Cortafuegos con GNU/Linux

Dr. Luis Gerardo de la Fraga

Departamento de Computación
Cinvestav
Correo-e: fraga@cs.cinvestav.mx

8 de mayo, 2014

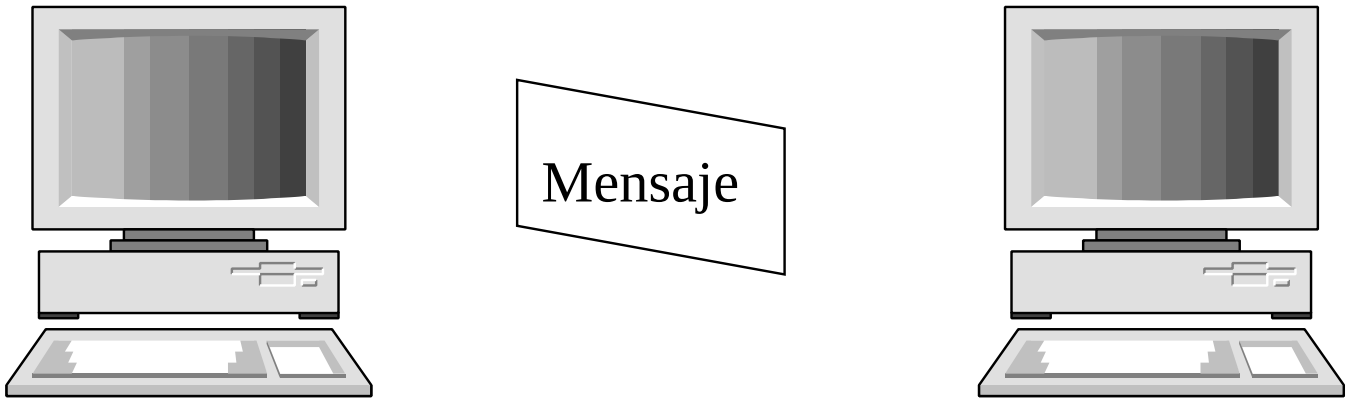
Contenido

1. ¿Por qué es inseguro Internet?
2. ¿Para qué se usa un cortafuegos?
3. Cómo hacer un cortafuegos con GNU/Linux, una receta.

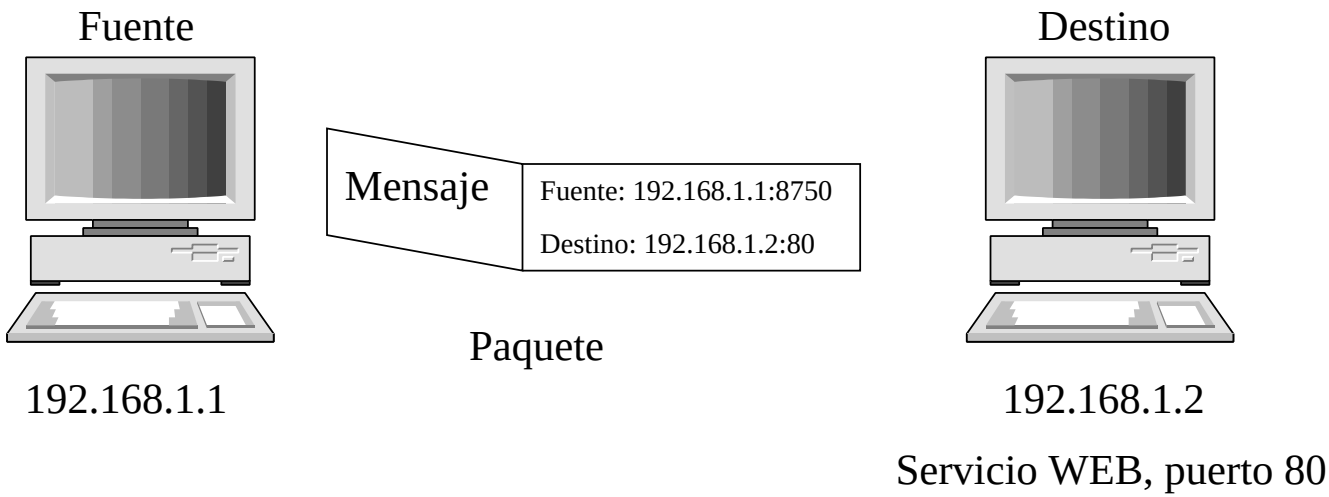
¿Por qué es inseguro Internet?

