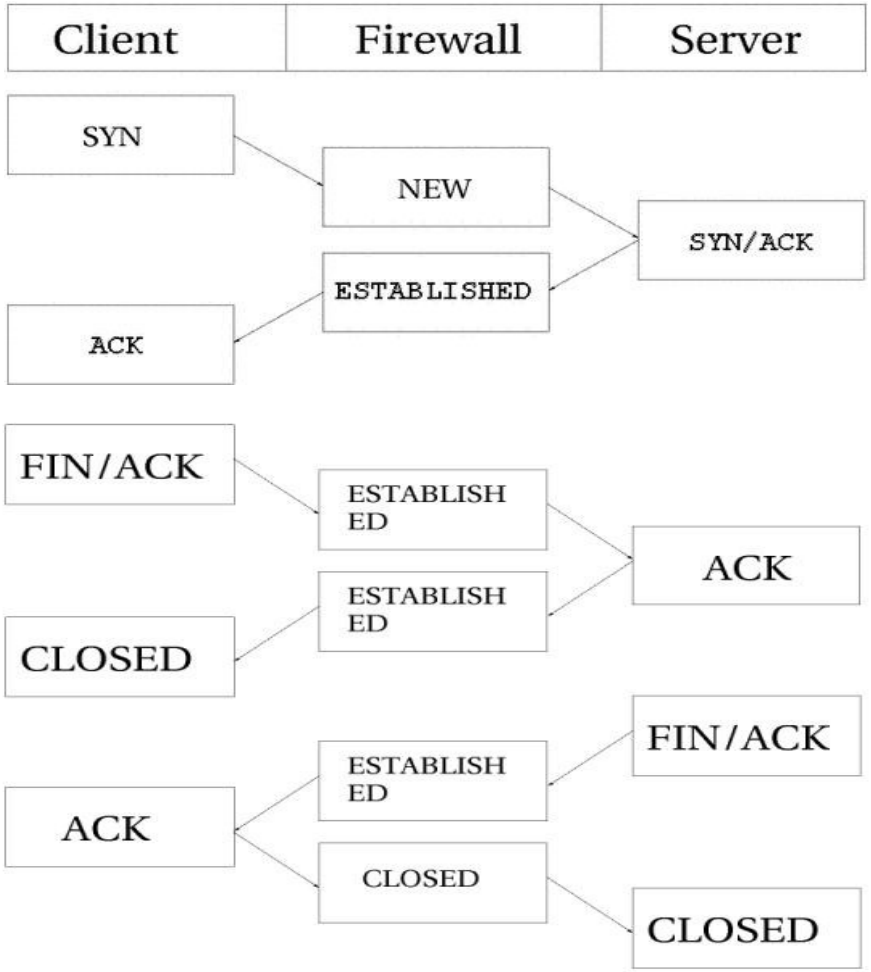
Firewalls – Paquet filtering



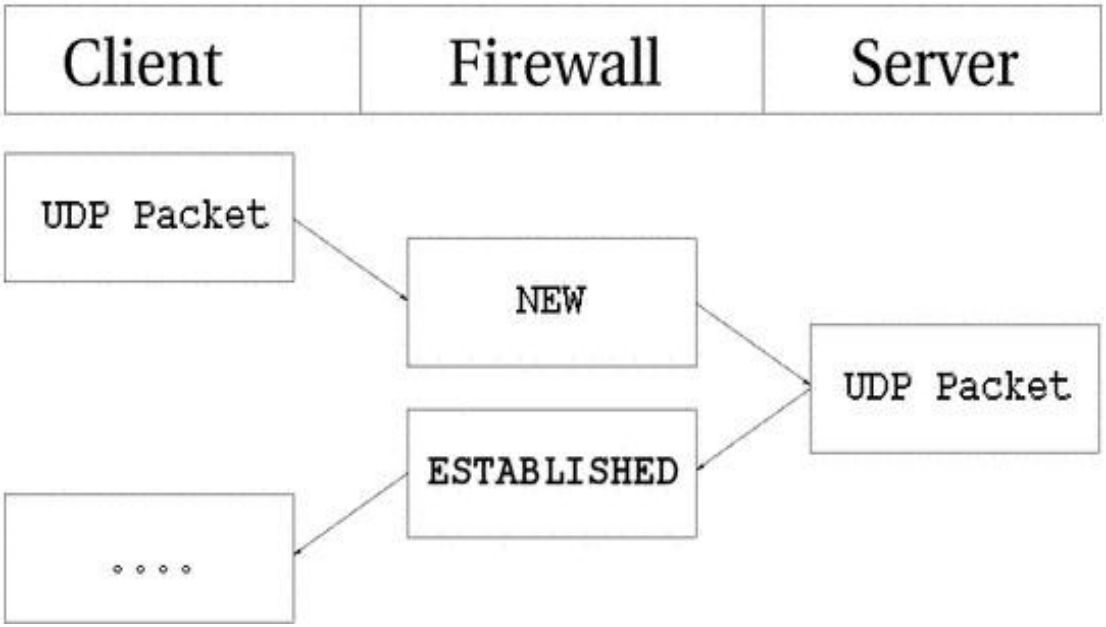
Stateful Firewall



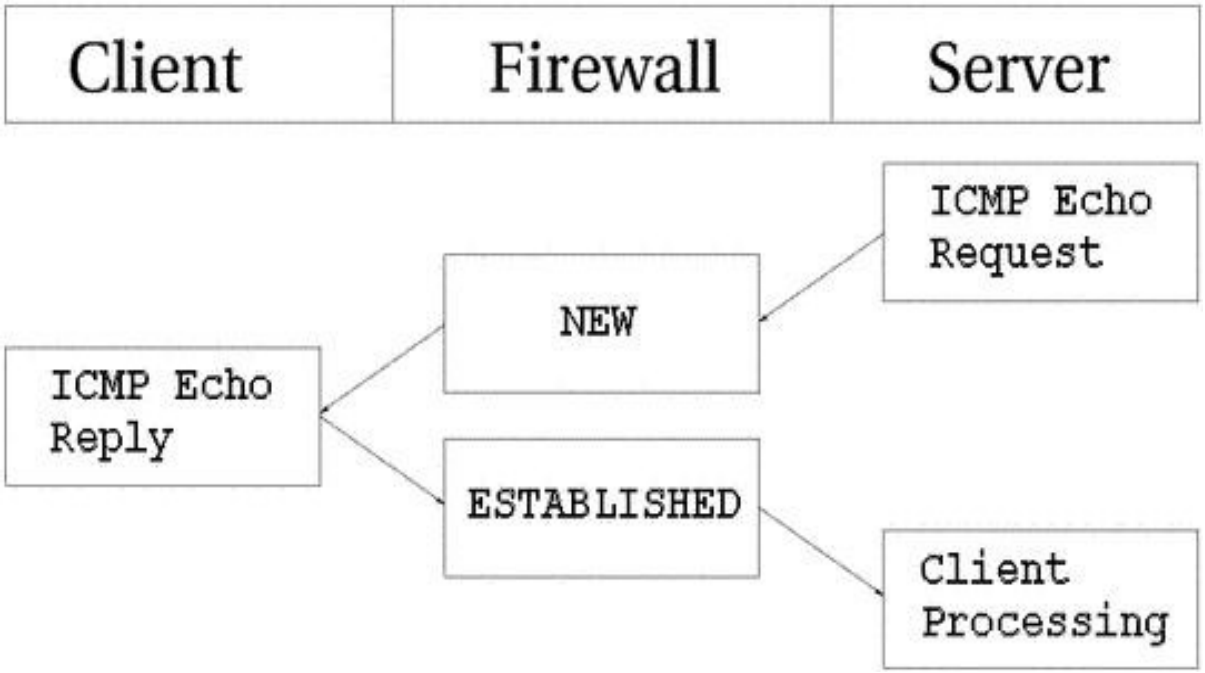
Estados en TCP



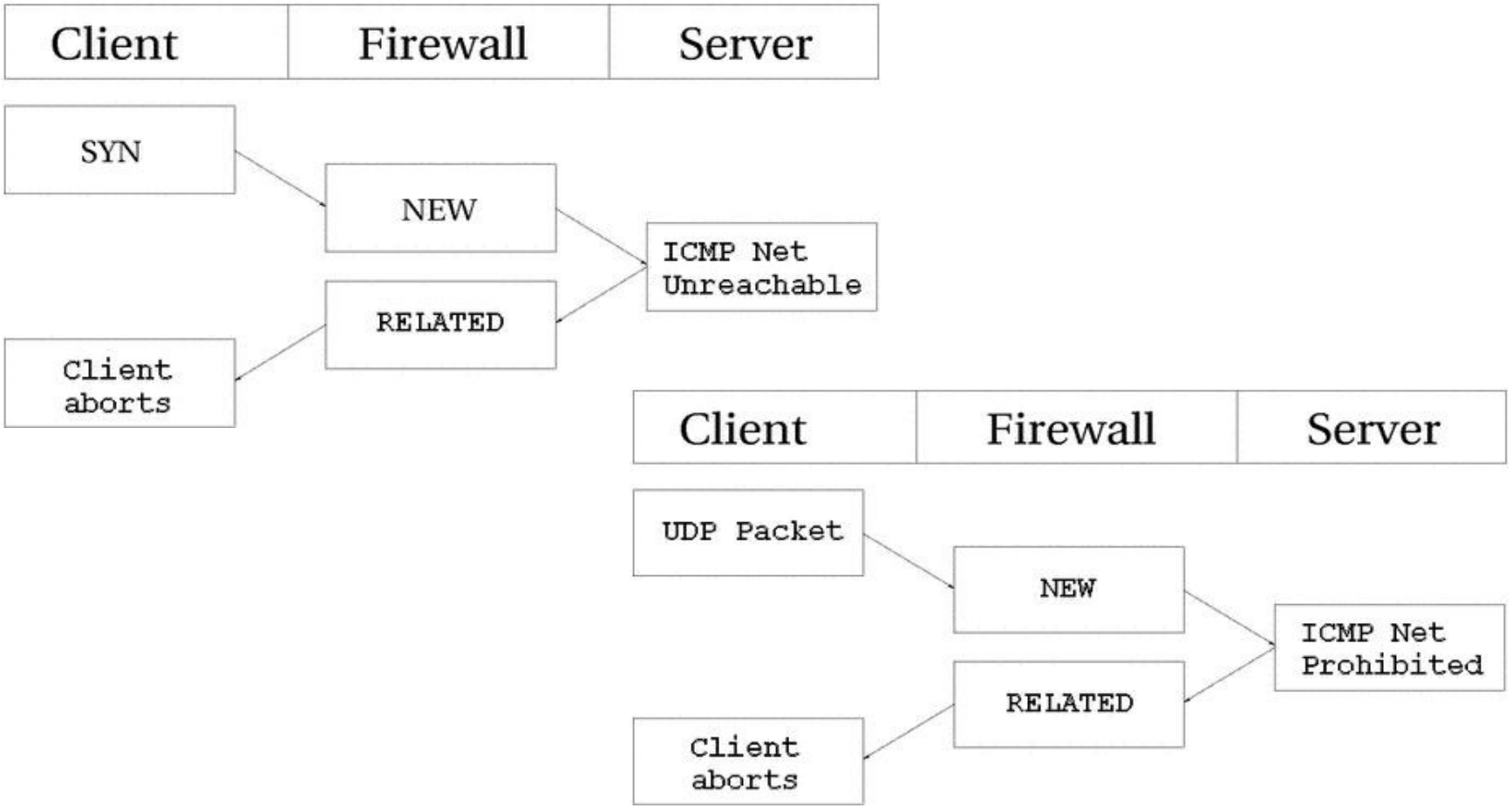
Estados en UDP



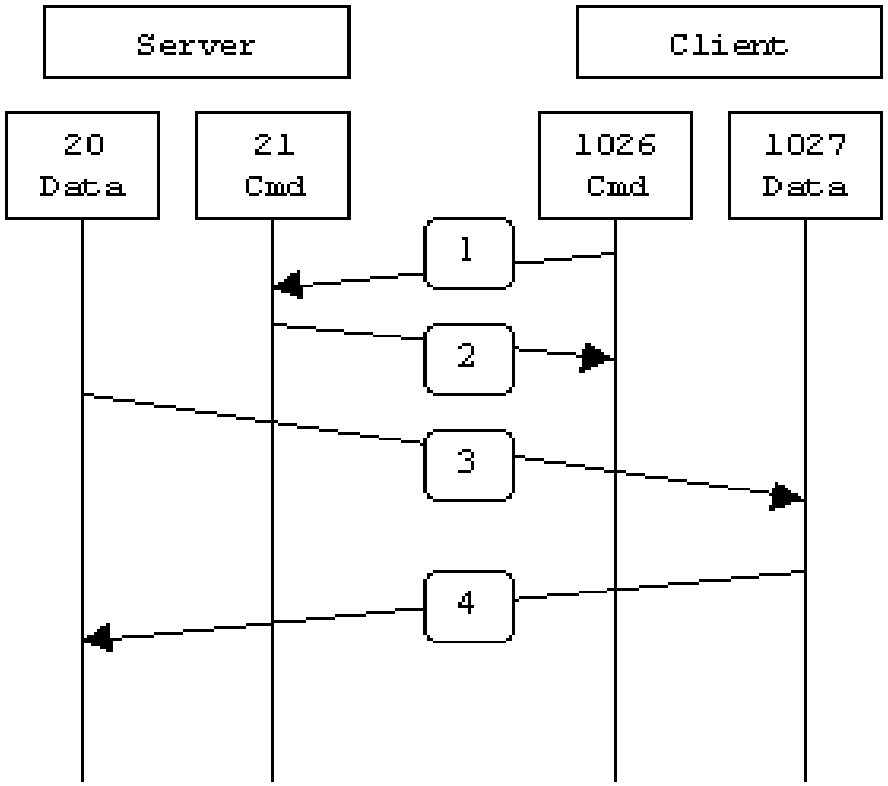
Estados en ICMP



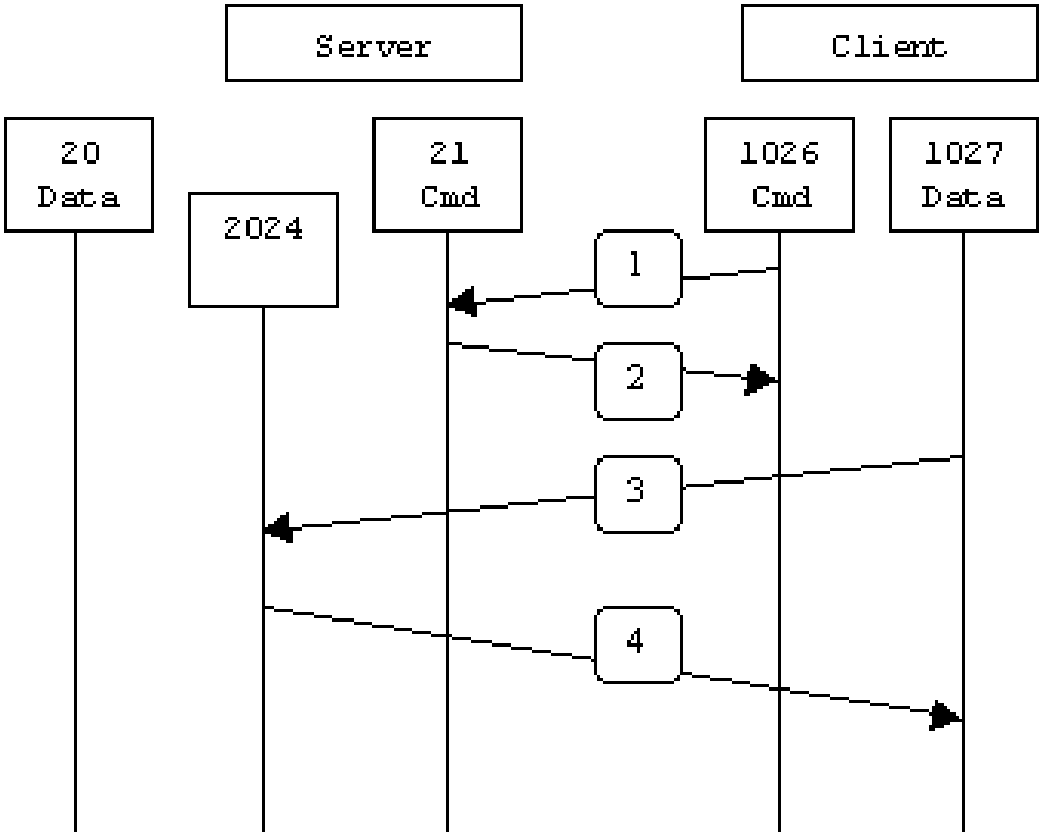
