Seguridad en Sistemas Informáticos (SSI)

Seguridad perimétrica

Carlos Pérez Conde

Departament d'Informàtica

Escola Técnica Superior d'Enginyeria

Universitat de València

Guión

- Concepto de cortafuegos
- Filtrado de paquetes
- Proxies
- Diseño de cortafuegos
- Integración de VPNs
- Detección de intrusos

Guión

- **Concepto de cortafuegos**
- Filtrado de paquetes
- Proxies
- Diseño de cortafuegos
- Integración de VPNs
- Detección de intrusos

Modelos de seguridad

- **Seguridad basada en el nodo**
  - controlar el acceso y la utilización de cada nodo
  - en cada máquina (nodo) se aplican las medidas de seguridad apropiadas (ver tema: "Seguridad centrada en el nodo")
  - no es escalable:
    - consume muchos recursos
    - más difícil con sistemas heterogéneos
    - diferentes configuraciones implican problemas de seguridad diferentes
    - la seguridad también depende de los usuarios (ej: contraseñas débiles)
- **Seguridad basada en la red**
  - controlar el acceso en los puntos de conexión con otras redes
  - medidas de seguridad a nivel de red (a abordar en este tema)
- **Ningún modelo es suficiente por sí solo**

¿Qué es un cortafuegos?

- **Un sistema informático que actúa como separador permeable**
  - analiza y filtra el tráfico de red de fuera a dentro
  - y de dentro a fuera
- **¿Para qué sirve un cortafuegos?**
  - como punto donde focalizar las decisiones de seguridad
  - para forzar el cumplimiento de una política de seguridad
    - de acceso a otras redes (ej: Internet)
    - de acceso desde otras redes (ej: Internet)
  - para registrar los accesos a/desde otras redes
  - para limitar los daños (ej: aislando subredes internas)
  - para ganar tiempo

¿Qué es un cortafuegos?

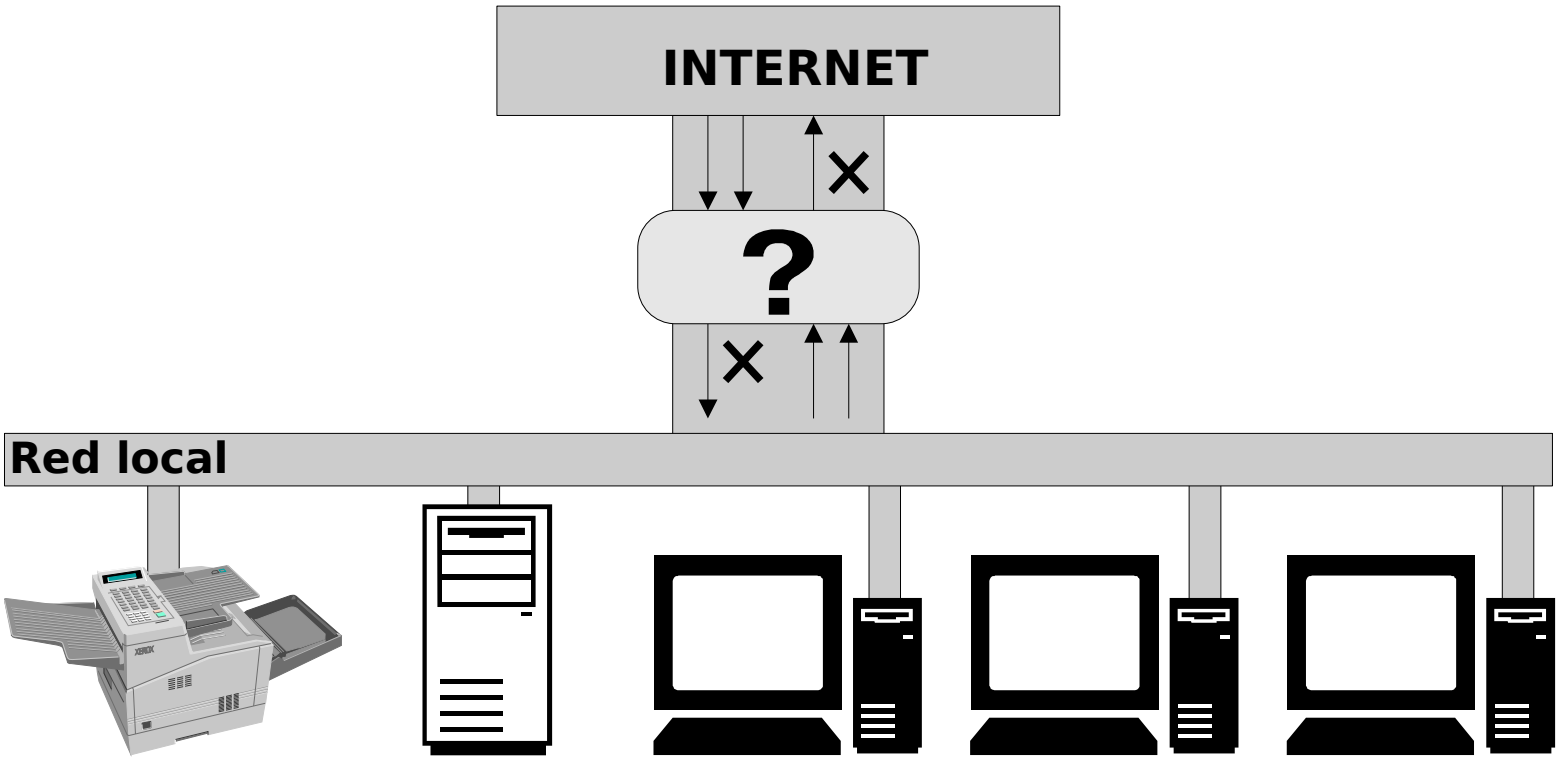
- **¿Para qué no sirve un cortafuegos?**
  - para proteger las máquinas frente a usuarios internos maliciosos
  - para proteger la red frente a puertas traseras (ej: un módem de un ordenador conectado a la red interna)
  - para proteger frente a ataques no conocidos
  - para proteger completamente frente a virus (ej: adjunto encriptado)
  - un cortafuegos no puede configurarse solo
  - un cortafuegos no puede mantenerse solo

Guión

- Concepto de cortafuegos
- **Filtrado de paquetes**
- Proxies
- Diseño de cortafuegos
- Integración de VPNs
- Detección de intrusos

Filtrado de paquetes

- **Cortafuegos (FW) = encaminador selectivo**
  - examina las los paquetes que recibe
  - aplica unas reglas que deben reflejar la política de seguridad
  - decide si el paquete puede atravesar el cortafuegos o no



¿Qué tiene en cuenta el FW?

- **La cabecera de los paquetes:**
  - direcciones IP, protocolo, puertos, tipo de mensaje ICMP, tamaño
- **Contenido del paquete (algunos cortafuegos)**
  - paquetes conformes al protocolo, direcciones
- **Interfaces de red de entrada y salida**
- **Registro de paquetes procesados**
  - ¿es una respuesta a otro paquete previo?
  - ¿cuántos paquetes han llegado desde este mismo nodo en un cierto intervalo de tiempo?
  - ¿es una repetición de un paquete anterior?
  - ¿es un fragmento de un paquete mayor?

¿Qué acciones puede realizar?

- **Cortafuegos sencillos**
  - enviar el paquete a su destino
  - descartar el paquete
  - rechazar el paquete, devolviendo error al emisor
  - registrar información sobre el paquete
  - notificar la llegada del paquete
- **Cortafuegos sofisticados**
  - modificar el paquete (ej: NAT)
  - desviar el paquete (ej: equilibrado de cargas, alta disponibilidad...)
  - alterar las reglas de filtrado (ej: aceptar una respuesta a un paquete, descartar todo el tráfico procedente de un nodo hostil)

Ventajas e inconvenientes del filtrado de paquetes

- **Ventajas del filtrado de paquetes**
  - un cortafuegos que filtre paquetes puede proteger a toda una red
  - un filtrado sencillo es extremadamente eficiente
  - disponible en una gran cantidad de encaminadores
- **Desventajas del filtrado de paquetes**
  - las herramientas actuales no son perfectas
    - son difíciles de configurar
    - son difíciles de comprobar
    - es posible que un fallo permita el paso de paquetes no autorizados
  - pueden ralentizar notablemente al encaminador
    - sobrecarga mayor cuanto más complejas son las reglas
    - pueden impedir optimizaciones para encaminar
  - no pueden implementar todas las políticas de seguridad
    - ej: no se pueden rechazar paquetes de un usuario concreto

Configuración de filtros

- **Registro de paquetes**
  - especialmente importante para paquetes rechazados
  - cuidado con ataques de denegación de servicio
- **Devolución de códigos de error**
  - avisa al emisor del paquete
    - evita reintentos
    - facilita la recolección de información sobre la red
  - sobrecarga el encaminador
  - recomendación por defecto:
    devolver códigos de error a las redes internas exclusivamente

Recomendaciones de filtrado

- Rechazar por defecto
- Reensamblar los fragmentos antes de introducirlos en la red interna
- Rechazar paquetes con encaminamiento establecido en la fuente (*source routing*)
- Rechazar paquetes ICMP con tamaño mayor que unos pocos KB
- Rechazar los paquetes con direcciones incorrectas

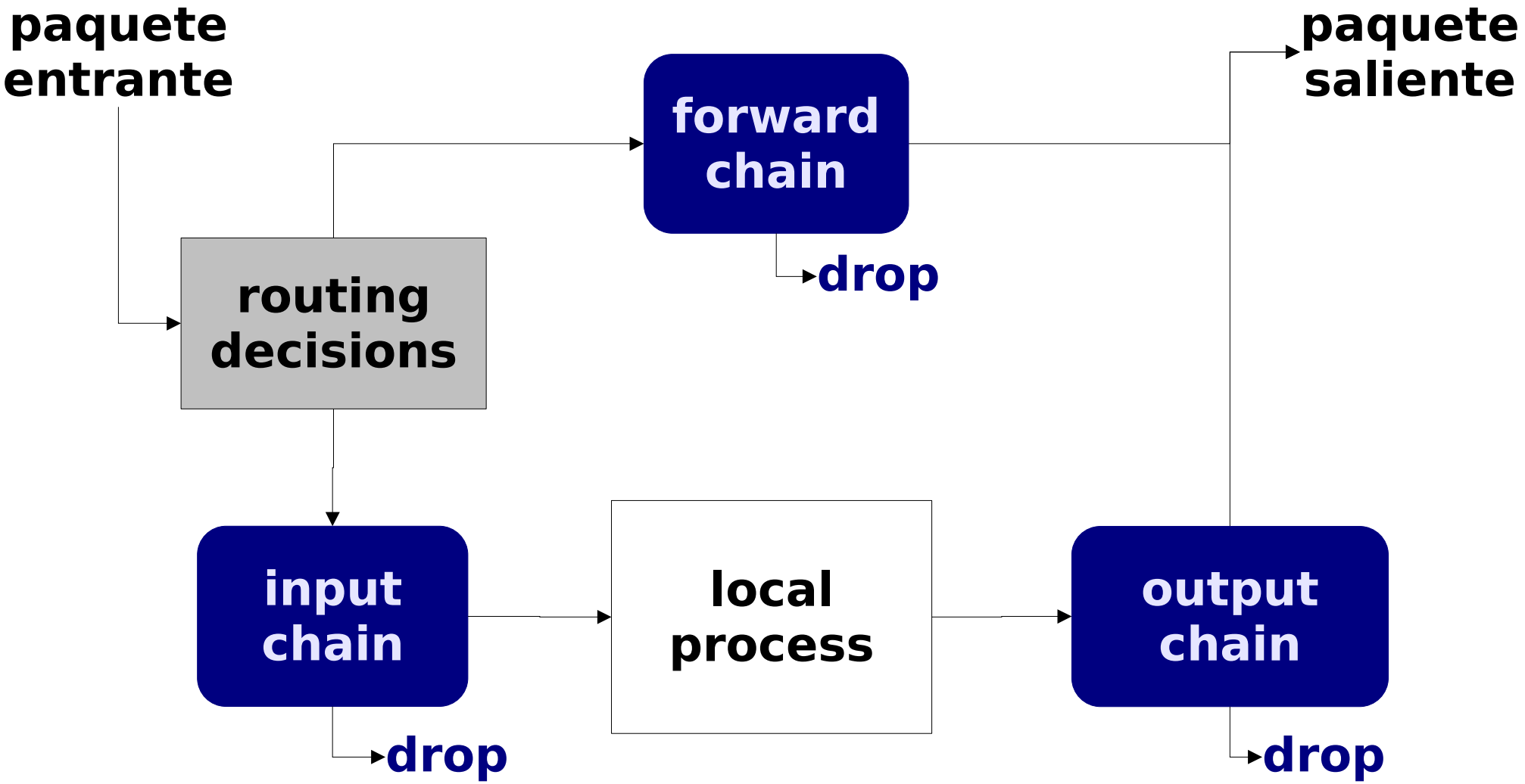
Ejemplos de direcciones incorrectas

- **con dirección de origen inválida**
  - entrantes con direcciones de origen internas
  - salientes con direcciones de origen externas
  - la de la interfaz externa del cortafuegos
  - las de la interfaz de bucle invertido (*loopback*): 127.0.0.0/8
  - la dirección de destino para difusión: 255.255.255.255
- **con dirección de destino inválida**
  - la dirección de origen para difusión: 0.0.0.0
- **reservadas**
  - para redes privadas: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16
  - para investigación: 240.0.0.0/5
  - por el IANA: http://www.iana.org/assignments/ipv4-address-space