1. Porque fue diseñado así
2. Porque no están bien especificados los encabezados de las tramas
3. Porque la seguridad se incrusta en última capa de aplicación

Comunicación entre dos computadoras



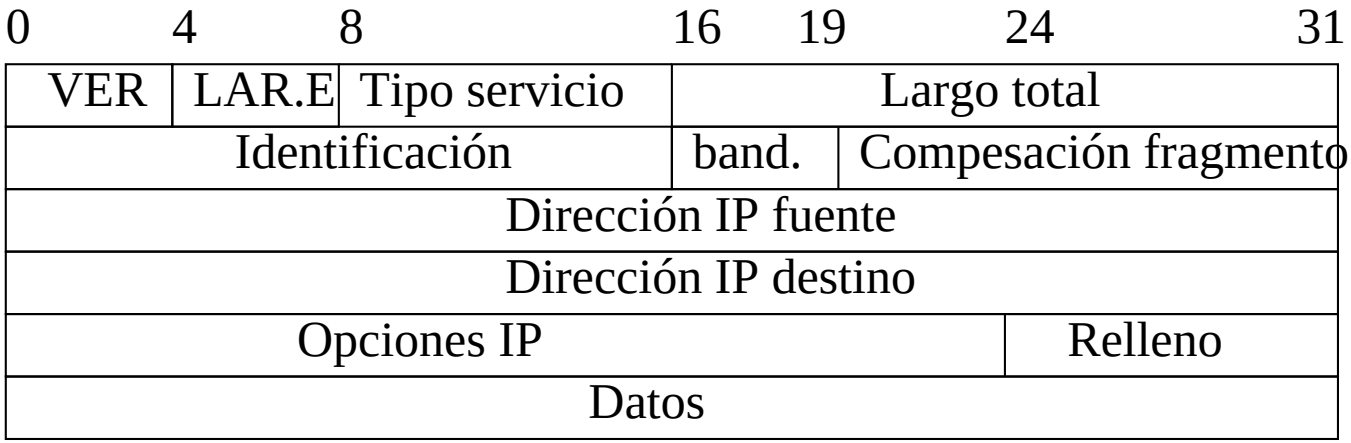
Formación de paquetes



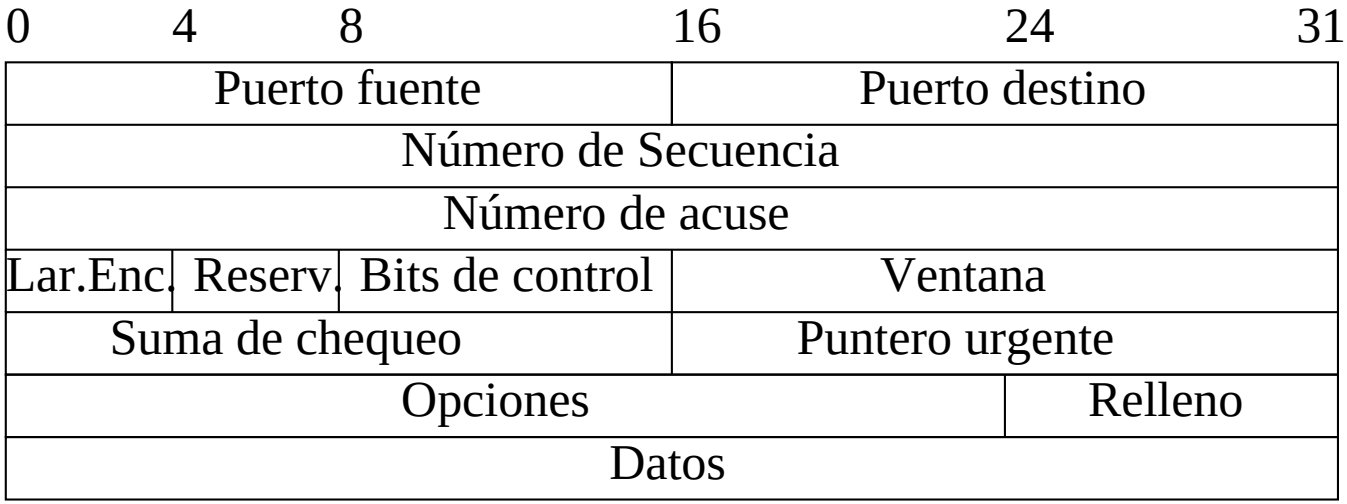
Internet

- Intenet nació en 1969
- Se definió el uso del protocolo TCP/IP para el intercambio de mensajes
- Se usó el concepto es switcheo de paquetes, inventado por Paul Baran (1926-2011).

TCP/IP. El encabezado de IP



TCP/IP. El encabezado de TCP

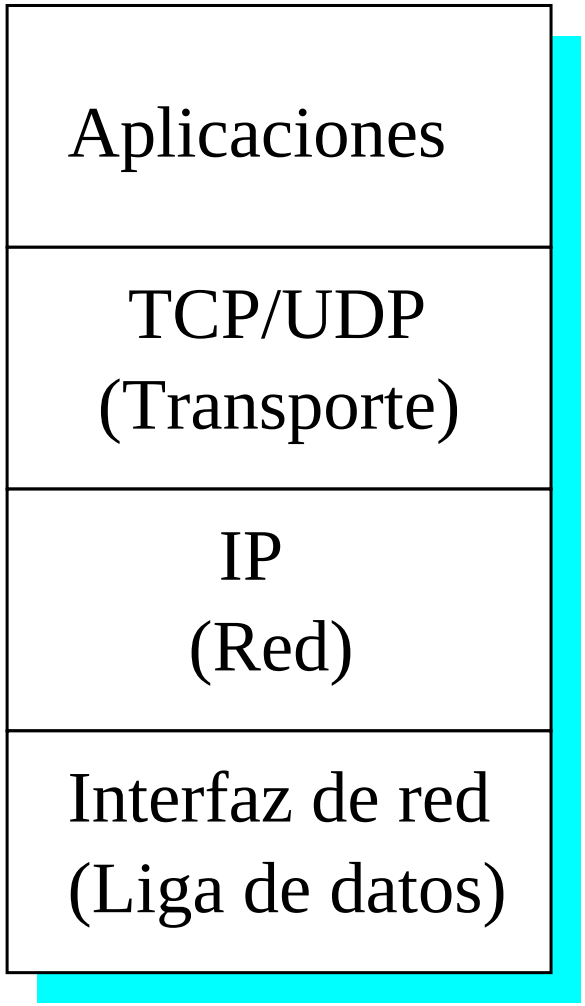


Ventajas de usar TCP/IP

TCP/IP permite plataformas-entrelazadas o administración de redes. TCP/IP también tiene las siguientes características:

- Buena recuperación de las fallas
- Habilidad de añadir redes sin interrumpir los servicios ya existentes.
- Manejo de alto porcentaje de errores
- Independencia de la plataforma
- Bajos gastos indirectos de información.

Capas TCP/IP


Aplicaciones
TCP/UDP
(Transporte)
IP
(Red)
Interfaz de red
(Liga de datos)

Inseguridades

- En IPv4 no está bien especificado el uso de todos los campos de los encabezados. Se pueden usar esos campos para transmitir información.
- En IPv6 si se tienen definidos todos los campos, si se necesitán más se definen nuevos encabezados.
- En IPv4 se implementa la seguridad en las aplicaciones.
- En IPv6 se puede usar IPSec en la primera capa de interfaz de red.