Estados en ICMP (cont)



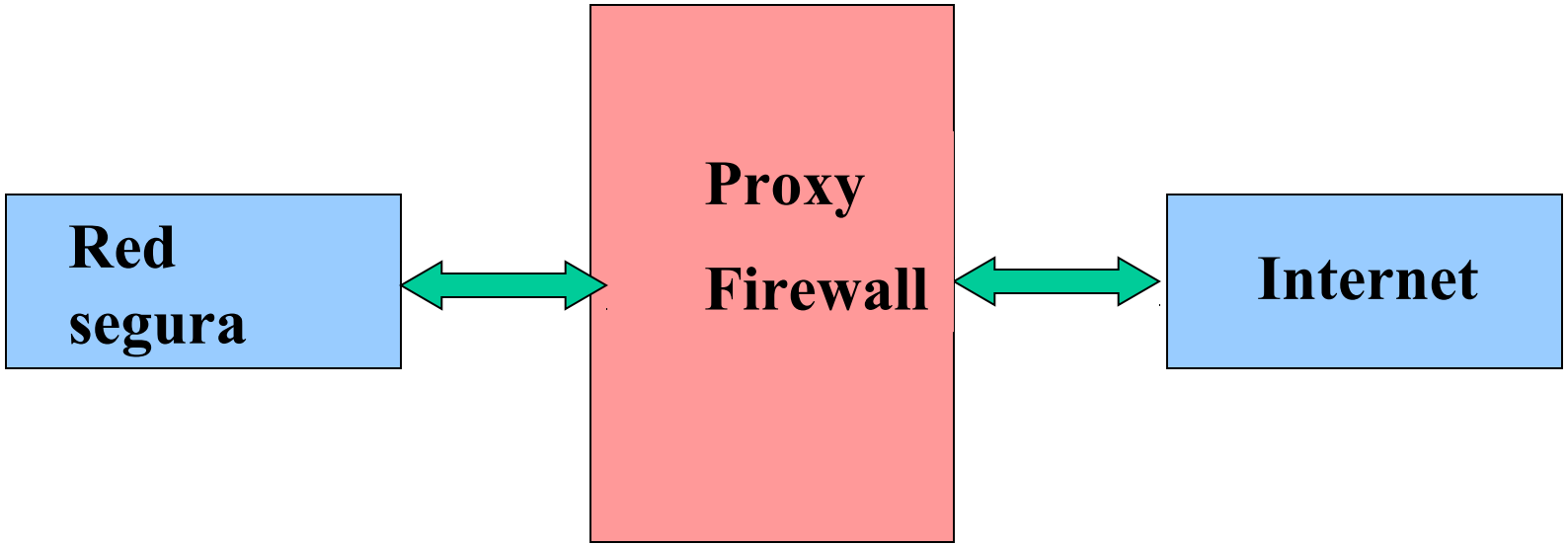
Ejemplo estados: FTP activo



Ejemplo estados: FTP pasivo



Proxy Firewall



El proxy permite una conexión indirecta desde y hacia Internet. Cada comunicación cliente/servidor requiere dos conexiones: una desde el cliente al firewall , el cual actúa