Caso de ejemplo: netfilter

- **Netfilter/iptables para Linux 2.4.x, 2.6.x**
  - http://www.netfilter.org
  - Recomendado: “Iptables tutorial”, Oskar Andreasson, http://iptables-tutorial.frozentux.net/iptables-tutorial.html
- **Cadenas (chains)**
  - grupos de reglas a aplicar a los paquetes la atraviesan
- **Tablas**
  - grupos de cadenas con un propósito específico
  - RAW, NAT, MANGLE, FILTER
- **FILTER: tabla de filtrado**
  - 3 cadenas básicas: INPUT, OUTPUT, FORWARD

FILTER: la tabla de filtrado



Configuración básica

Permitiendo el tráfico sólo a través de la interfaz de loopback

```
# iptables -F
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

# for x in INPUT OUTPUT FORWARD; do iptables -P $x DROP; done
# iptables -A INPUT -i lo -j ACCEPT
# iptables -A OUTPUT -o lo -j ACCEPT

# iptables -L -v
Chain INPUT (policy DROP 2 packets, 266 bytes)
 pkts bytes target     prot opt in     out     source     destination
    0     0 ACCEPT     all  --  lo     any     anywhere   anywhere
Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source     destination
Chain OUTPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source     destination
    0     0 ACCEPT     all  --  any    lo      anywhere   anywhere
```

Configuración básica

Permitiendo temporalmente el tráfico entre 147.156.17.80 y este nodo

```
# iptables -A INPUT -s 147.156.17.80 -j ACCEPT
# iptables -A OUTPUT -d 147.156.17.80 -j ACCEPT
# iptables -L -v
Chain INPUT (policy DROP 438 packets, 47173 bytes)
 pkts bytes target     prot opt in     out  source          destination
    0     0 ACCEPT     all  --  lo     any  anywhere        anywhere
   32  4607 ACCEPT     all  --  any    any  147.156.17.80   anywhere

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out  source     destination

Chain OUTPUT (policy DROP 104 packets, 7519 bytes)
 pkts bytes target     prot opt in     out  source     destination
    7   784 ACCEPT     all  --  any    lo   anywhere   anywhere
   32  4526 ACCEPT     all  --  any    any  anywhere   147.156.17.80

# iptables -D OUTPUT 2
# iptables -L OUTPUT
Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
```

Configuración adicional

Registrando las conexiones entrantes desde 147.156.17.80

```
# iptables -I INPUT -s 147.156.17.80 -p tcp --syn -j LOG

# tail /var/log/messages | grep "kernel:"
Mar 21 09:07:19 posets kernel: IN=eth1 OUT=
MAC=00:e0:18:af:a7:62:00:c0:ca:14:8c:08:08:00 SRC=147.156.17.80
DST=147.156.17.45 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=23983 DF PROTO=TCP
SPT=32791 DPT=22 WINDOW=5840 RES=0x00 SYN URGP=0
```

Filtrado de Telnet

- **Características**
  - los servidores escuchan en los puertos 23 (telnet)
  - el cliente utiliza un puerto no privilegiado (> 1023)
- **Recomendaciones**
  - restringir al máximo las conexiones entrantes (mejor SSH)
  - las conexiones de salida son seguras (para la red interna)
  - si el acceso a los datos debe ser restringido es mejor usar SSH

Filtrado de telnet

Sentido	D. Orig.	D. Dest.	Prot.	P. Orig.	P.Dest	ACK
entrada	externa	interna	TCP	>1023	23	!1, sí resto
salida	interna	externa	TCP	23	>1023	sí
salida	interna	externa	TCP	>1023	23	!1, sí resto
entrada	externa	interna	TCP	23	>1023	sí

Filtrado de SSH

- **Características:**
  - los servidores escuchan en el puerto 22
  - el cliente utiliza un puerto no privilegiado (> 1023)
    salvo cuando la autentificación está basada en rhosts (no recomendable)

Filtrado de SSH

Sentido	D. Orig.	D. Dest.	Prot.	P. Orig.	P.Dest	ACK
entrada	externa	interna	TCP	>1023 (*)	22	!1, sí resto
salida	interna	externa	TCP	22	>1023 (*)	sí
salida	interna	externa	TCP	>1023 (*)	22	!1, sí resto
entrada	externa	interna	TCP	22	>1023 (*)	sí

Filtrado de SSH

- **Recomendaciones**
  - no permitir autentificación basada en rhosts
  - deshabilitar, si se puede, la redirección de puertos y de X11
  - permitir conexiones entrantes sólo a servidores controlados (no administrados por usuarios)
  - deshabilitar, si se puede, conexiones salientes (posible mal uso de la redirección de puertos)

Filtrado de SSH

Sentido	D. Orig.	D. Dest.	Prot.	P. Orig.	P.Dest	ACK
entrada	externa	interna	TCP	>1023 (*)	22	!1, sí resto
salida	interna	externa	TCP	22	>1023 (*)	sí
salida	interna	externa	TCP	>1023 (*)	22	!1, sí resto
entrada	externa	interna	TCP	22	>1023 (*)	sí

Ejemplo: Netfilter (iptables)

Configuración de telnet entrante:

```
iptables -A INPUT -p tcp -s 0/0 --sport 1024: -d 147.156.17.29 --dport 23 -j ACCEPT
iptables -A OUTPUT -p tcp -s 147.156.17.29 --sport 23 -d 0/0 --dport 1024: ! --syn -j ACCEPT
```

Configuración de telnet saliente:

```
iptables -A OUTPUT -p tcp -s 147.156.17.29 --sport 1024: -d 0/0 --dport 23 -j ACCEPT
iptables -A INPUT -p tcp -s 0/0 --sport 23 -d 147.156.17.29 --dport 1024: ! --syn -j ACCEPT
```

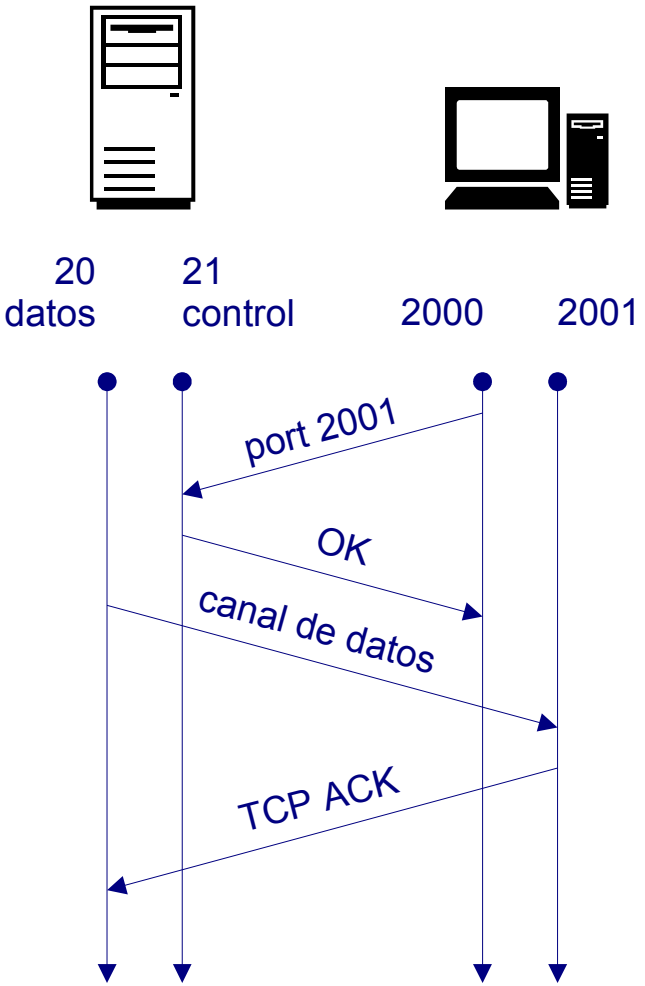
Configuración de SSH entrante:

```
iptables -A INPUT -p tcp -s 0/0 --sport 1024: -d 147.156.17.29 --dport 22 -j ACCEPT
iptables -A OUTPUT -p tcp -s 147.156.17.29 --sport 22 -d 0/0 --dport 1024: ! --syn -j ACCEPT
```

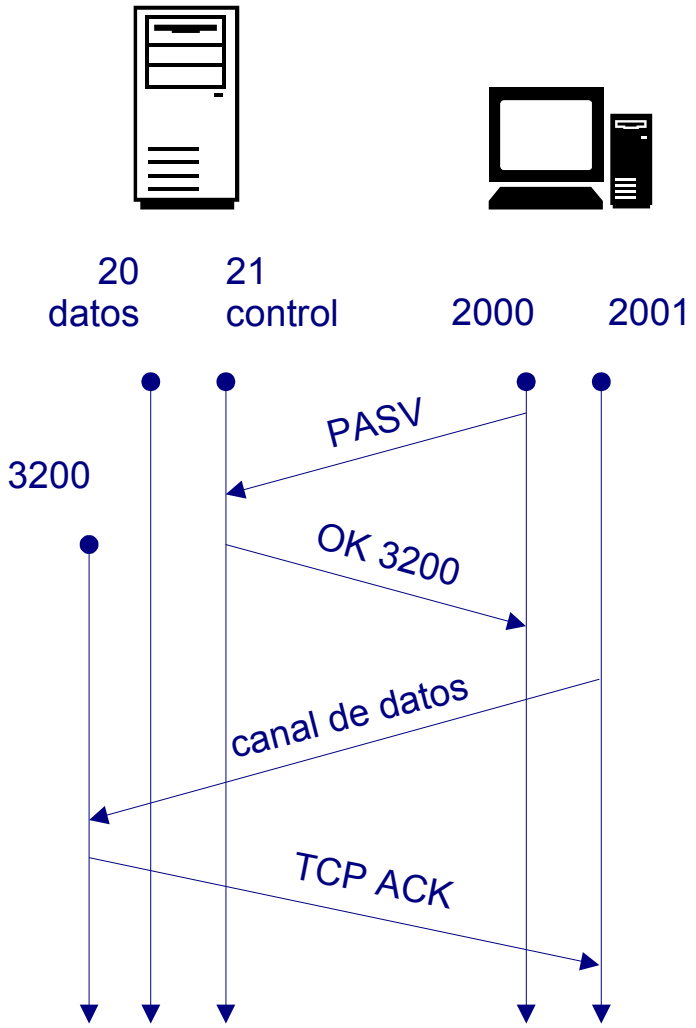
Configuración de SSH saliente:

```
iptables -A OUTPUT -p tcp -s 147.156.17.45 --sport 1024: -d 0/0 --dport 22 -j ACCEPT
iptables -A INPUT -p tcp -s 0/0 --sport 22 -d 147.156.17.45 --dport 1024: ! --syn -j ACCEPT
```

Filtrado de FTP (I)



Conexión FTP normal



Conexión FTP pasiva

Filtrado de FTP entrante

- **Permitirlo exclusivamente al nodo bastión**
- **Utilizar un servidor lo más actualizado posible**
- **FTP anónimo:**
  - limitar el acceso a datos públicos (ej: chroot)
  - evitar que otros lo utilicen para distribuir sus datos
  - evitar que sea utilizado para atacar a otras máquinas

Sentido	D. Orig.	D. Dest.	Prot.	P. Orig.	P.Dest	ACK
entrada	externa	interna	TCP	>1023	21	!1, sí resto
salida	interna	externa	TCP	21	>1023	sí
salida	interna	externa	TCP	20	>1023	!1, sí resto
entrada	externa	interna	TCP	>1023	20	sí
entrada	externa	interna	TCP	>1023	>1023	!1, sí resto
salida	interna	externa	TCP	>1023	>1023	sí

Filtrado de FTP saliente

- **Modo normal**
  - exige permitir conexiones desde fuera (desde el puerto 20)
- **Modo pasivo**
  - exige permitir conexiones hacia fuera (puertos >1023)
  - los clientes deben soportarlo
  - posibles problemas de compatibilidad cliente/servidor

Sentido	D. Orig.	D. Dest.	Prot.	P. Orig.	P.Dest	ACK
salida	interna	externa	TCP	>1023	21	!1, sí resto
entrada	externa	interna	TCP	21	>1023	Sí
entrada	externa	interna	TCP	20	>1023	!1, sí resto
salida	interna	externa	TCP	>1023	20	sí
salida	interna	externa	TCP	>1023	>1023	!1, sí resto
entrada	externa	interna	TCP	>1023	>1023	sí

Filtrado de HTTP

- **Características**
  - el servidor escucha en el puerto 80 (no necesariamente)
  - el cliente utiliza puertos no privilegiados (>1023)
- **Recomendaciones para un servidor HTTP:**
  - utilizar el puerto estándar
  - emplear un nodo bastión dedicado
  - controlar con cuidado los programas y ficheros a los que puede acceder el servidor

Sentido	D. Orig.	D. Dest.	Prot.	P. Orig.	P.Dest	ACK
entrada	externa	interna	TCP	>1023	80 (hab.)	!1, sí resto
salida	interna	externa	TCP	80 (hab.)	>1023	sí
salida	interna	externa	TCP	>1023	80 (hab.)	!1, sí resto
entrada	externa	interna	TCP	80 (hab.)	>1023	sí

Filtrado de HTTP

- **Recomendaciones para clientes HTTP**
  - configurarlos cuidadosamente
  - educar a los usuarios para que no cambien la configuración basándose en consejo externo