Uso de cortafuegos

El CERT/CC publica que un sitio ideal en seguridad debe contar con:

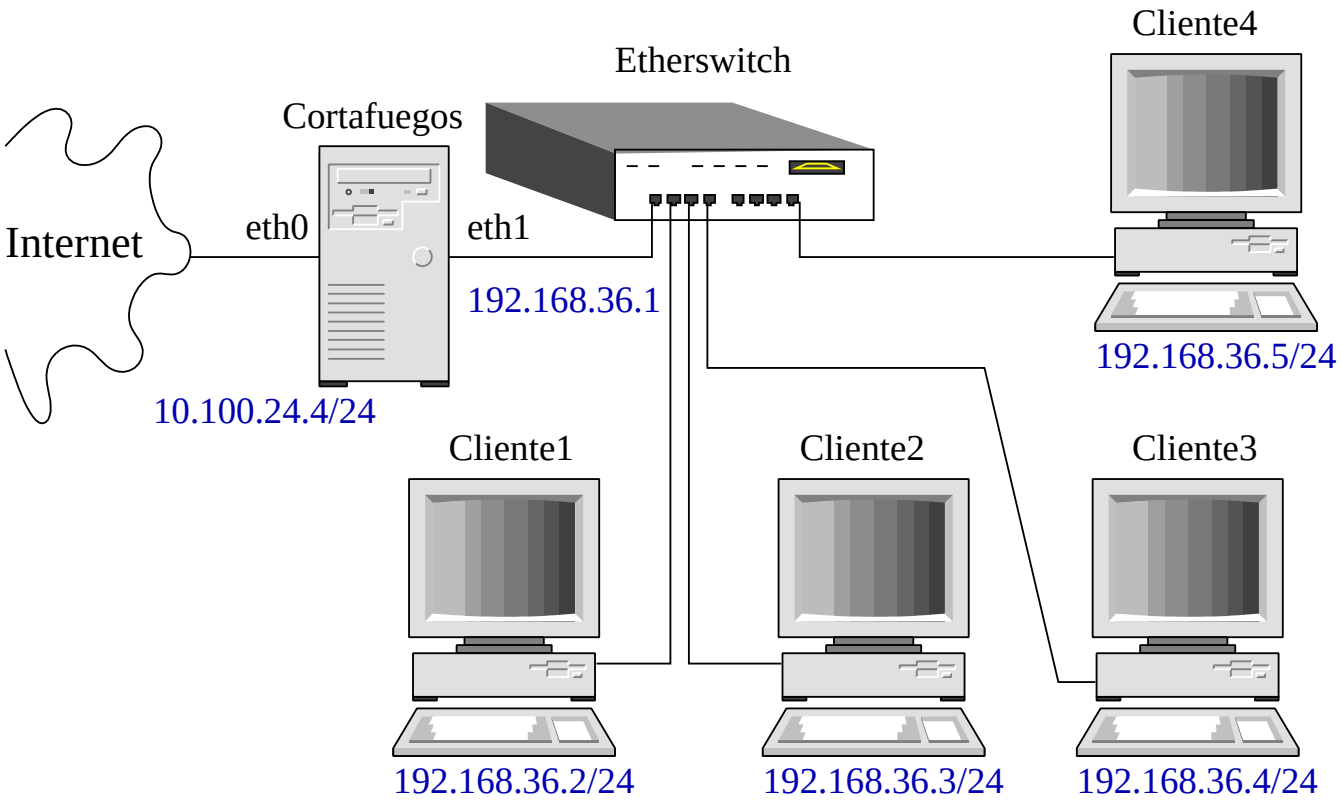
1. Estar al día en parches
2. Usar cortafuegos
3. Debe monitorearse la red
4. Deben deshabilitarse los servicios y características que no son necesarios
5. Tener un software de antivirus instalado, configurado y actualizado
6. Una política para la realización de respaldos
7. Un equipo entrenado y con capacidad de respuesta a incidentes