como "proxy" del servidor deseado, y otra desde el firewall hacia el servidor deseado.

Firewalls Personales

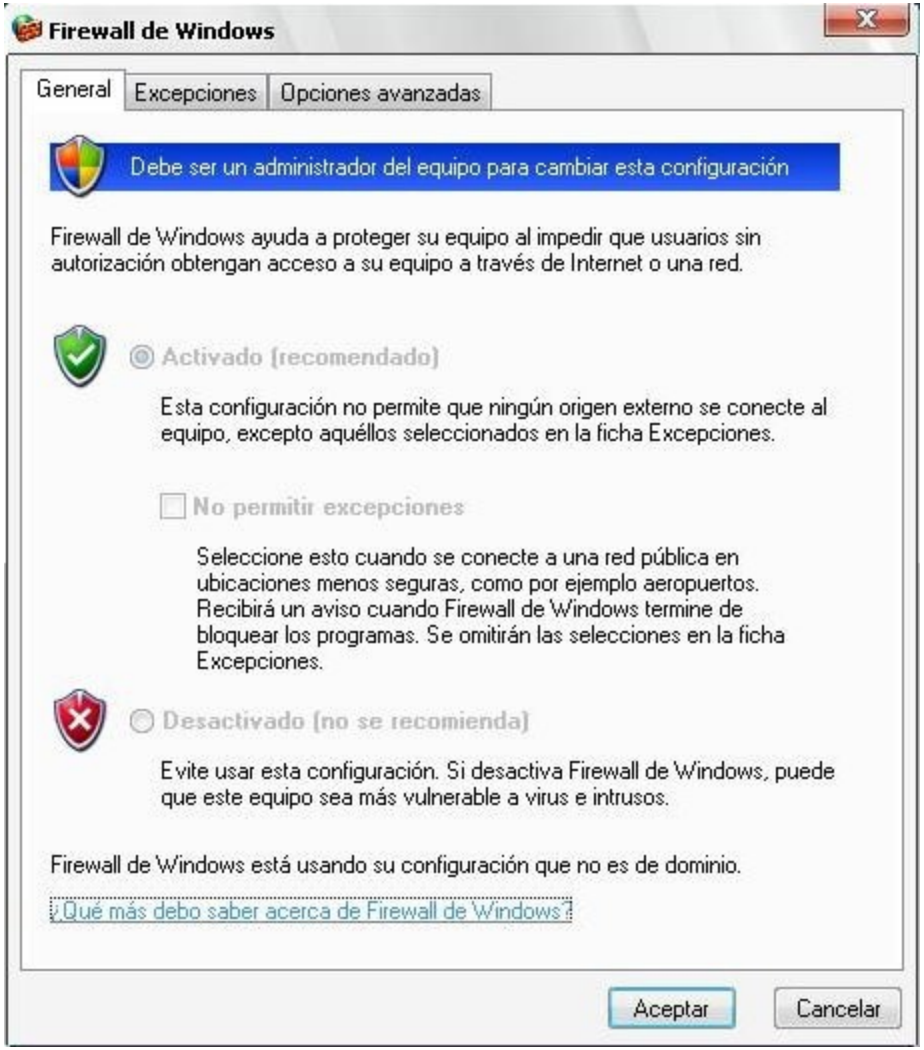
Un firewall personal es una aplicación que controla el tráfico de red que entra y sale de una computadora, permitiendo o denegando comunicaciones, en base a una política de seguridad.