Sentido	D. Orig.	D. Dest.	Prot.	P. Orig.	P.Dest	ACK
entrada	externa	interna	TCP	>1023	80 (hab.)	!1, sí resto
salida	interna	externa	TCP	80 (hab.)	>1023	sí
salida	interna	externa	TCP	>1023	80 (hab.)	!1, sí resto
entrada	externa	interna	TCP	80 (hab.)	>1023	sí

Filtrado de DNS

- **Características**
  - el servidor escucha en los puertos 53 de **TCP y UDP**
  - los clientes usan puertos no privilegiados (>1023)
  - los servidores contactan entre sí:
    - para resolver consultas de los clientes
    - para realizar transferencias de zona

Sentido	D. Orig.	D. Dest.	Prot.	P. Orig.	P.Dest	ACK
entrada	externa	interna	UDP	>1023	53	
salida	interna	externa	UDP	53	>1023	
entrada	externa	interna	TCP	>1023	53	!1, sí resto
salida	interna	externa	TCP	53	>1023	sí
salida	interna	externa	UDP	>1023	53	
entrada	externa	interna	UDP	53	>1023	
salida	interna	externa	TCP	>1023	53	!1, sí resto
entrada	externa	interna	TCP	53	>1023	sí

Filtrado de DNS

- **Características**
  - el servidor escucha en los puertos 53 de **TCP y UDP**
  - los clientes usan puertos no privilegiados (>1023)
  - los servidores contactan entre sí:
    - para resolver consultas de los clientes
    - para realizar transferencias de zona

Sentido	D. Orig.	D. Dest.	Prot.	P. Orig.	P.Dest	ACK
entrada	externa	interna	UDP	>1023	53	
salida	interna	externa	UDP	53	>1023	
salida	interna	externa	UDP	>1023	53	
entrada	externa	interna	UDP	53	>1023	
entrada	externa	interna	UDP	53	53	
salida	interna	externa	UDP	53	53	
entrada	externa	interna	TCP	>1023	53	!1, sí resto
salida	interna	externa	TCP	53	>1023	sí
salida	interna	externa	TCP	>1023	53	!1, sí resto
entrada	externa	interna	TCP	53	>1023	sí

Filtrado de DNS

- **Recomendaciones**
  - proporcionar servicio de DNS al exterior desde un nodo bastión
  - no permitir el acceso desde el exterior a información adicional (ej: registros HINFO)
  - utilizar una versión actual de BIND y utilizar consultas dobles (double-reverse lookups)
  - deshabilitar las transferencias de zona (excepto a los servidores secundarios propios)

Filtrado con estado (*stateful filtering*)

- **El CF mantiene una tabla con conexiones establecidas**
- **TCP**
  - se aprovechan los indicadores (*flags*) de los paquetes
  - temporizador para conexiones no terminadas (ej: DOS)
- **UDP**
  - se utilizan las direcciones y puertos de origen y destino
  - temporizador para todas las conexiones
- **ICMP**
  - se usan las direcciones y los tipos de paquete (para petición/respuesta; ej: ping)
  - asociación con “conexiones” TCP/UDP

Filtrado con estado de HTTP

Configurando iptables para que registre las conexiones salientes

```
# iptables -F
# iptables -A OUTPUT -m state --state NEW -p tcp --dport 80 -j LOG
# iptables -L OUTPUT
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
LOG        tcp  --  anywhere             anywhere            state NEW tcp
dpt:www LOG level warning
#
```

Comprobando el resultado

```
# wget 147.156.1.4 >/dev/null 2>&1
# tail -1 /var/log/messages
Dec 29 11:04:49 posets kernel: [17782280.204000] IN= OUT=eth0
SRC=147.156.13.171 DST=147.156.1.4 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=31245 DF
PROTO=TCP SPT=51822 DPT=80 WINDOW=5840 RES=0x00 SYN URGP=0
# cat /proc/net/ip_conntrack | grep 147.156.1.4
tcp      6 118 TIME_WAIT src=147.156.13.171 dst=147.156.1.4 sport=51823
dport=80 packets=10 bytes=729 src=147.156.1.4 dst=147.156.13.171 sport=80
dport=51823 packets=14 bytes=13797 [ASSURED] mark=0 use=1
#
```

Filtrado con estado y contenido

- **El CF mantiene una tabla con conexiones establecidas**
- **Y analiza el contenido de ciertos protocolos**
- **Ejemplo: FTP**
  - detecta la orden PORT del cliente
  - y añade a la tabla la entrada correspondiente a la conexión de datos

Filtrado de FTP activo

Registrando la conexión de datos de FTP saliente

```
# iptables -F
# modprobe ip_conntrack_ftp
# iptables -A INPUT -m state --state RELATED -p tcp --sport 20 -j LOG
#
```

Comprobando el resultado

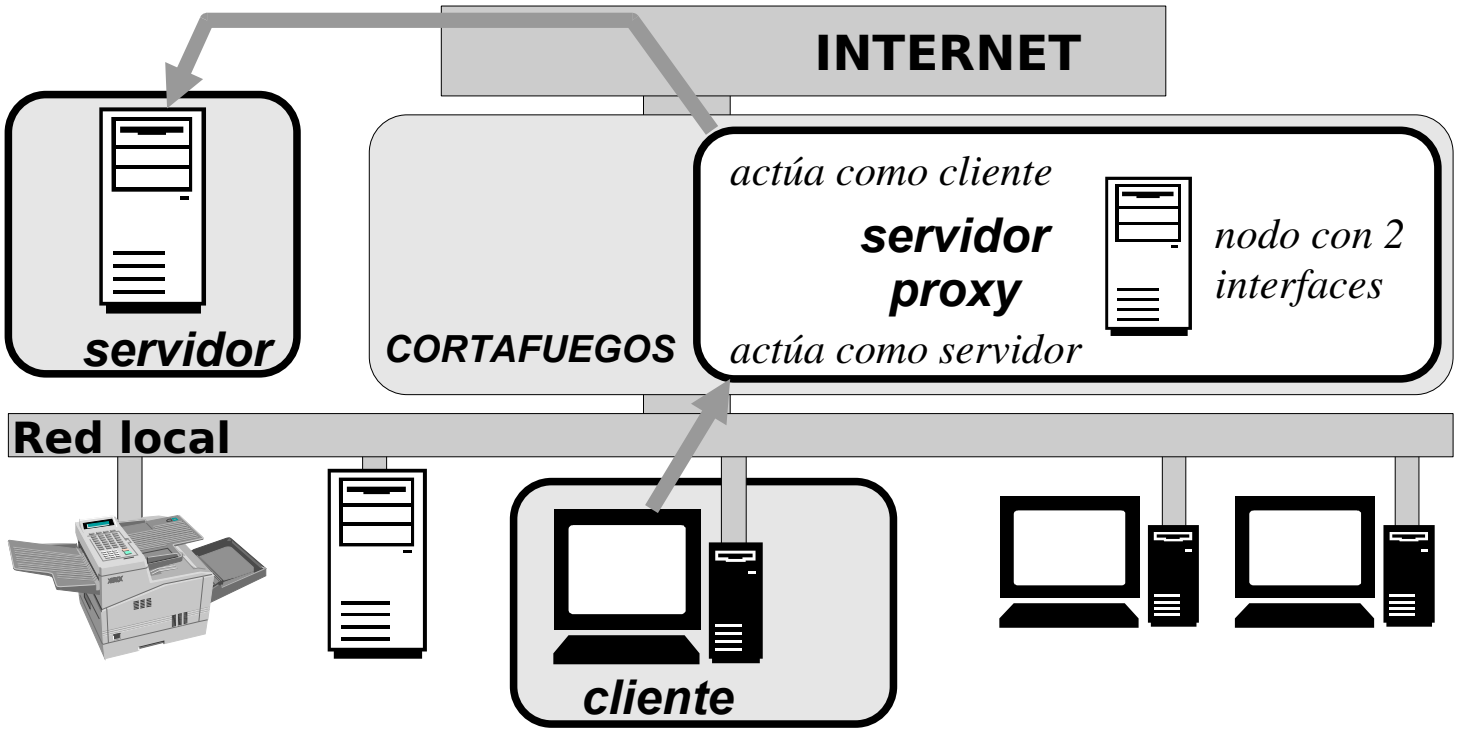
```
# wget --no-passive-ftp ftp://147.156.1.14 >/dev/null 2>&1
# tail -1 /var/log/messages
Dec 29 11:35:26 posets kernel: [17784117.200000] IN=eth0 OUT=
MAC=00:0d:61:19:0d:91:00:0b:bf:27:80:00:08:00 SRC=147.156.1.14
DST=147.156.13.171 LEN=60 TOS=0x00 PREC=0x00 TTL=63 ID=40497 DF PROTO=TCP
SPT=20 DPT=39157 WINDOW=5840 RES=0x00 SYN URGP=0
# cat /proc/net/ip_conntrack | grep 147.156.1.14
tcp      6 117 TIME_WAIT src=147.156.13.171 dst=147.156.1.14 sport=38547
dport=21 packets=14 bytes=818 src=147.156.1.14 dst=147.156.13.171 sport=21
dport=38547 packets=16 bytes=1380 [ASSURED] mark=0 use=2
tcp      6 117 TIME_WAIT src=147.156.1.14 dst=147.156.13.171 sport=20
dport=47411 packets=5 bytes=1343 src=147.156.13.171 dst=147.156.1.14
sport=47411 dport=20 packets=3 bytes=164 [ASSURED] mark=0 use=1
#
```

Guión

- Concepto de cortafuegos
- Filtrado de paquetes
- **Proxies**
- Diseño de cortafuegos
- Integración de VPNs
- Detección de intrusos

Proxies

- **Proxies**
  - representantes locales de un servicio remoto
  - pasarelas de nivel de aplicación, *application-level gateways)*
  - el servicio remoto sólo debe resultar accesible a través del proxy



Ventajas y desventajas

- **Ventajas de los proxies**
  - pueden registrar selectivamente información detallada
  - pueden mantener cachés
  - pueden filtrar inteligentemente (ej: impedir el uso de Java o Javascript)
  - pueden autentificar a los usuarios, tratándolos de forma diferenciada
  - protegen a los clientes frente a paquetes mal formados
- **Desventajas de los proxies**
  - los servicios con proxy van a la zaga de los servicios sin proxy
  - pueden requerir proxies diferentes para cada servicio
  - suelen requerir modificaciones a los clientes, las aplicaciones o los procedimientos

¿Cómo hacer que se use el proxy?

- Aplicaciones preparadas para usar proxies
- Sistemas operativos preparados para usar proxies
- Procedimientos de usuario modificados
- Encaminador preparado para usar proxies

Aplicaciones preparadas para usar Proxies

- **El SW sabe cómo contactar con el proxy y cómo decirle con qué servidor desea conectarse**
- **Inconvenientes**
  - el sw puede no estar disponible en todas las plataformas en las que se tenga que utilizar
  - los clientes disponibles pueden no ser populares entre los usuarios (excepto los navegadores Web; todos soportan proxies)
  - es necesario configurar adecuadamente las aplicaciones (participación del usuario)
  - se puede acabar usando sw diferente para las conexiones internas y para las externas
- **La situación tiende a mejorar**

Sistemas Operativos preparados para usar Proxies

- **La aplicación intenta conectarse al servidor y el SO la redirige**
  - a nivel de librería (posible con librerías dinámicas)
  - a nivel del núcleo del SO (más complejo)
- **Inconvenientes**
  - la existencia de aplicaciones enlazadas estáticamente
  - sw con librerías dinámicas para el acceso a la red propias
  - aplicaciones que trabajen a bajo nivel, sin usar las librerías
  - protocolos con nº de puerto o direcciones IP incluidas en los datos
  - cuando algo falla, el usuario no sabe qué está ocurriendo
- **Ventaja**
  - es transparente para el usuario

Procedimientos de Usuario Modificados

- **El usuario le indica al cliente que se conecte al proxy, y al proxy que se conecte al servidor**
- **Inconvenientes**
  - el cliente debe saber qué proxy usar y cómo indicarle a qué servidor conectarse
  - los clientes pueden no permitir esta conexión intermedia
    (ej: un cliente de ftp que automáticamente intente una conexión anónima)
  - hay que educar adecuadamente a todos los usuarios (depende del tipo de usuarios), yendo en contra de todas las demás fuentes de información (libros, www...)

Encaminador preparado para usar Proxies

- **Los clientes intentan conectarse al servidor y el encaminador redirige los paquetes al proxy**
- **Ventajas**
  - las del filtrado de paquetes (transparencia)
  - las de los proxies (ej: se pueden mantener caches)
- **Inconvenientes**
  - las del filtrado de paquetes
    - un error puede permitir conexiones directas no autorizadas
    - no se puede utilizar con protocolos que no funcionen con filtrado de paquetes
    - todos los nodos internos deben poder traducir los nombres de nodos externos en direcciones, para permitir que intenten conectarse a ellos
  - las de los proxies
    - no se puede utilizar con protocolos que no funcionen con proxies

Algunos aspectos adicionales

- **Proxies a nivel de aplicación/a nivel de circuito**
  - nivel de aplicación
    - conoce la aplicación, es decir, las órdenes del protocolo de aplicación (ej: FTP)
    - ej: sendmail (protolo de almacenar y reenviar -*store and forward*-)
  - nivel de circuito
    - establece un circuito entre cliente y servidor (sin interpretar las órdenes)
    - ej: *plug-gw* (simplemente redirecciona todos los datos)
    - :-) sirve para una amplia gama de aplicaciones
    - :-(proporciona poco control sobre lo que ocurre (similar al filtrado de paquetes)
- **Proxies genéricos frente a específicos**
  - específico = nivel de aplicación, genérico = nivel de circuito
- **Usando proxies sin un servidor proxy**
  - los servidores intermedios están actuando como proxies
  - ej: SMTP, NNTP, NTP...