Problemas de seguridad que pueden resolverse usando cortafuegos

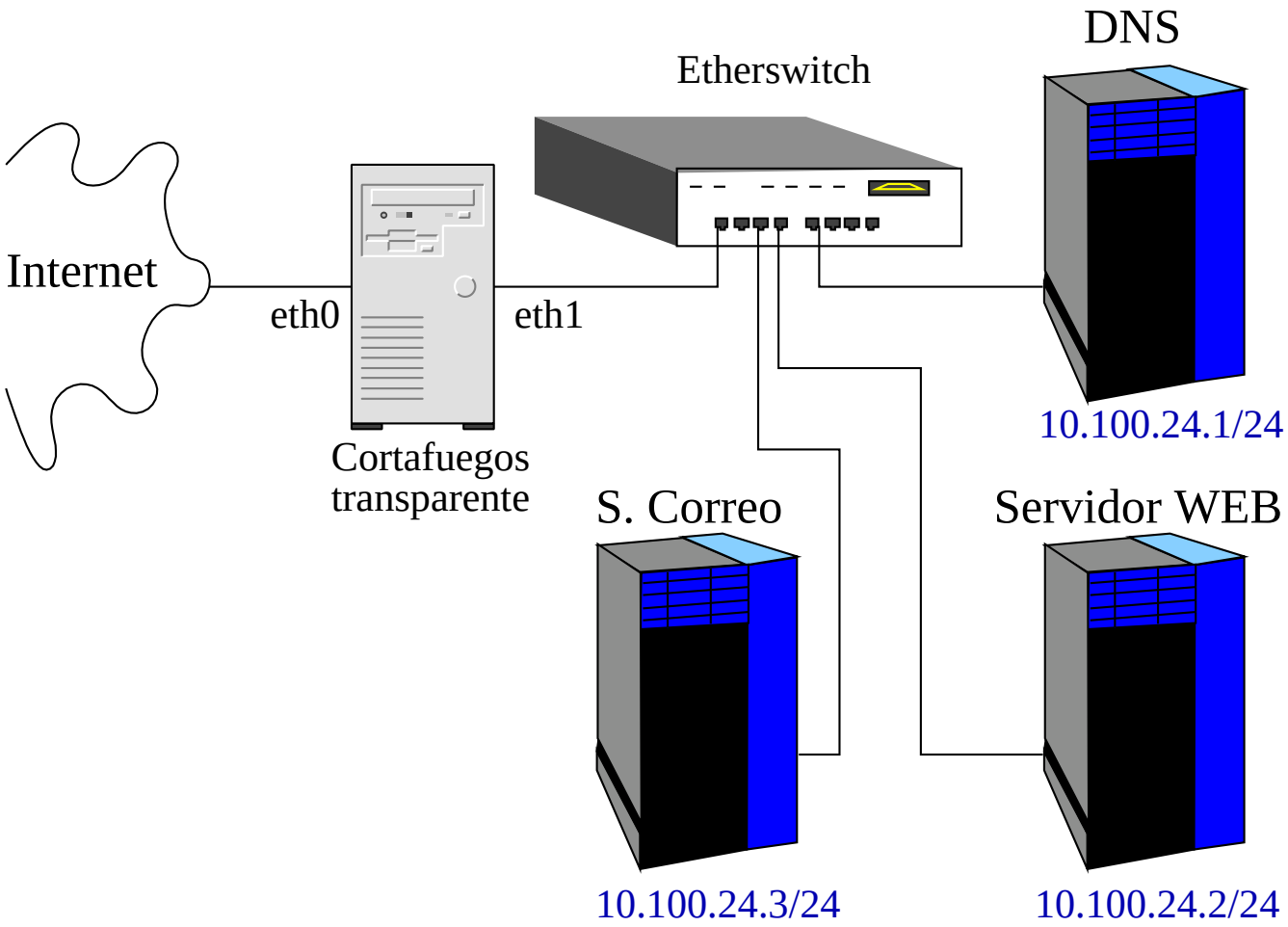
Un red con IPs públicos para todas las máquinas nos generan los siguientes problemas:

1. Los estudiantes en su trabajo de tesis se les asigna una computadora propia. Ellos instalaban servidores propios, como chat o música, que consumen todo el ancho de banda de la red.
2. Posibles fallos de los estudiantes al empezar a trabajar en redes TCP/IP afectan a toda la red.
3. Los ataques provenientes de Internet nos pone en una actitud defensiva.
4. Virus