Diseñados típicamente para usuarios finales.

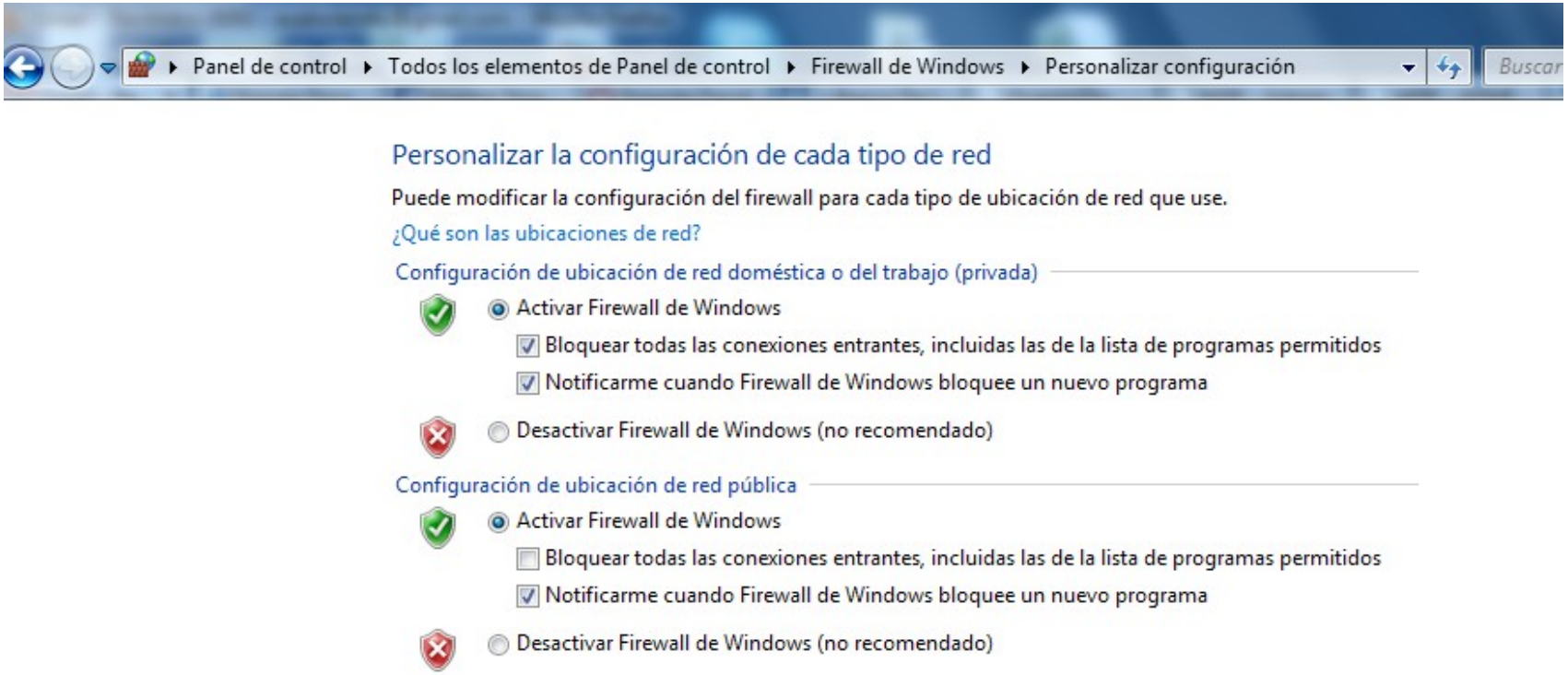
Algunos consultan al usuario cada vez que detectan un intento de conexión permitiendo que la política definida se vaya adaptando.

Ej: ZoneAlarm, Windows Firewall, Kerio Personal Firewall, Comodo Free Firewall y por suspuesto Iptables

Firewalls Personales



Firewalls Personales



Listado en:
http://www.matousec.com/projects/proactive-security-challenge/results.php

Firewalls Personales

Los FW personales entrantes NO alcanzan:
Hay que agregar FW personales que filtren la SALIDA.

Desventaja: Incomodan al usuario con muchas preguntas:

Ideas que podrían añadirse a un FW saliente:

- Que sólo las aplicaciones firmadas pudieran salir sin problema a Internet. Las no firmadas, pedirían confirmación.
- Que sólo las aplicaciones alojadas en ciertas rutas (en las que el usuario no tiene permiso de escritura como tal, sólo como administrador) pudieran salir sin problemas.
- Función para bloquear aplicaciones por hash
- Función para bloquear aplicaciones según el lugar de donde proviene (descargada desde alguna zona concreta de Internet Explorer...)

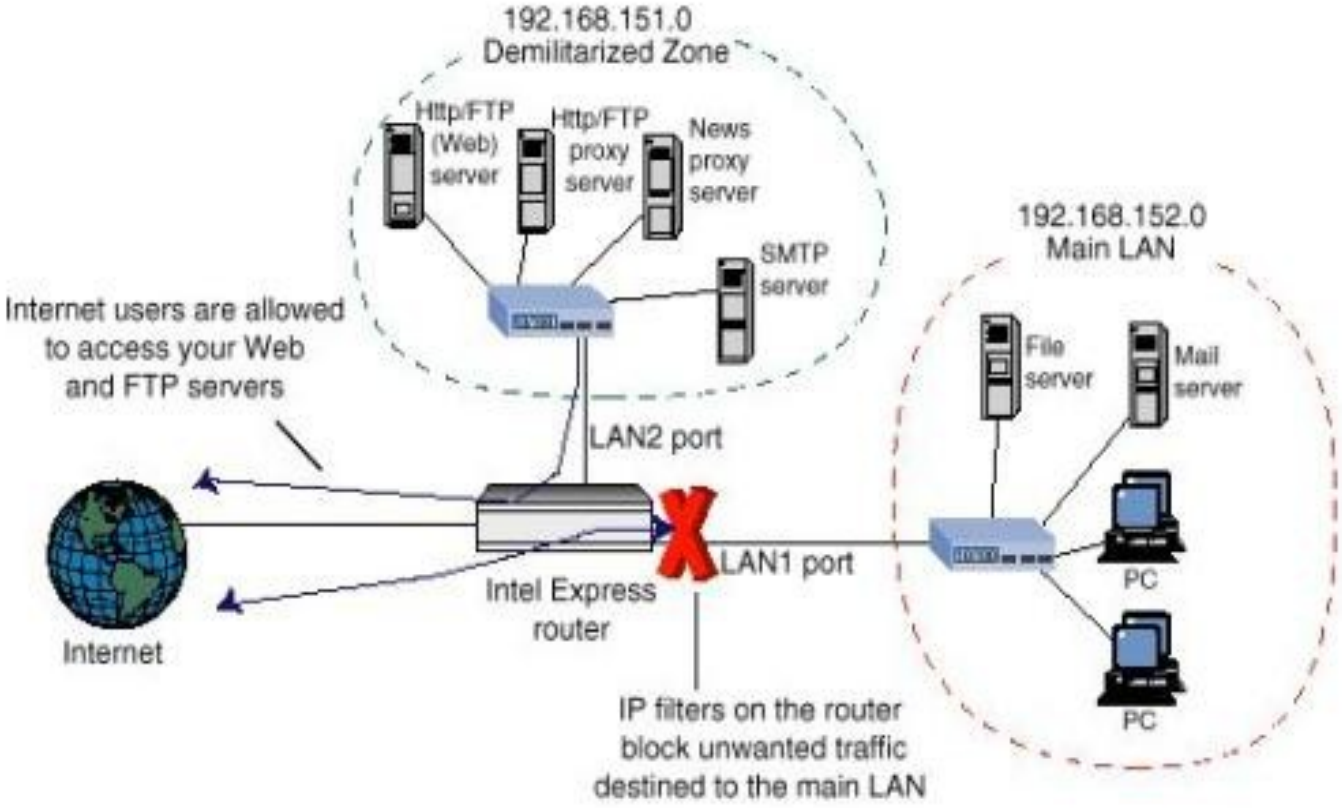
DMZ

DMZ: Demilitarized zone (Zona desmilitarizada).