Caso de ejemplo: SOCKS

- **Proxy genérico, http://www.inet.no/dante/**
  - versiones: SOCKS4 y SOCKS5 (RFCs 1928, 1929 y 1961)
  - SOCKS5 (AFT, Authenticated Firewall Traversal) aporta:
    - autentificación de usuarios (RFC1929)
    - UDP e ICMP
    - resolución de nombres en el servidor SOCKS
- **Características adicionales**
  - registro configurable
  - acciones configurables ante la denegación de acceso
  - POPULAR → muchos clientes que lo soportan (también intrusos)
- **Componentes**
  - servidor, librerías, clientes para SOCKS de ftp, telnet..., *wrappers* para *ping* y *traceroute*, *socksify*

¿Cómo desarrollar clientes para SOCKS?

- **Recompilando**
  - identificar el programa ante *syslog*: SOCKSinit(argv[0]);
  - utilizar las funciones que reemplazan a las de sockets BSD: connect, getsockname, getpeername, bind, accept, listen y select
    - V4: -Dconnect=Rconnect ...
    - V5: -DSOCKS
  - enlazar la aplicación con la librería apropiada: -lsocks (v4), -lsocks5 (v5)
- **Utilizando *socksify* (guión del int. de órdenes)**
  - runsocks program-name args
  - inicializa adecuadamente las variables de entorno apropiadas para conseguir que el programa se enlace dinámicamente con la librería de SOCKS
- **Utilizando SocksCap (extensión para WinSock)**

Caso de ejemplo: TIS FWTK

- **Trusted Information Systems FireWall ToolKit**
  http://www.fwtk.org
  - proxies específicos para:
    - telnet (*telnet-gw*)
    - rlogin (*rlogin-gw*)
    - ftp (*ftp-gw*)
    - x (*x-gw*)
    - http (*http-gw*)
    - SMTP (*smap*/*smapd*)
  - un proxy genérico:
    - *plug-gw*
  - se basa en la modificación de los procedimientos del usuario

Ejemplo de telnet

```
$ telnet pasarela.empresa.com
Trying 192.33.112.117 ...
Connected to pasarela.empresa.com.
Escape character is '^]'.
pasarela telnet proxy (Version V1.0) ready:
tn-gw-> connect proyecto.otra-empresa.com

Red Hat Linux release 7.0 (Guinness)
Kernel 2.4.3 on an i686
login: yo
Password: ##################
Last login: Mon Apr 2 08:54:27 on :0
proyecto_$
```

Guión

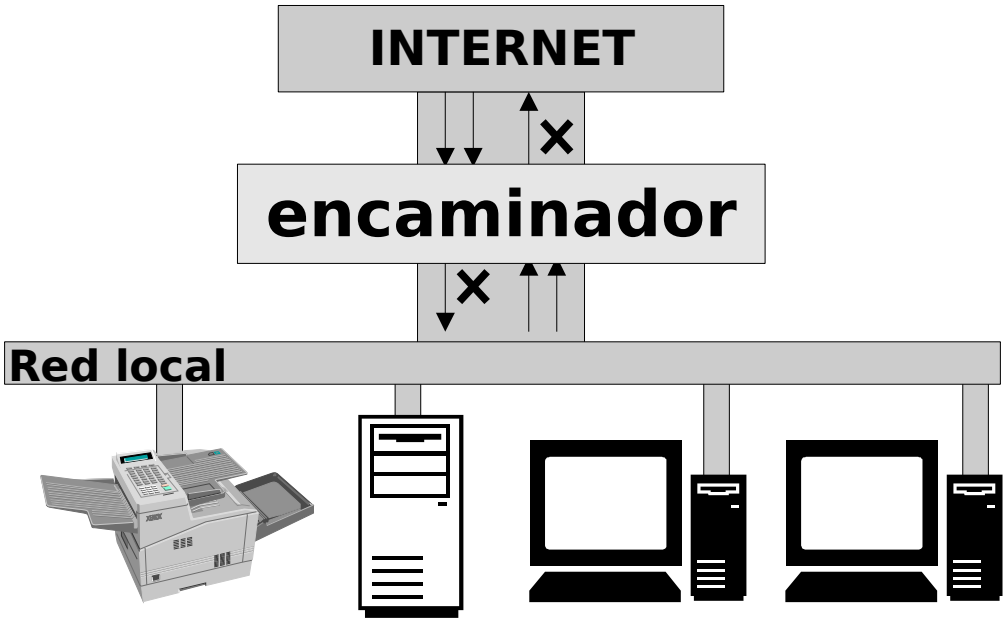
- Concepto de cortafuegos
- Filtrado de paquetes
- Proxies
- **Diseño de cortafuegos**
- Integración de VPNs
- Detección de intrusos

Arquitectura: una sola caja

- **Un solo objeto (encaminador, nodo) actúa como cortafuegos**
- **Ventajas:**
  - menor coste que otras arquitecturas
  - más fáciles de comprender y explicar
  - más fáciles de obtener de proveedores externos
- **No obstante:**
  - la seguridad depende de un único objeto
  - su configuración y su mantenimiento requieren el mismo cuidado que el de cualquier otro cortafuegos
- **Resulta adecuada para sitios pequeños**

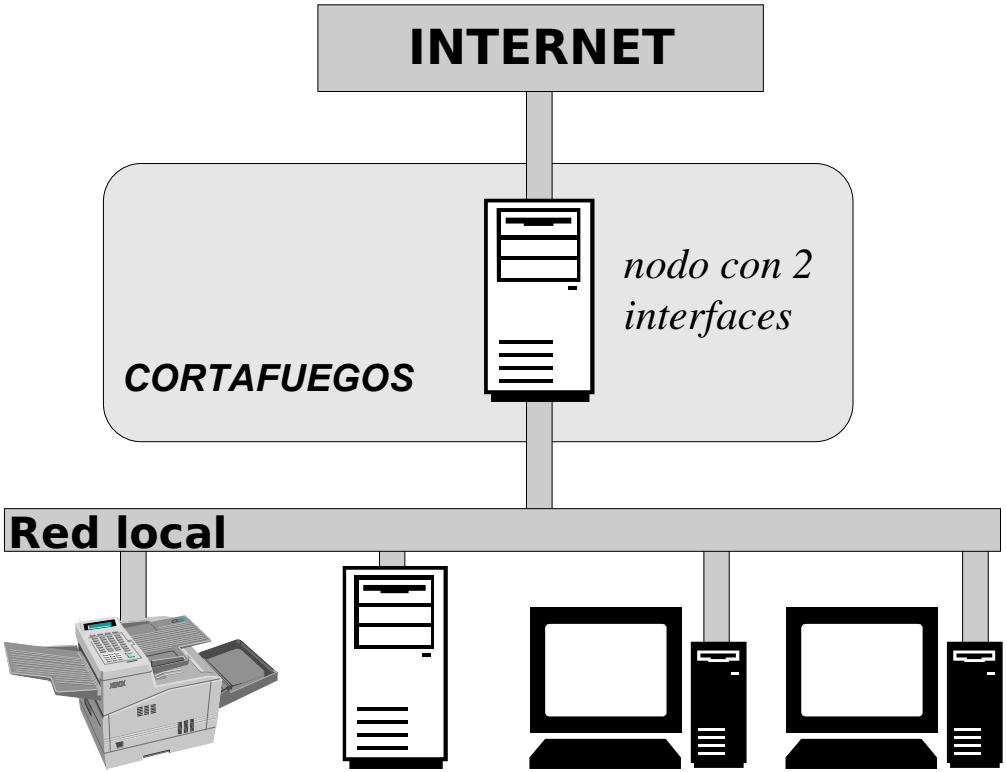
Arquitectura: una sola caja
Encaminador que filtra

- **Ventajas**
  - bajo coste
- **Limitaciones**
  - poca flexibilidad
  - punto único de fallo
- **Apropiado para redes**
  - con nodos con elevado nivel de seguridad
  - en las que se usan pocos protocolos y éstos son simples
  - se requieren buenas prestaciones
- **Usos habituales**
  - cortafuegos internos
  - redes dedicadas a proporcionar servicios de Internet (ej: ISP)



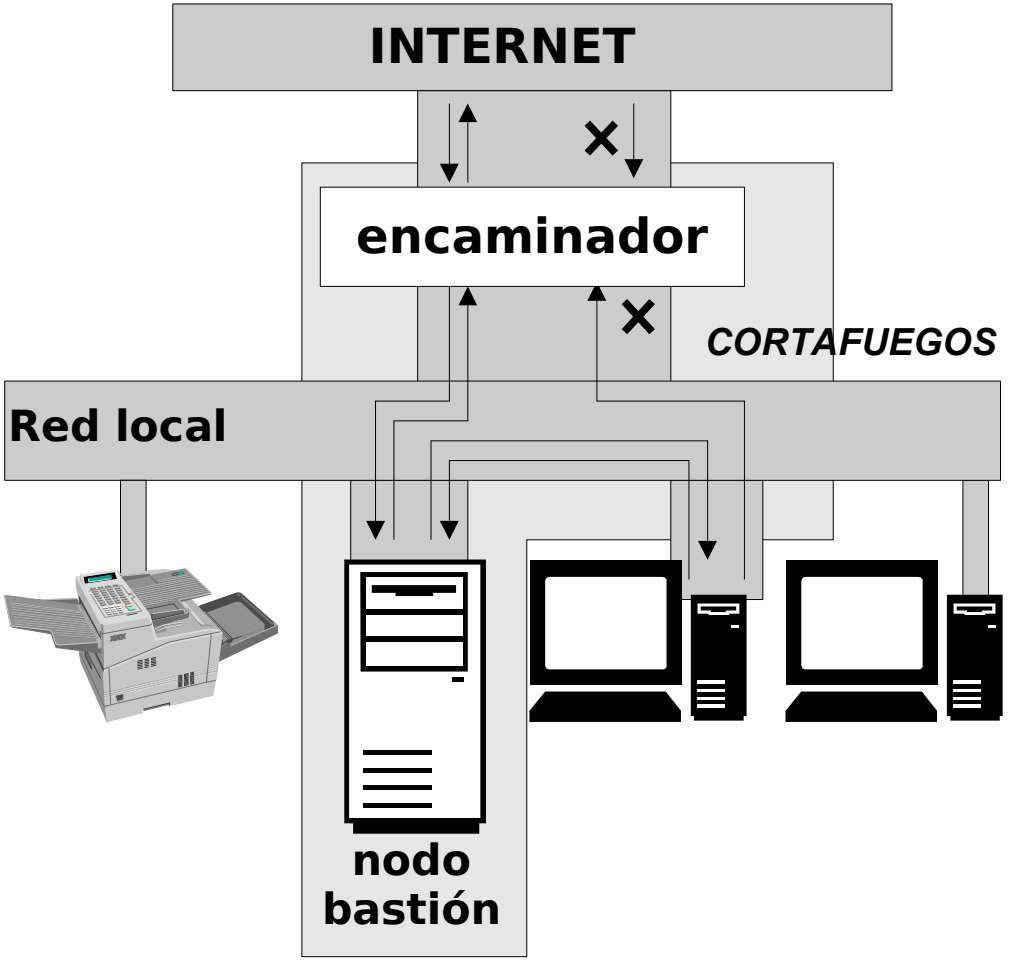
Arquitectura: una sola caja
Computador con 2 interfaces de red

- **Requiere que el nodo NO ENCAMINE paquetes**
- **Ventajas**
  - alto nivel de control
- **Limitaciones**
  - pocas prestaciones
  - punto único de fallo
  - requiere proxies
- **Apropiado para redes**
  - con acceso a servicios externos, no se ofrecen servicios a Internet
  - con poco tráfico hacia/de Internet, no crítico para el negocio
  - los datos almacenados en la red no son especialmente valiosos