Es una subred física o lógica que aloja los servicios que una organización brinda a redes externas no confiables. El propósito de una DMZ es agregar un nivel de seguridad a la red local de la organización (ubicando los servidores en otra red que las estaciones de los usuarios)

- Si queremos tener servidores accesibles desde Internet, es conveniente ubicarlos en la DMZ.
- Los servidores locales deben estar ubicados en la red interna o en otra red (es mejor debido a ataques contra navegadores por ej.)
- En la red interna, los usuarios deberían usar proxy servers ubicados en la DMZ para acceder a los servicios de Internet.

DMZ


192.168.151.0
Demilitarized Zone
Http/FTP (Web) server
Http/FTP proxy server
News proxy server
SMTP server
192.168.152.0
Main LAN
File server
Mail server
PC
PC
Internet users are allowed to access your Web and FTP servers
LAN2 port
LAN1 port
Intel Express router
Internet
IP filters on the router block unwanted traffic destined to the main LAN

Productos

FIREWALL: NETFILTER/IPTABLES

Iptables es parte de un framework que reside dentro del kernel de Linux, el cual provee filtrado de paquetes, NAT y manejo de otros atributos de los paquetes tales como TOS y TTL.

Una de las principales características que lo distinguen de su predecesor “ipchains” es la posibilidad de manejar estados para el fitrado de paquetes.

Firewalls

Estructura de Iptables

Tabla Filter

- INPUT
- FORWARD
- OUTPUT

Tabla Nat

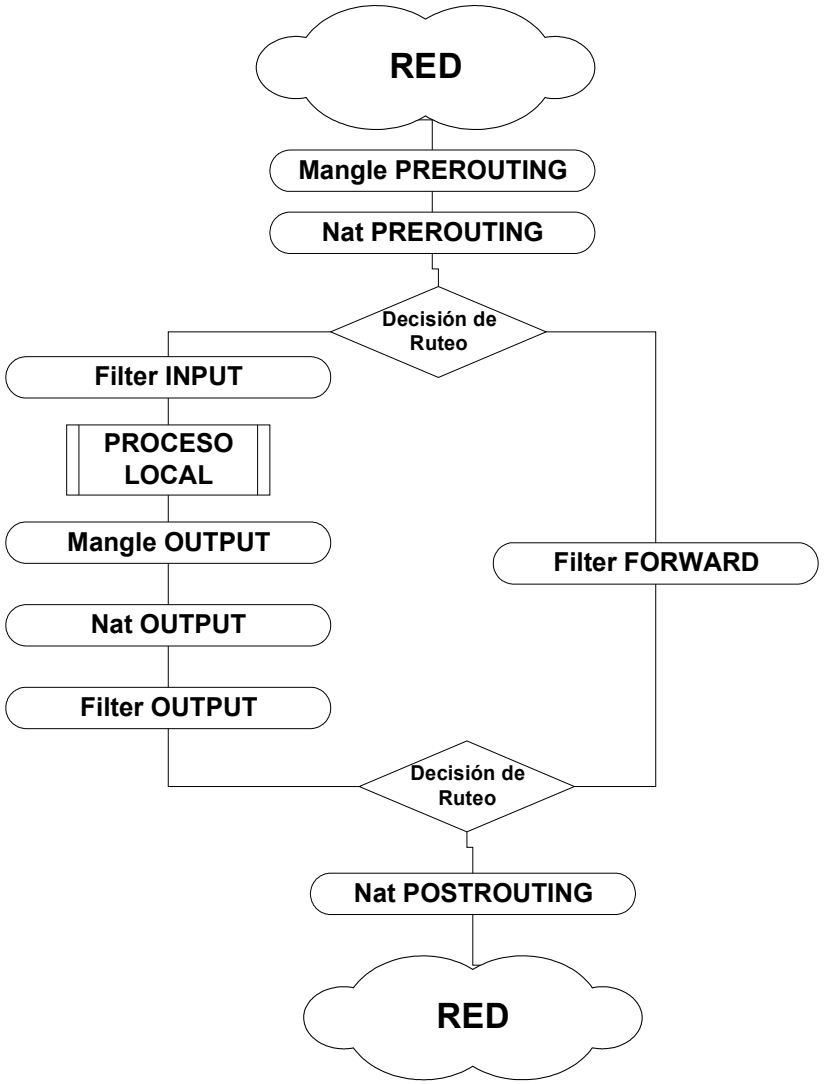
- PREROUTING
- OUTPUT
- POSTROUTING

Tabla Mangle

- PREROUTING
- OUTPUT
- POSTROUTING
- INPUT
- FORWARD

Firewalls
Redirecciones

Iptables- Pasando por las tablas



Firewalls

Ejemplo: Políticas

¿Cómo se define una política por defecto para una cadena en Iptables?

- Política Restricitiva:
  Inicialmente: iptables -P INPUT DROP
  Luego lo que queremos permitir

- Política Permisiva:
  Inicialmente: iptables -P INPUT ACCEPT
  Luego lo que queremos prohibir

Firewalls – Iptables

Definición de reglas

Ejemplo: Salida a Web

En este ejemplo lo que queremos es permitir el acceso al tráfico HTTP para los hosts de la red privada (10.0.50.0/24)

Presentaremos tres soluciones posibles para este problema:

- Solución 1: Permitir sólo el tráfico saliente con puerto destino 80 y sólo el entrante con puerto mayor a 1023
- Solución 2: Permitir sólo el tráfico saliente con puerto destino 80 y sólo el entrante con puerto origen 80.
- Solución 3: Permitir sólo el tráfico saliente con puerto destino 80 y sólo el entrante relacionado con éste.

Firewalls – Iptables (solución 1)

Definición de reglas

Ejemplo Salida a Web: Permitir sólo el tráfico saliente con puerto destino 80 y sólo el entrante con puerto mayor a 1023

```
Iptables -P FORWARD DROP
Iptables -A FORWARD -s 10.0.50.0/24 -p tcp –dport 80 -j ACCEPT
Iptables -A FORWARD -d 10.0.50.0/24 -p tcp –dport 1024:65535 -j ACCEPT
```

Por la política de DROP (primer línea) se desechan todos los paquetes que no coinciden cn ninguna regla. Con la segunda especificamos que se aceptan todos los paquetes con origen igual a nuestra red interna (10.0.50.0) y destino cualquier otro IP con puerto destino TCP 80. Con la tercer línea especificamos que se aceptan todos los paquetes cuyo destino sea nuestra red interna y cuyo origen sea cualquier otra IP. Además con puerto destino TCP mayor o igual a 1024.