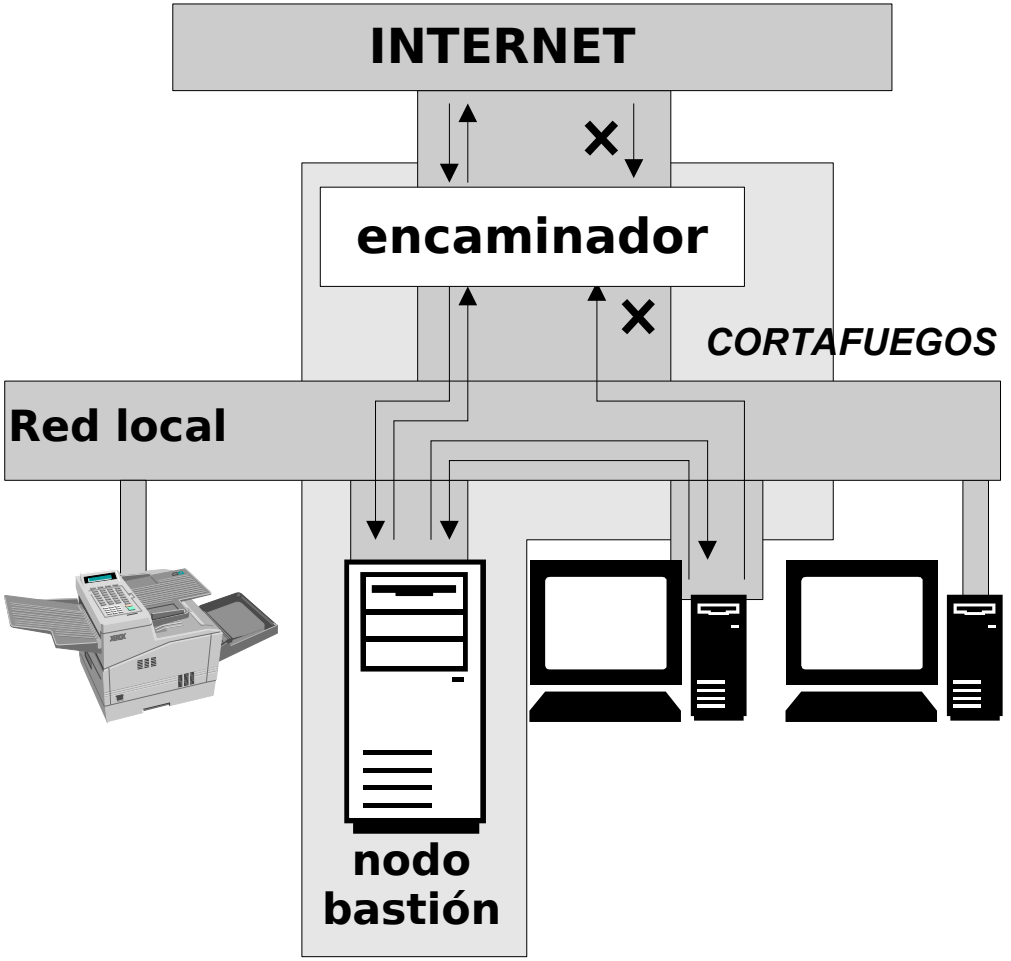
Arquitectura: nodo filtrado

- **Seguridad por filtrado de paquetes**
  - acceso exterior sólo al nodo bastión
  - acceso interior a Internet
    - para ciertos servicios
    - sólo a través de proxies
    - solución intermedia

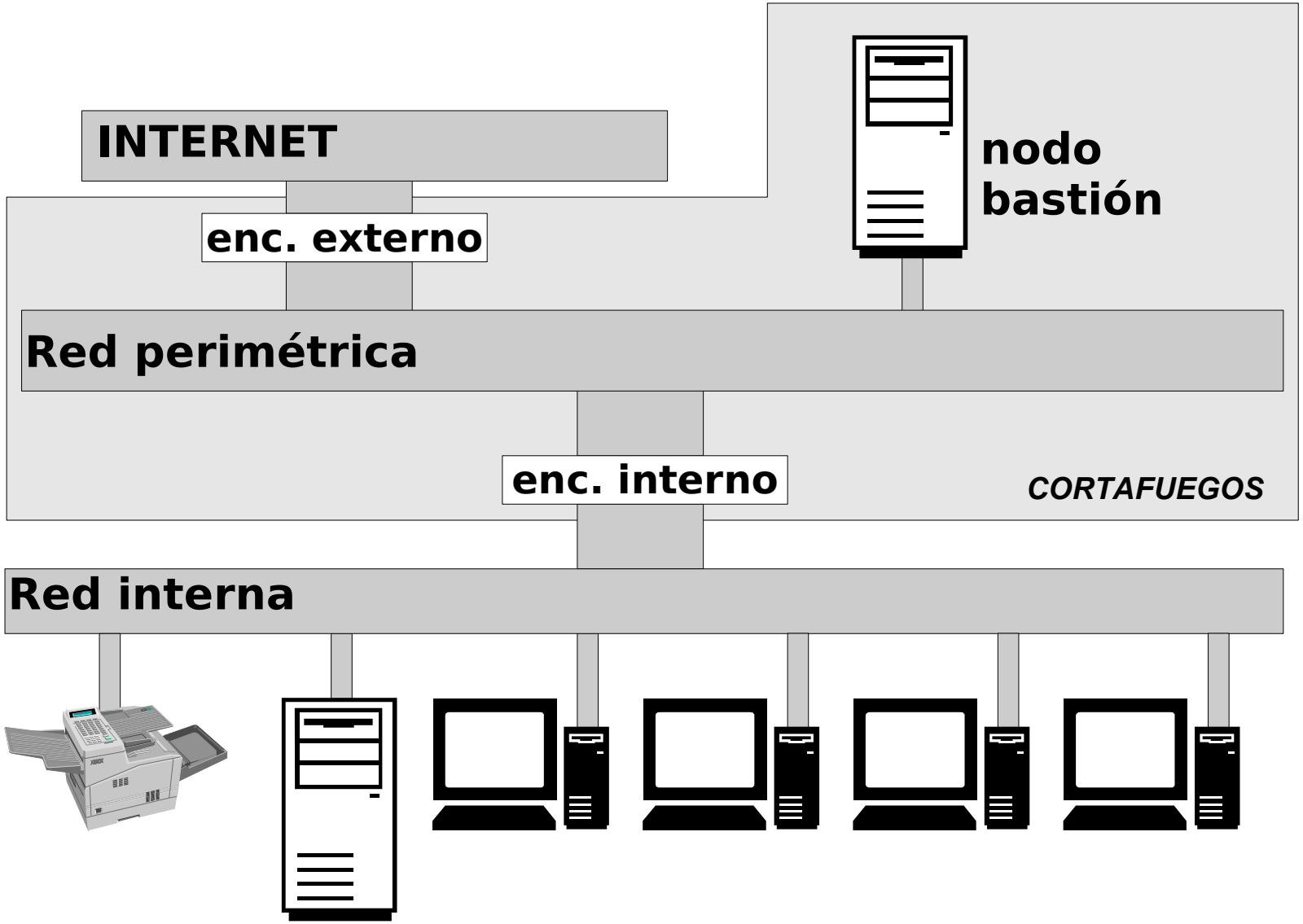


Arquitectura: nodo filtrado

- **Ventajas sobre nodos con 2 interfaces de red**
  - mayor flexibilidad
  - mayor seguridad
- **Peligros**
  - que el nodo bastión se vea comprometido
  - que se vea comprometido el encaminador
- **Usos apropiados**
  - pocas conexiones externas (no sirve para servidores Web)
  - nodos de la red interna con nivel elevado de seguridad



Arquitectura: subred filtrada (I)



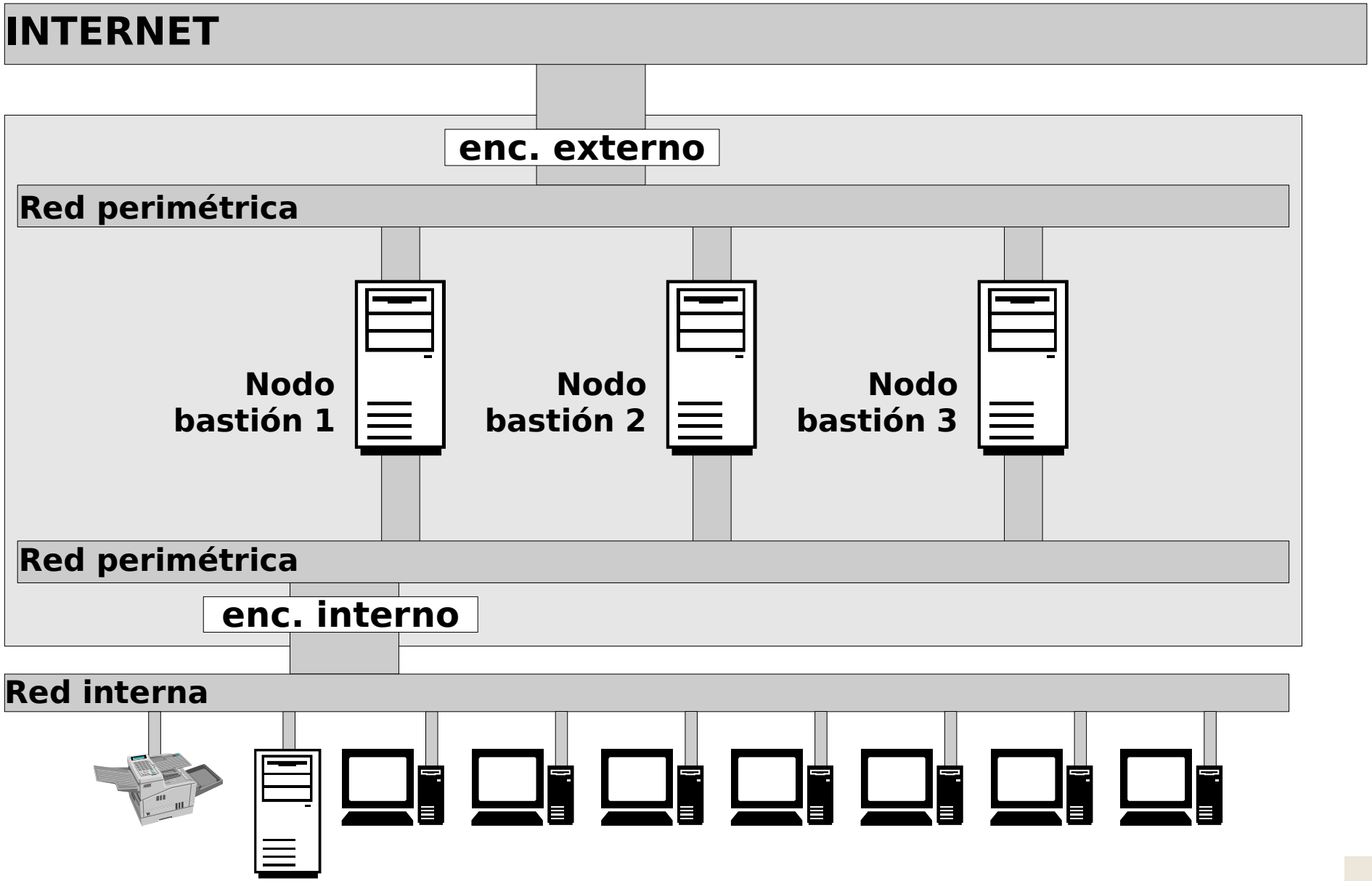
Arquitectura: subred filtrada (II)

- **Motivo fundamental:**
  - aislar la red interna en la que está el nodo bastión
- **En caso de ataque con éxito al nodo bastión:**
  - el encaminador interno restringe el acceso a la red interna
  - sólo el tráfico presente en la red perimétrica se ve comprometido

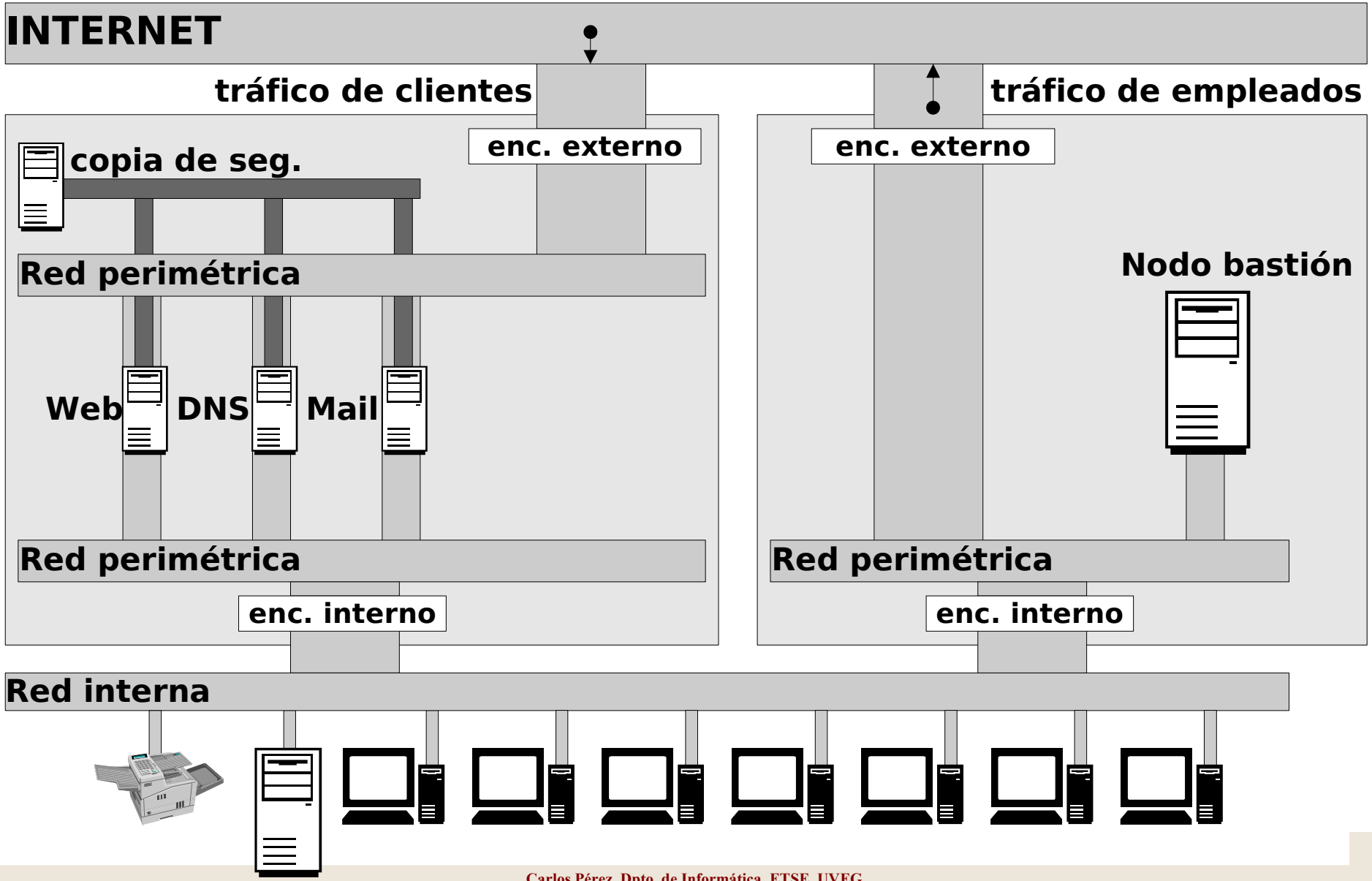
Arquitectura: subred filtrada (III)

- **Ofreciendo servicios:**
  - los proporciona el nodo bastión (o los nodos)
    - encaminador externo: permitir conexiones desde fuera al nodo bastión
- **Servicios de servidores externos:**
  - filtrado de paquetes para permitir acceso directo a/desde nodos concretos
  - proxies en el nodo bastión
    - encaminador interno: permitir conexiones al nodo bastión
    - encaminado externo: permitir conexiones desde el nodo bastión
- **Usos apropiados: la mayoría**

Arquitectura: subred filtrada dividida



Arquitectura: múltiples subredes filtradas independ.