Firewalls – Iptables (solución 1)

Definición de reglas

Iptables -P FORWARD DROP

Iptables -A FORWARD -s 10.0.50.0/24 -p tcp –dport 80 -j ACCEPT

Iptables -A FORWARD -d 10.0.50.0/24 -p tcp –dport 1024:65535 -j ACCEPT

Verificación:

Iptables -nL (permite ver las reglas definidas para cada cadena)

Chain INPUT (policy ACCEPT)

Target prot opt source destination

Chain FORWARD (policy DROP)

Target	**prot**	**opt**	**source**	**destination**
ACCEPT	tcp	–	10.0.50.0/24	0.0.0.0/0 tcp dpt:80
ACCEPT	tcp	–	0.0.0.0/0	10.0.50.0/24 tcp dpts:1024: 65535

Chain OUPUT (policy ACCEPT)

Target prot opt source destination

Firewalls – Iptables (solución 2)

Definición de reglas

Ejemplo Salida a Web: Permitir sólo el tráfico saliente con puerto destino 80 y sólo el entrante con puerto origen 80

```
Iptables -P FORWARD DROP
Iptables -A FORWARD -s 10.0.50.0/24 -p tcp –dport 80 -j ACCEPT
Iptables -A FORWARD -d 10.0.50.0/24 -p tcp –sport 80 -j ACCEPT
```

Por la política de DROP (primer línea) se desechan todos los paquetes que no coinciden cn ninguna regla. Con la segunda especificamos que se aceptan todos los paquetes con origen igual a nuestra red interna (10.0.50.0) y destino cualquier otro IP con puerto destino TCP 80. Con la tercer línea especificamos que se aceptan todos los paquetes cuyo destino sea nuestra red interna y cuyo origen sea cualquier otra IP. Además el puerto origen debe ser el 80.

Firewalls – Iptables (solución 2)

Definición de reglas

Iptables -P FORWARD DROP

Iptables -A FORWARD -s 10.0.50.0/24 -p tcp –dport 80 -j ACCEPT

Iptables -A FORWARD -d 10.0.50.0/24 -p tcp –sport 80 -j ACCEPT

Verificación:

Iptables -nL (permite ver las reglas definidas para cada cadena)

Chain INPUT (policy ACCEPT)

Target prot opt source destination

Chain FORWARD (policy DROP)

Target	**prot**	**opt**	**source**	**destination**	
ACCEPT	tcp	–	10.0.50.0/24	0.0.0.0/0	tcp dpt:80
ACCEPT	tcp	–	0.0.0.0/0	10.0.50.0/24	tcp spt:80

Chain OUPUT (policy ACCEPT)

Target prot opt source destination

Firewalls – Iptables (solución 3)

Definición de reglas

Ejemplo Salida a Web: Permitir sólo el tráfico saliente con puerto destino 80 y sólo el entrante establecido

```
Iptables -P FORWARD DROP
Iptables -A FORWARD -s 10.0.50.0/24 -p tcp –dport 80-m state –state NEW -j ACCEPT
Iptables -A FORWARD -m state –state RELATED, ESTABLISHED -j ACCEPT
```

La política es la misma que en los casos anteriores (DROP).Con la segunda volvemos a habilitar la salida al puerto 80 desde la red interna. Con la tercer línea especificamos que se aceptan todos los paquetes con origen distinto a nuestra red interna y destino cualquier otra IP, siempre que el tráfico entrante sea perteneciente a una comunicación ya establecida o esté relacionado con el tráfico de salida.

Firewalls – Iptables (solución 3)

Definición de reglas

Iptables -P FORWARD DROP

Iptables -A FORWARD -s 10.0.50.0/24 -p tcp –dport 80-m state –state NEW -j ACCEPT

Iptables -A FORWARD -m state –state RELATED, ESTABLISHED -j ACCEPT

Verificación:

Iptables -nL -v (permite ver las reglas definidas para cada cadena y los resultados de los paquetes procesados)

Chain INPUT (policy ACCEPT 8567 packets 6433 bytes)

Pkts bytes Target prot opt source destination

Chain FORWARD (policy DROP 0 packets 6433 bytes)

Pkts bytes Target prot opt source destination

0 0 ACCEPT tcp – 10.0.50.0/24 0.0.0.0/0 tcp dpt:80 state NEW

0 0 ACCEPT tcp – 0.0.0.0/0 0.0.0.0/24 state RELATED, ESTABLISHED

Chain OUPUT (policy ACCEPT 9344 packets 1187 bytes)