Variaciones

- **Es correcto:**
  - utilizar múltiples nodos bastión: separación de datos y/o servicios
  - unir los encaminadores interno y externo (si es capaz de gestionar independientemente cada interfaz de red)
  - unir el encaminador externo y el nodo bastión (ej: conexión PPP)
  - utilizar más de un encaminador externo (ej: varias conexiones a Internet, conexión a Internet y a otras redes...)
- **En cambio, es peligroso:**
  - unir el nodo bastión y el encaminador interno
  - utilizar más de un encaminador interno
  - utilizar al mismo tiempo una subred filtrada y nodos filtrados

Variaciones

- **Cortafuegos internos:**
  - redes de investigación o prueba
  - redes poco seguras (ej: para demostraciones o enseñanza)
  - redes especialmente seguras (ej: proyectos de desarrollo, con información privilegiada...)

Recomendaciones finales

- **Definir las necesidades**
  - ¿Qué tendrá que hacer el cortafuegos? Servicios a ofrecer, seguridad, uso, fiabilidad.
  - ¿Qué limitaciones existen? Presupuesto, personal, entorno actual.
- **Evaluar los productos disponibles**
  - Escalabilidad
  - Fiabilidad y redundancia
  - Auditabilidad
  - Precio (hw, sw, soporte y actualizaciones, administración e instalación)
  - Gestión y configuración
  - Adaptabilidad
  - Adecuación

Recomendaciones finales (II)

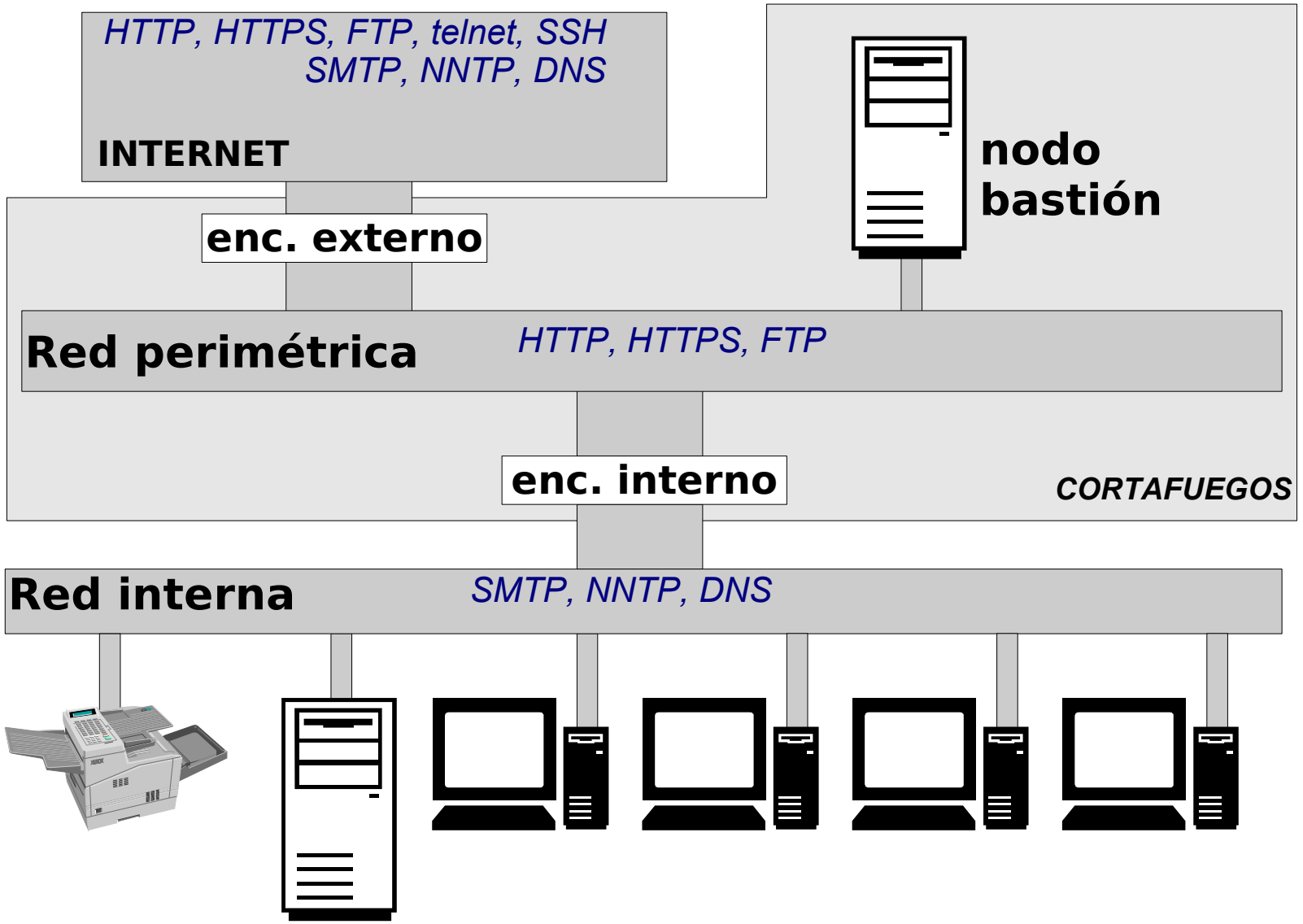
- **Integrando los componentes**
  - Registros, informes rutinarios y alarmas
  - Copias de seguridad
  - Servicios de los que depende el cortafuegos
  - Acceso al cortafuegos para administración y mantenimiento
- **Seguir el principio del mínimo privilegio (*least privilege*)**
- **Utilizar varias capas de protección (*defense in depth*)**
- **Utilizar puntos de estrangulamiento (*choke points*)**
- **Cuidado con el eslabón más débil (*weakest link*)**

Recomendaciones finales (III)

- **Involucración universal**
- **Inocuidad ante averías (*fail-safe stance*)**
- **Diversificar las defensas. Cuidado con:**
  - protocolos comunes
  - configuraciones comunes
  - herencia común
- **Simplicidad (*KISS*)**

Caso de ejemplo: Subred Filtrada

Building Internet Firewalls, pp. 681-704



Configuración de servicios HTTP y HTTPS

- **Consideraciones:**
  - El filtrado de paquetes sólo permite el acceso a servidores en puertos estándar
    - se podría permitir el acceso a los puertos 80 (http), 443 (https) y >1023 (útil también para ftp)
    - >1023 es arriesgado (especialmente con usuarios poco expertos o mal intencionados)
  - Los clientes habituales soportan proxies
  - Si el proxy tiene cache (habitual) se benefician
    - los clientes HTTP (mayor rapidez con éxito de cache)
    - los clientes de otros servicios (HTTP no consume tanto ancho de banda)
    - los servidores HTTP de otros sitios (menos visitas de nuestros usuarios)

Configuración de servicios HTTP y HTTPS

- **Consideraciones (continuación):**
  - HTTPS no admite el uso de cache
  - Muchos proxies pueden actuar al mismo tiempo como servidores HTTP
    - combinar ambos es poco seguro
    - inaceptable para páginas con muchos accesos
  - Coste de instalación, configuración y mantenimiento elevado (amortizable si se va a usar proxies con otros servicios)
- **DECISIÓN: empleo de proxy y servidor combinado**

Configuración de servicios SMTP entrante

- **a una máquina interna con serv. SMTP seguro**
  - poner un servidor SMTP seguro en el nodo bastión
  - redirigir el correo entrante al nodo mediante registros MX en DNS
  - el nodo bastión lo reenvía a un nodo interno bien configurado
  - el nodo interno lo distribuye
- **alternativas**
  - redirigir directamente el correo a la máquina interna
    - si la máquina es comprometida, toda la red interna puede ser comprometida
  - distribuir el correo desde el nodo bastión
    - si el nodo bastión es comprometido, todos los nodos que puedan recibir correo peligran y, por lo tanto, también la red interna
    - sería necesario realizar tareas de mantenimiento frecuentes en el nodo bastión

Configuración de servicios SMTP saliente

- **enviarlo al nodo interno**
  - permite ocultar información sobre las máquinas internas
    - a Internet
    - al nodo bastión
  - el correo interno no pasa por el nodo bastión

Configuración de servicios
Acceso remoto: telnet

- **Saliente: filtrado de paquetes**
  - un proxie permite ejercer mayor control (autentificación, registro...)
  - pero en este caso se confía en los usurios
  - además tenemos direcciones IP válidas
  - el filtrado permite ofrecerlo de forma segura
- **Entrante**
  - no es posible ofrecerlo de forma segura → usar otra alternativa (ej: SSH)

Configuración de servicios
Acceso remoto: SSH

- **entrante: conexión directa**
  - permitirlo sólo al nodo bastión
    (y de allí a los nodos de la red interna)
    - mayor control (seguridad de que el servidor está bien configurado)
    - necesidad de mantener cuentas de usuario en el nodo bastión
  - permitir la conexión directa a las máquinas internas
    - posibilidad de que se ejecute otro servidor
    - o un servidor SSH mal configurado (ej: que permita la redirección de puertos)
- **saliente: conexión directa**
  - la redirección de puertos abre un agujero de seguridad
    (que no abre el telnet saliente)

Configuración de servicios FTP saliente

- **Opciones:**
  - permitir FTP pasivo mediante filtrado de paquetes
    - exige que los clientes sean capaces de realizarlo (ej: los de los navegadores)
    - permite conexiones a cualquier puerto > 1023
  - permitir FTP normal mediante proxies: ftp-gw de TIS FWTK
    - permite supervisar las conexiones
    - exige modificaciones en los procedimientos de usuario (ej: ftp-gw de TIS FWTK)
- **Decisión: permitir ambos**
  - permite usar directamente clientes que soporten ftp pasivo
  - permite usar otros clientes modificando los procedimientos de los usuarios
  - supone:
    - que los usuarios no son malintencionados
    - que saben distinguir entre servicios seguros e inseguros o
    - que no prueban servicios desconocidos

Configuración de servicios FTP entrante

- **Decisión: sólo FTP anónimo usando TIS FWTK**
  - el ftp entrante de usuarios es tan inseguro como telnet → desactivado
- **Para una gran carga:**
  - ftp anónimo usando un servidor especializado (ej: wu-ftpd)
  - ubicarlo en un nodo bastión dedicado

Configuración de servicios NNTP y DNS

- **NNTP**
  - permitir la comunicación entre el servidor NNTP (ej: el del ISP) externo y el interno
  - no es aconsejable poner el servidor de NNTP en el nodo bastión
    - lo sobrecargaría
    - suelen fallar habitualmente (normalmente no tiene que ver con la seguridad)
- **DNS**
  - utilizar un par de servidores
    - uno en el nodo bastión (secundario para nuestro dominio)
    - otro en un nodo interno (el primario de nuestro dominio)
  - no ocultar información (es complicado con ftp pasivo directo)

Configuración del cortafuegos (I)

- **Suponiendo que el filtro permite:**
  - distinguir entre paquetes entrantes y salientes
  - filtrado basado en: dirección fuente y destino, tipo de paquete, puerto origen y destino
  - filtrado basado en el bit ACK (paquetes TCP)
- **Encaminador interno**
  - Protege la red interna de Internet y del nodo bastión
  - reglas:
    - Usurpación de direcciones: rechazar paquetes entrantes con dirección fuente interna y paquetes salientes con dir. fuente externa
    - HTTP: permitir conexiones salientes al puerto 80 y tráfico entre este puerto externo y puertos internos >1023
    - TELNET: permitir conexiones salientes al puerto 23 y tráfico entre este puerto externo y puertos internos >1023
    - ....
    - Denegar todos los paquetes (en ambos sentidos) no permitidos explícitamente

Configuración del cortafuegos (II)

- **Encaminador externo**
  - conecta la red perimétrica (e, indirectamente, la red interna) a Internet
  - protege las redes perimétrica e interna de Internet
  - sirve como respaldo del encaminador interno
  - reglas:
    - Usurpación de direcciones:
      rechazar los paquetes entrantes con dirección fuente interna o perimétrica y
      los salientes con dirección externa
    - HTTP:
      - permitir conexiones desde el nodo bastión a cualquier puerto externo
        y el tráfico correspondiente
      - permitir conexiones al puerto 80 del nodo bastión y el tráfico correspondiente
    - Telnet: permitir conexiones desde nodos internos (no desde el nodo bastión) al puerto 23 y el tráfico correspondiente
    - ...
    - Denegar todos los paquetes (en ambos sentidos) no permitidos explícitamente

Conf. del cortafuegos (III)

- **Máquinas internas**
  - el correo saliente debe ir al servidor SMTP interno
  - instalar clientes de ftp pasivo
  - instalar clientes y servidores de SSH con la redirección de puertos desactivada
- **Servidor de correo interno**
  - instalar un servidor de correo de confianza
- **Servidor de nombres interno (primario)**
  - poner un registro MX por cada reg. A apuntando al nodo bastión
  - poner registros MX adicionales para la gestión del tráfico interno
  - configurar el nodo bastión como servidor de nombres secundario
  - eliminar todos los registros adicionales TXT y HINFO (para evitar que sean públicos)

Conf. del cortafuegos (III)

- **Nodo bastión**
  - configurar el nodo como nodo bastión (ver apartado correspondiente)
  - instalar TIS FWTK (gw-ftp, servidor SMTP, ftp anónimo)
  - instalar el servidor HTTP y configurarlo como proxy
  - preparar las páginas que deben ser mostradas al exterior

Análisis de la propuesta (I)

- **Se aplica el principio del mínimo privilegio**
  - ej: sólo el nodo bastión se conecta al exterior para enviar correo
- **Defensa en profundidad**
  - la red interna protegida por los dos encaminadores
  - el nodo bastión cuidadosamente configurado y protegido por el encaminador externo
  - el empleo de un servidor SMTP interno (entre el nodo bastión y los nodos internos)
- **Punto de estrangulamiento**
  - ej: la red perimétrica, la utilización de proxies...

Análisis de la propuesta (II)

- **El eslabón más débil**
  - proxy de ftp: permite atacar los puertos >1023 internos tras comprometer el bastión
  - SSH: contraseñas débiles, redirección de puertos
  - Servidor Web en el nodo bastión: comprometer nodos internos, ataques DOS
- **Configuración inocua ante averías**
  - el filtrado de paquetes deniega todo lo que no se permite explícitamente
  - redundancia en el filtrado

Análisis de la propuesta (III)

- **Involucración universal**
  - obligatoriamente: acceso a Internet a través del cortafuegos
  - voluntariamente: telnet, SSH, ftp...
  - se ha supuesto que no existen conexiones adicionales a Internet (ej: módems)
- **Diversidad de mecanismos de defensa**
  - utilizar encaminadores de fabricantes distintos
  - configuración por equipos independientes y comprobación posterior
  - utilizar diferentes servidores SMTP en el nodo bastión y en el nodo interno
- **Simplicidad**
  - separación en componentes (ej: 2 encaminadores independientes)

Guión

- Concepto de cortafuegos
- Filtrado de paquetes
- Proxies
- Diseño de cortafuegos
- **Integración de VPNs**
- Detección de intrusos

Posibles VPNs

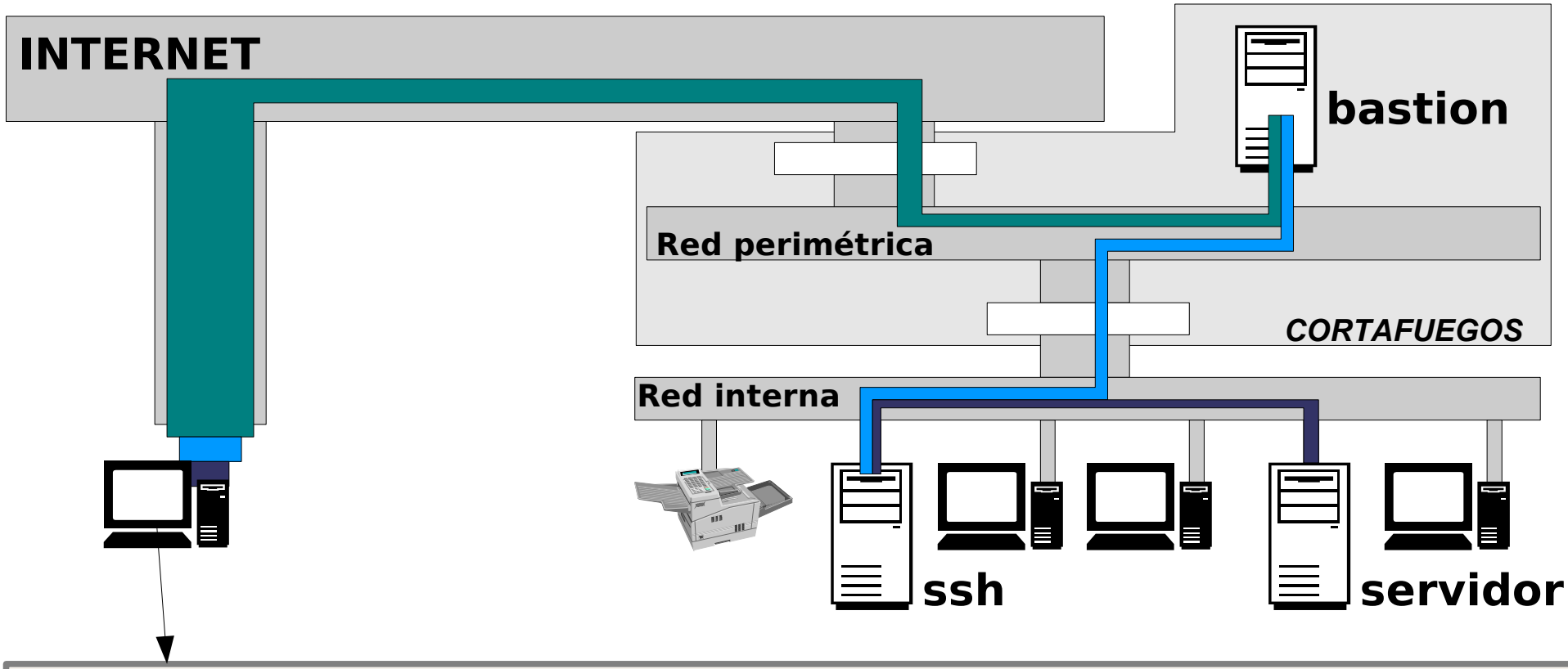
- Conexiones y túneles SSH
- IPSec
- Conexiones, túneles y proxies de SSL
- Software para escritorios remotos

Conexiones y túneles SSH

- **Conexiones : abrir la entrada al puerto 22**
  - a todos los servidores de SSH
  - a un servidor en la red perimétrica (más seguro)
    - desde este servidor, permitir el resto de conexiones
  - ¿sustituir FTP por SSH?
    - se evitan los problemas y vulnerabilidades de FTP
- **Túneles SSH (*port forwarding*)**
  - es necesario abrir demasiadas máquinas y puertos
  - a no ser que... hagamos túneles dentro de túneles

```
$ ssh -fN -L 7022:ssh.empresa.com:22 bastion.empresa.com
$ ssh -fN -o "HostKeyAlias ssh.empresa.com" localhost -p 7022 -L
7023:servidor.empresa.com:22
$ ssh -o "HostKeyAlias servidor.empresa.com" localhost -p 7023
Last login: Fri Jan 12 11:52:04 2007 from cperez.empresa.com
Have a lot of fun...
cperez@servidor:~>
```

Ejemplo de túneles SSH




```
$ ssh -fN -L 7022:ssh.empresa.com:22 bastion.empresa.com
$ ssh -fN -o "HostKeyAlias ssh.empresa.com" localhost -p 7022 -L
7023:servidor.empresa.com:22
$ ssh -o "HostKeyAlias servidor.empresa.com" localhost -p 7023
Last login: Fri Jan 12 11:52:04 2007 from cperez.empresa.com
Have a lot of fun...
cperez@servidor:~>
```

IPSec

- **Opciones**
  - de nodo a nodo
  - de nodo a pasarela
  - de pasarela a pasarela
- **Recomendaciones**
  - asumir que los nodos externos pueden ser maliciosos
  - no utilizar implementaciones con algoritmos propietarios
  - permitir que el tráfico resultante sea inspeccionado
  - pasarela de VPN en la DMZ ó red perimétrica
  - sensores específicos para la detección de intrusos

Guión

- Concepto de cortafuegos
- Filtrado de paquetes
- Proxies
- Diseño de cortafuegos
- Integración de VPNs
- **Detección de intrusos**

Detección de intrusos en seguridad perimétrica

- **NIDS (*Network-based Intrusion Detection System*)**
  - analiza el tráfico de red
  - detecta intentos de intrusión
- **Detección de anomalías**
  - análisis estadístico → tráfico anormal
  - análisis de protocolos → violaciones del protocolo, tráfico inusual
- **Detección de firmas**
  - firma → patrón que identifica un intento de intrusión
- **Falsos positivos y falsos negativos**

El papel del NIDS en la seguridad perimétrica

- Identificar debilidades
- Auditoría de seguridad
- Violaciones de la política de seguridad
- Detección de ataques internos
- Gestión de incidentes y análisis forense
- Complemento de otros componentes

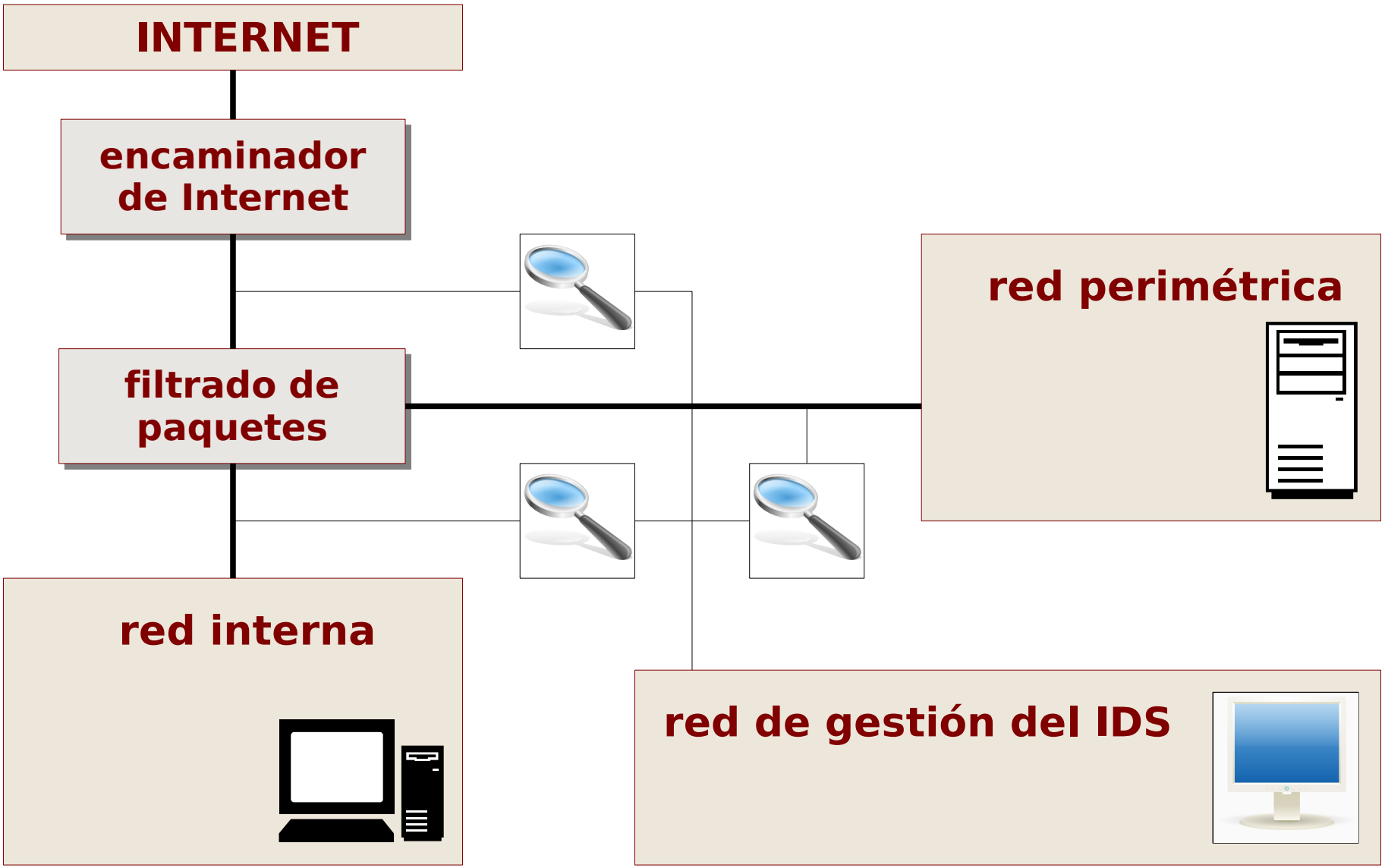
Ubicación de sensores

- **Múltiples sensores**
  - permite ajuste fino a las características del segmento de red
  - tolerancia a fallos
- **Situados cerca los dispositivos de filtrado**
  - externos: intentos externos, ataques internos
  - internos: errores de configuración, intentos internos, ataques externos
- **Otras consideraciones**
  - canales encriptados (ej: VPNs)
  - tráfico muy intenso
  - red de gestión de IDSs
  - seguridad de los sensores