Target prot opt source destination

Firewalls - Redirecciones

Repaso Estructura

de Iptables

Tabla Filter

- INPUT
- FORWARD
- OUTPUT

Tabla Nat

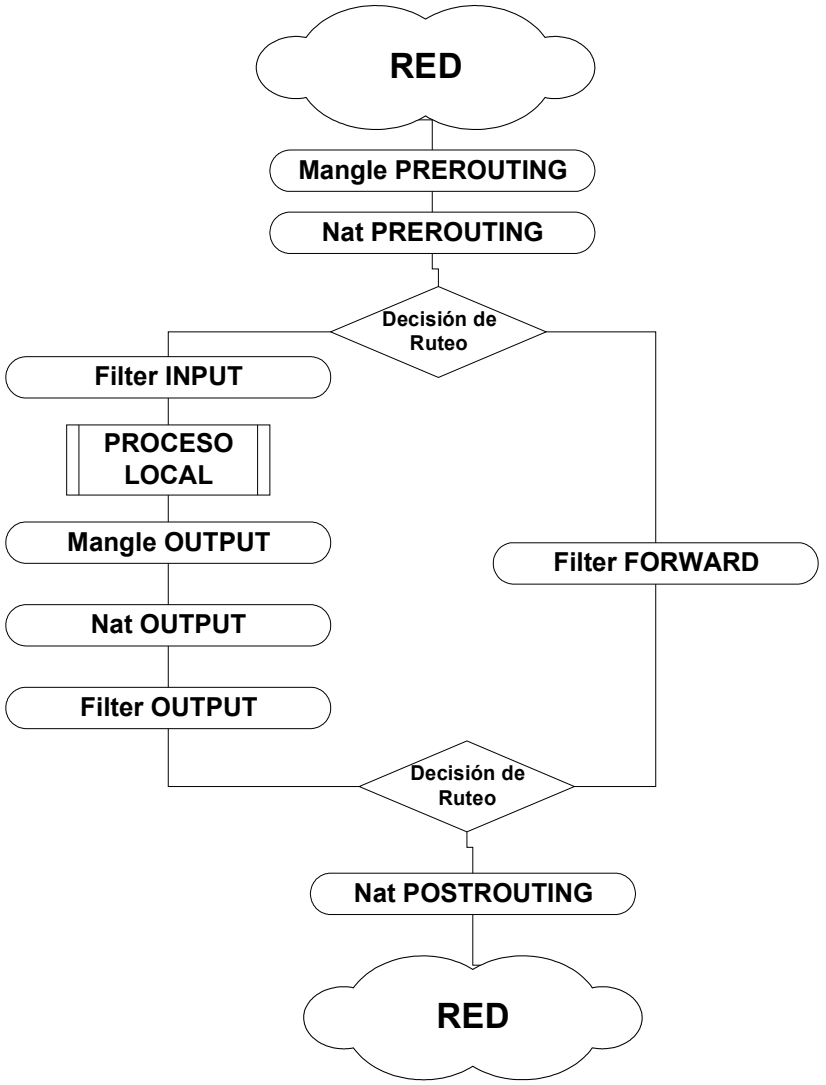
- PREROUTING
- OUTPUT
- POSTROUTING

Tabla Mangle

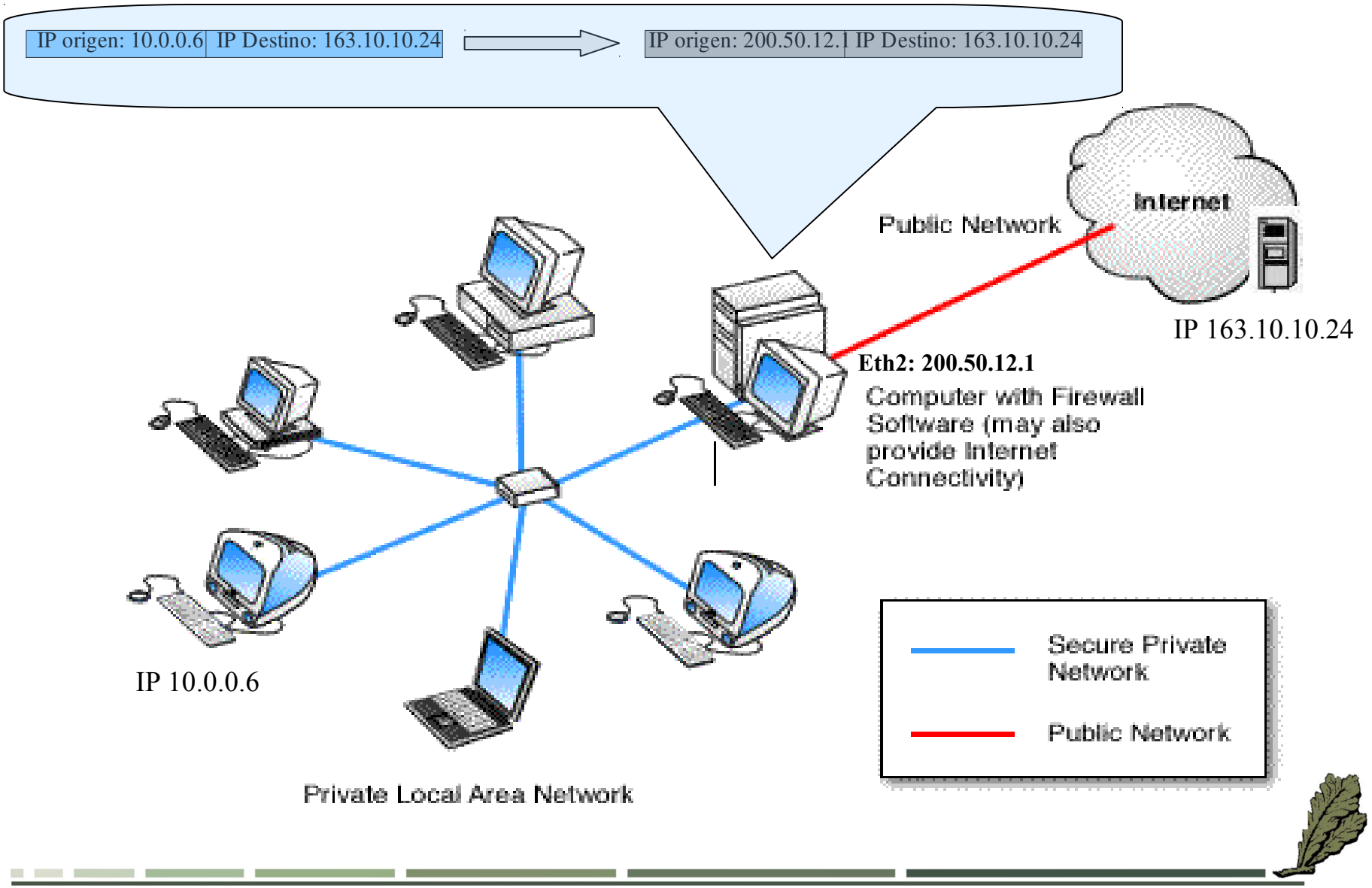
- PREROUTING
- OUTPUT
- POSTROUTING
- INPUT
- FORWARD

Firewalls
Redirecciones

Iptables- Pasando por las tablas



Firewalls - Enmascaramiento



Firewalls - Enmascaramiento

Iptables- Ejemplo: Enmascaramiento

Lo que queremos es poder acceder a los servicios de Internet desde una red interna que utiliza direccionamiento privado. El router/firewall tiene configurada la única IP pública disponible.

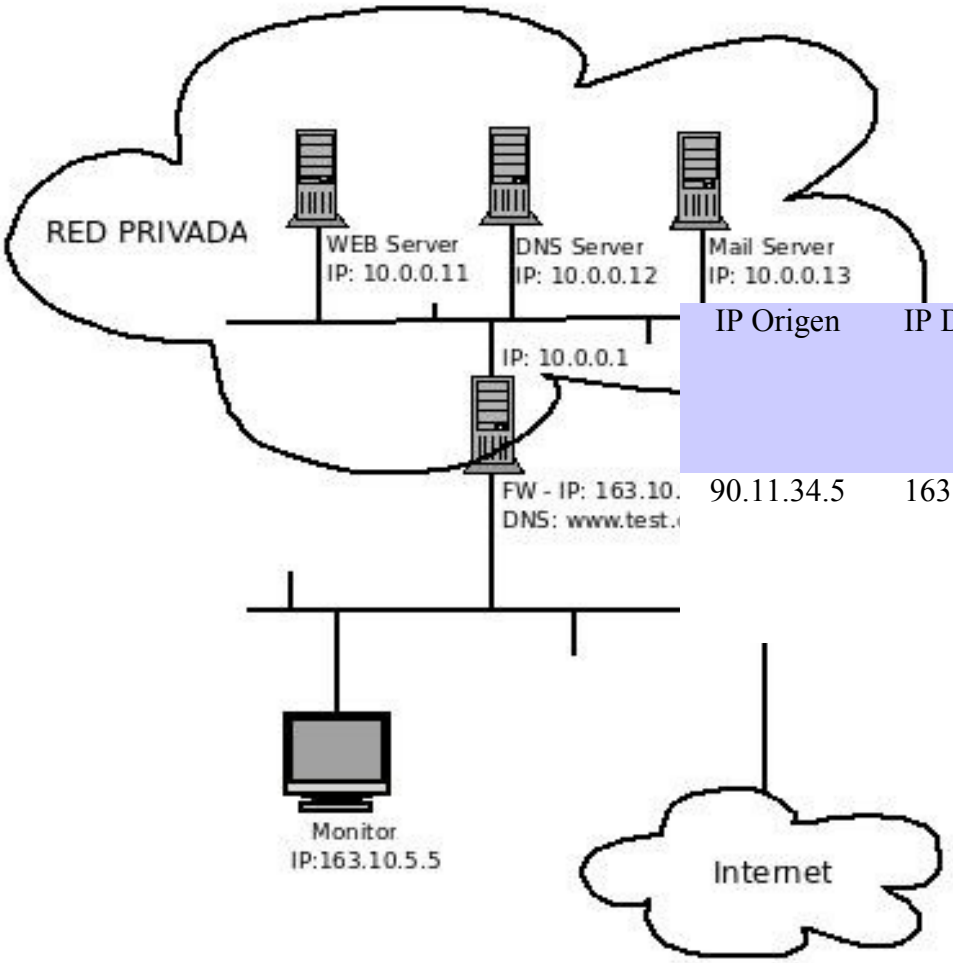
```
iptables -t nat -A POSTROUTING -o eth2 -j MASQUERADE

iptables -nL -t nat
        Chain PREROUTING (policy ACCEPT)
        target     prot opt source               destination

        Chain POSTROUTING (policy ACCEPT)
        target     prot opt source               destination
        MASQUERADE  all  --  0.0.0.0/0           0.0.0.0/0

        Chain OUTPUT (policy ACCEPT)
        target     prot opt source               destination
```