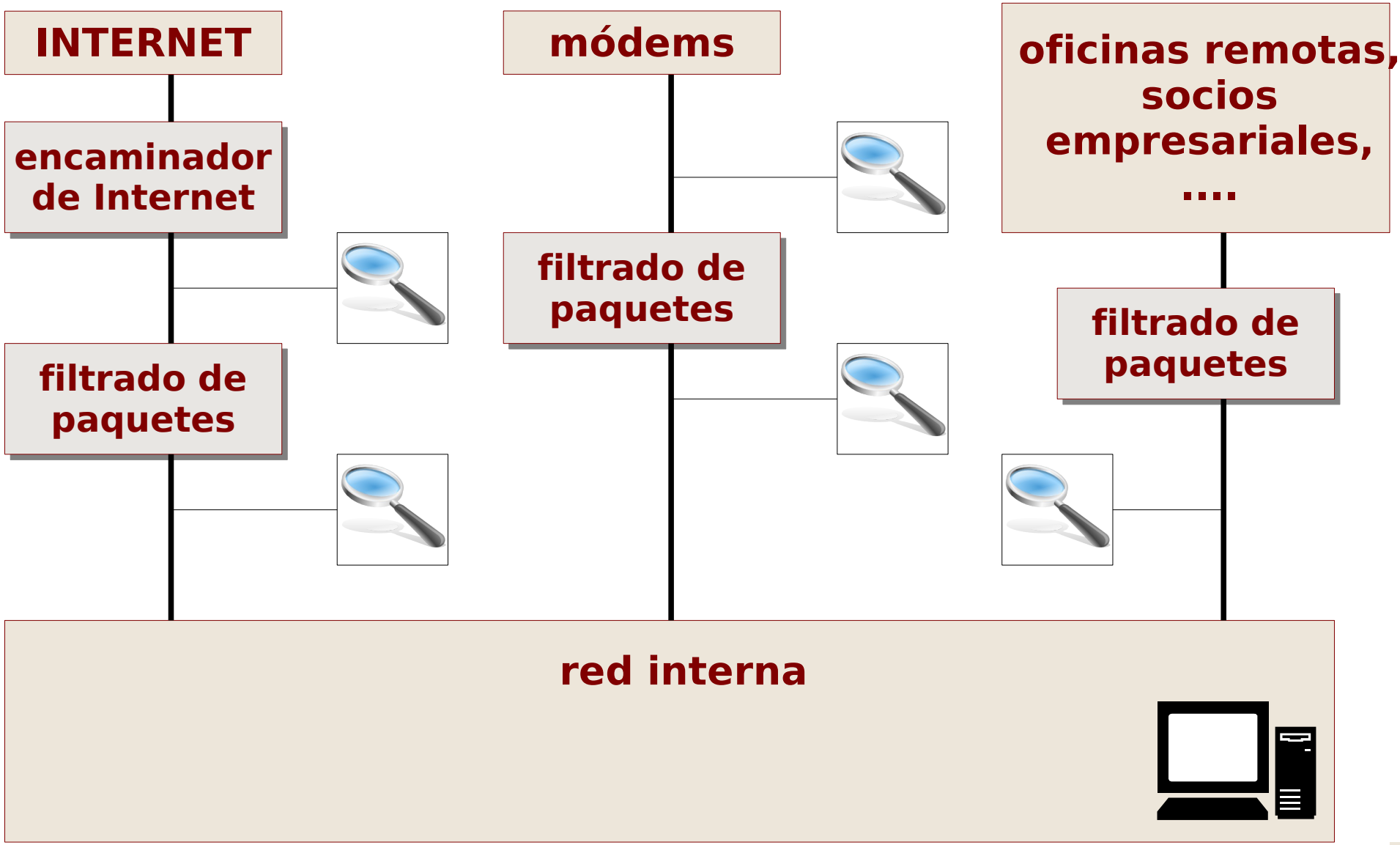
Limitaciones de los NIDSs

- **No todo es observable**
  - eventos en otras redes
  - NIDS no operativo
  - protocolo desconocido para el NIDS
  - paquetes descartados por el NIDS
- **El peligro de las reglas estándar**
- **Factores humanos que limitan la detección**
  - limitaciones del analista
  - limitaciones de los CIRTs

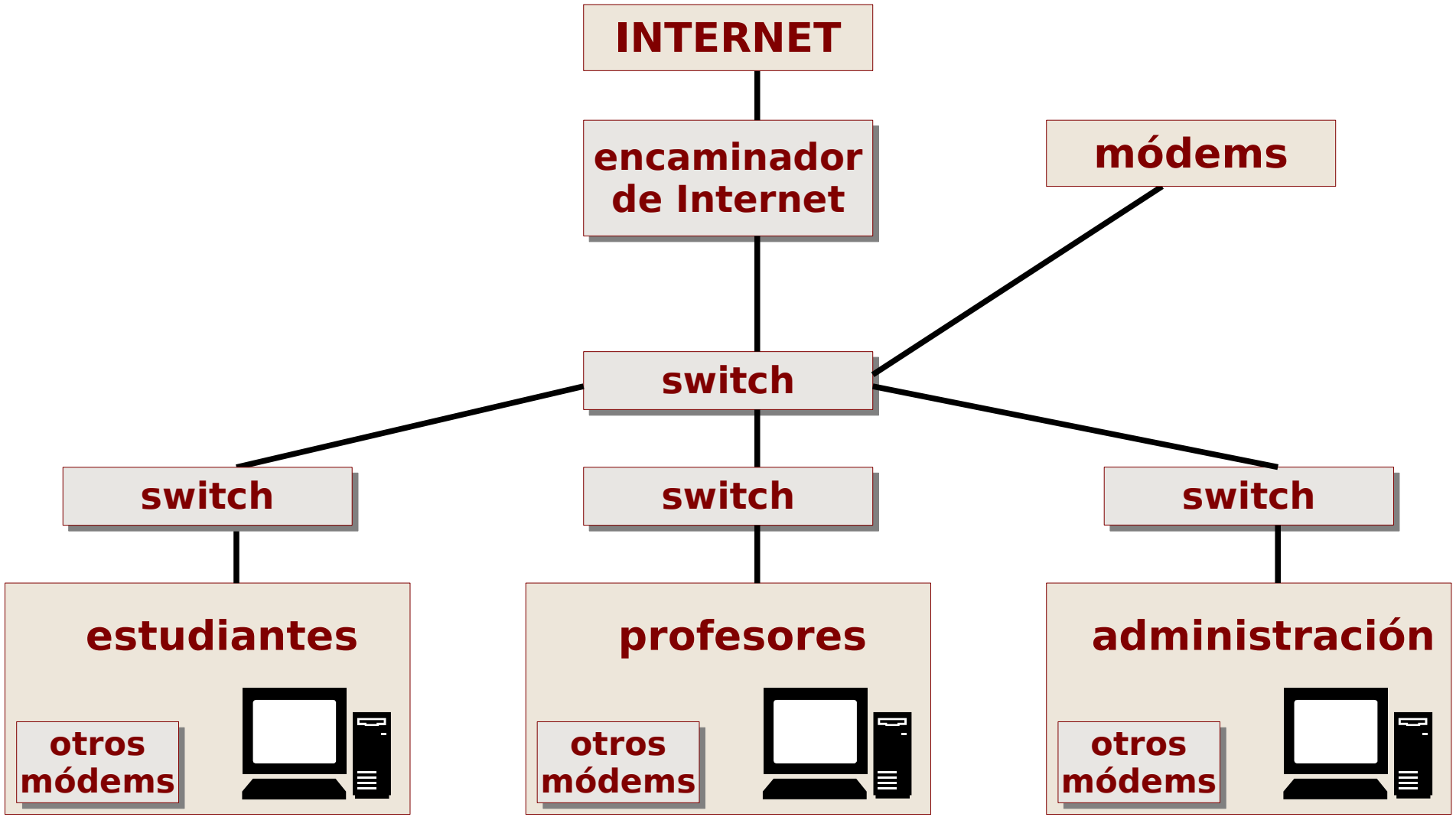
Caso 1: Red sencilla



Caso 2: Red compleja



Caso 3: Red universitaria



Snort, ¿para qué sirve?

- **esnifar (*sniffer mode*)**
  - lee paquetes de la red y los muestra en pantalla
  - snort -v [-d] [-e]
- **guardar paquetes (*packet logger mode*)**
  - lee paquetes de la red y los almacena en disco
  - ASCII:snort -l log-dir
  - binario (compatible con tcpdump): snort -l log-dir -b
- **detección de intrusos (*NIDS mode*)**
  - analiza el tráfico y realiza acciones en base a reglas predefinidas
  - snort -c snort.conf
- **en línea (*inline mode*)**
  - obtiene los paquetes de *iptables* y hace que se acepten/rechacen

Las reglas de Snort

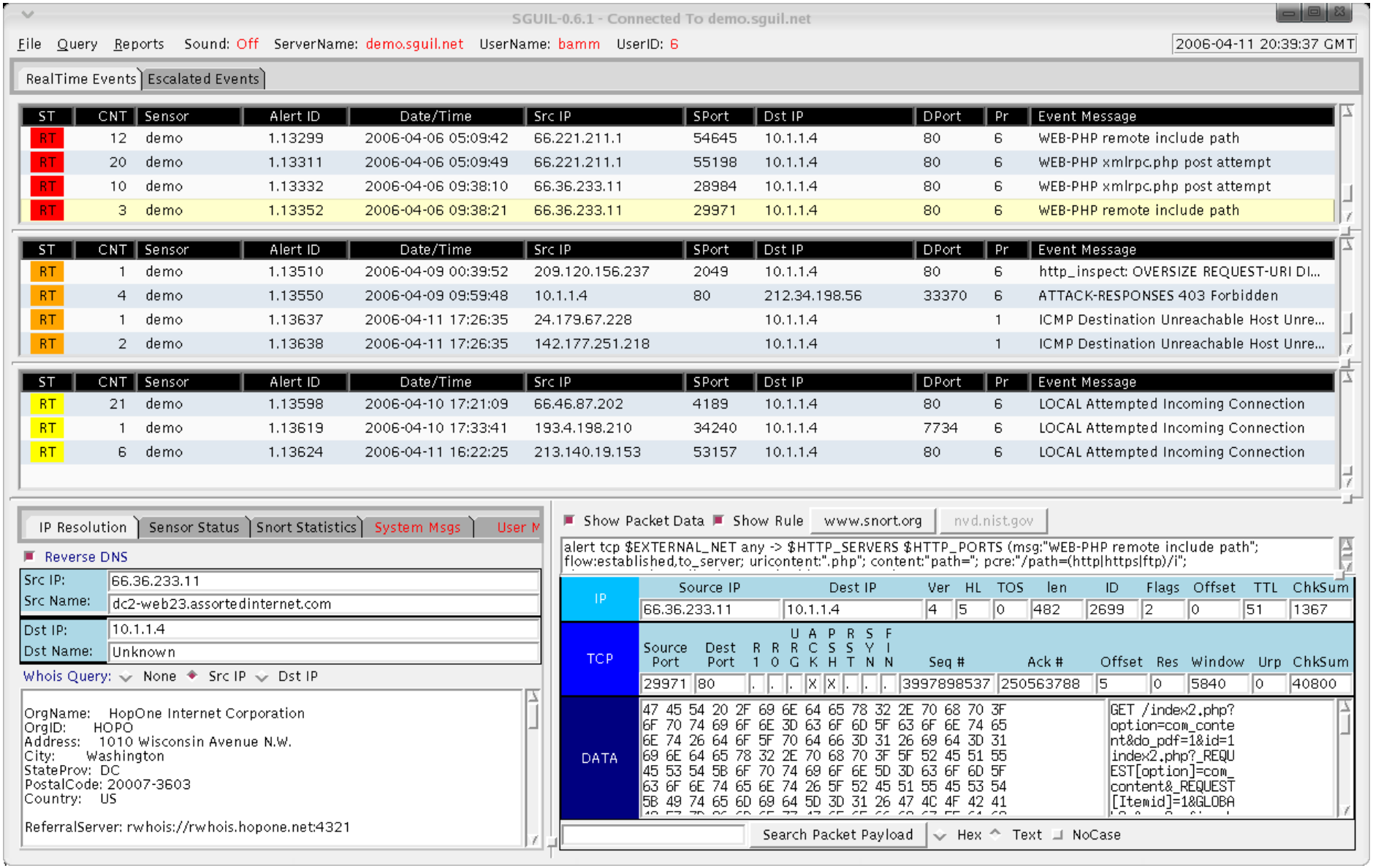
```
# grep 'ICMP PING NMAP' /etc/snort/rules/icmp.rules
alert icmp $EXTERNAL_NET any -> $HOME_NET any \
     (msg:"ICMP PING NMAP"; \
      dsize:0; itype:8; \
      reference:arachnids,162; classtype:attempted-recon; sid:469; rev:3;)
#
```

- **Cabecera**
  - acción (alert, log, pass...)
  - protocolo
  - origen (dirección IP, puerto) y destino
- **Opciones**
  - mensaje
  - atributos del paquete (tamaño, indicadores...)

Consola del analista

- Sguil
  - http://sguil.sourceforge.net/
- OSSIM
  - http://www.ossim.net/
- AirCERT
  - http://aircert.sourceforge.net/
- BASE
  - http://sourceforge.net/projects/secureideas/
- ACID
  - http://acidlab.sourceforge.net/

Consola del analista: SGUIL



Monitorización de la red (NSM)