Firewalls - Redirección



IP Origen	IP Destino	-- >>	IP Origen	IP Destino
90.11.34.5	163.10.5.1	-- >>	90.11.34.5	10.0.0.11

Firewalls - Redirección

Queremos redireccionar los requerimientos WEB al servidor web alojado en una red privada. Para que los usuarios en Internet puedan acceder, deben acceder a la IP pública. El firewall debe encargarse de realizar la redirección a la IP interna.

iptables -t nat -A PREROUTING -d 163.10.5.1 -p tcp –dport 80 -j DNAT –to-destination 10.0.0.11

iptables -nL -t nat -v

Chain PREROUTING (policy ACCEPT 406 packets, 71212 bytes)

pkts	bytes	target	prot	opt	in	out	source	destination	
0	0	DNAT	tcp	--	*	*	0.0.0.0/0	163.10.5.1	tcp dpt:80 to:10.0.0.11

Chain POSTROUTING (policy ACCEPT 645 packets, 39825 bytes)

pkts	bytes	target	prot	opt	in	out	source	destination

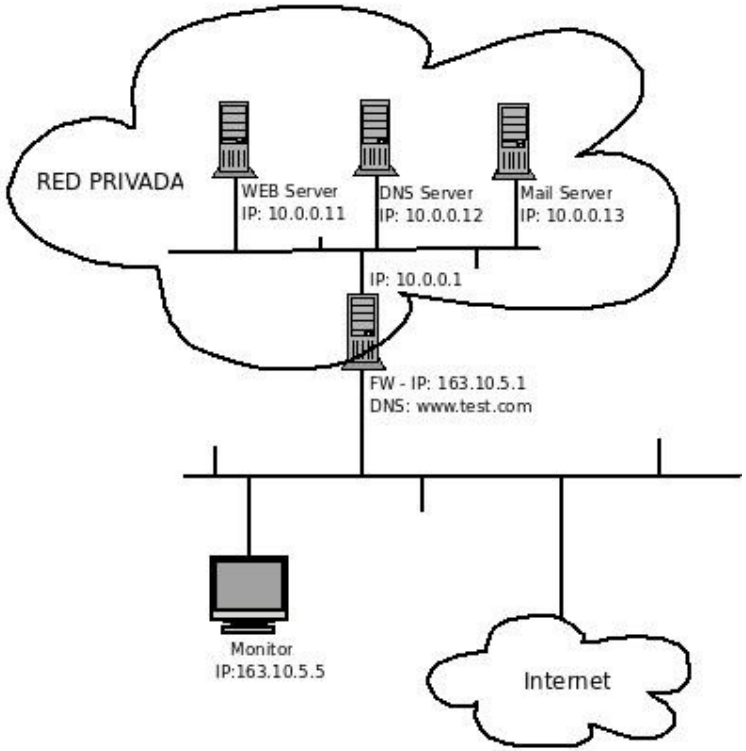
Chain OUTPUT (policy ACCEPT 645 packets, 39825 bytes)

pkts	bytes	target	prot	opt	in	out	source	destination

Firewalls – Otros ejemplos

Queremos que:

- El monitor de red pueda conectarse a cualquier lado pero que no se puedan conectar a él.
- Asegurar el acceso al FW, desde el monitor vía SSH.
- Habilitar las redirecciones
- Habilitar el forwardeo



Reglas en el Monitor

Permitiendo todo al monitor y nada contra éste

```
iptables -P INPUT DROP
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -nL
    Chain INPUT (policy DROP)
    target     prot opt source               destination
    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED

    Chain FORWARD (policy ACCEPT)
    target     prot opt source               destination

    Chain OUTPUT (policy ACCEPT)
    target     prot opt source               destination
```

Firewall

Asegurando el acceso al Firewall

```
iptables -P INPUT DROP
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -s 163.10.5.5 -p tcp --dport 22 -j ACCEPT
```

Habilitando el acceso a los servidores

```
iptables -t nat -A PREROUTING -d 163.10.5.1 -p tcp --dport 80 -j DNAT --to-destination 10.0.0.11

iptables -t nat -A PREROUTING -d 163.10.5.1 -p udp --dport 201 -j DNAT --to-destination 10.0.0.11:161
iptables -t nat -A PREROUTING -d 163.10.5.1 -p udp --dport 202 -j DNAT --to-destination 10.0.0.12:161
iptables -t nat -A PREROUTING -d 163.10.5.1 -p udp --dport 203 -j DNAT --to-destination 10.0.0.13:161
```

Firewall (cont)

Restringiendo el tráfico que puede pasar por el firewall

```
iptables -P FORWARD DROP
iptables -A FORWARD -d 10.0.0.0/24 -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -s 163.10.5.5 -p udp --dport 161 -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -j LOG
```

Firewall (cont)

Configuración final de la tabla NAT

```
iptables -nL -t nat
    Chain PREROUTING (policy ACCEPT)
    target     prot opt source               destination
    DNAT       tcp  --  0.0.0.0/0            163.10.5.1          tcp dpt:80 to:10.0.0.11
    DNAT       udp  --  0.0.0.0/0            163.10.5.1          udp dpt:201 to:10.0.0.11:161
    DNAT       udp  --  0.0.0.0/0            163.10.5.1          udp dpt:202 to:10.0.0.12:161
    DNAT       udp  --  0.0.0.0/0            163.10.5.1          udp dpt:203 to:10.0.0.13:161

    Chain POSTROUTING (policy ACCEPT)
    target     prot opt source               destination

    Chain OUTPUT (policy ACCEPT)
    target     prot opt source               destination
```

Firewall (cont)

Configuración final de la tabla FILTER

```
iptables -nL
    Chain INPUT (policy DROP)
    target     prot opt source               destination
    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    ACCEPT     tcp  --  163.10.5.5           0.0.0.0/0           tcp dpt:22

    Chain FORWARD (policy DROP)
    target     prot opt source               destination
    ACCEPT     udp  --  163.10.5.5           0.0.0.0/0           udp dpt:161
    ACCEPT     tcp  --  0.0.0.0/0            10.0.0.0/24         tcp dpt:80
    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    LOG        all  --  0.0.0.0/0            0.0.0.0/0           LOG flags 0 level 4

    Chain OUTPUT (policy ACCEPT)
    target     prot opt source               destination
```

Buenas prácticas para un diseño de red seguro

Las siguientes pueden ser buenas medidas para realizar un diseño seguro de la topología de red.

- Definir una DMZ para alojar a los servidores públicos.
- Definir una red separada para los servidores internos, para evitar que estén en la misma red que los usuarios.

Buenas prácticas para un diseño de red seguro

- Evitar el contacto directo de la red de usuarios con Internet, para ello, maximizar el uso de firewalls de aplicación para las aplicaciones que los usuarios necesiten (por ejemplo proxy web con antivirus).
- Restringir el flujo entre las distintas redes que separa el firewall (DMZs, red interna, Internet) a sólo el mínimo necesario.

Buenas prácticas para un diseño de red seguro

- Loguear el tráfico que intenta violar nuestra política de firewall, especialmente el que se origina en las DMZs y la red interna de la organización (Extrusion detection).
- Configurar el acceso de redes wireless a una red separada con los controles de accesos definidos.
- Ubicar el servidor de VPN en una DMZ distinta a la DMZ de servidores para el acceso remoto de usuarios.
- Configurar IDSs en todos los segmentos de red.

Referencias

https://datatracker.ietf.org/doc/rfc2979/

www.netfilter.org/

http://www.faqs.org/docs/iptables/