Red militarizada



Red desmilitarizada



Flujo de paquetes en TCP

TCP, el Protocolo de Control de Transmisión, provee una entrega fiable del flujo y el servicio de conexión a las aplicaciones

1. Huésped A —— SYN(ISN) $\longrightarrow$ Huésped B
2. Huésped A $\leftarrow$ SYN(ISN+1)/ACK — Huésped B
3. Huésped A —— ACK $\longrightarrow$ Huésped B

Esto no sucede con los paquetes de UDP, los cuales se consideras "no fiables" y no intentan corregir los errores ni negociar una conexión antes del envio a un huésped remoto.

Configuración de una red TCP/IP

Dirección IP	192.168.120.21
Máscara de red	255.255.255.0
Número de red	192.168.120.
Número de huésped	.21
Dirección de Red	192.168.120.0
Dirección de Difusión	192.168.120.255

Direcciones IP *inválidas* son las especificadas en el RFC1918 para diseñar redes privadas o intranets, y son las recomendadas para usarse cuando se experimenta con redes. Estas direcciones son $10.*.*.*$, $172{,}16.*.*$—$172{,}31.*.*$ y $192{,}168.*.*$.

Seguridad en redes con TCP/IP

- Podemos bloquear los inicios de conexión
- Podemos bloquear por direcciones IP y redes
- Podemos bloquear por servicios
- Podemos bloquear por protocolo

IPTables (1/2)

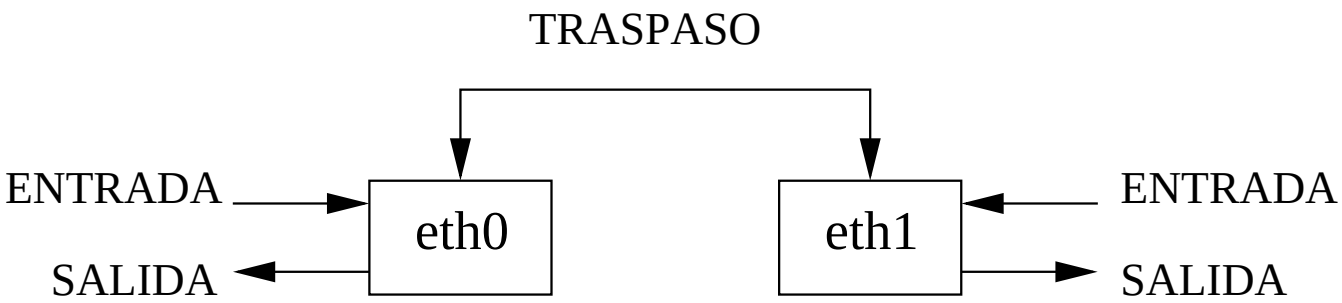
Tres cosas puede significar IPTables:

1. El software general se llama *Netfilter*, el cual provee los ganchos dentro de la pila de IP en los cuales se pueden cargar módulos que realzan operaciones sobre los paquetes.
2. IPTables viene en dos partes: los módulos en el espacio del núcleo (que son distribuidos con el mismo núcleo). El múdulo principal es `ip_table` y existen módulos específicos para NAT, log, seguimiento de conexiones, etc.
3. La segunda parte son programas en el espacio del usuario que son distribuidos de forma separada. Estos comandos pueden adicionar, remover o editar reglas en los módulos. *iptables* se refiere a este comando.

IPTables (2/2)

La manera en que se manejan (o filtran) los paquetes es insertando reglas dentro de los módulos que realizar una función determinada. Una lista de reglas es una *cadena*.

Cadenas por defecto en una interfaz de Red



Sintaxis de iptables (1/5)

- Se crea una cadena con:
  `iptables -N <nombre-de-la-cadena>`
- Se borra una cadena con:
  `iptables -X <nombre-de-la-cadena>`
- Se vacía toda una cadena:
  `iptables -F <nombre-de-la-cadena>`
- Lista las reglas de una cadena:
  `iptables -nL <nombre-de-la-cadena>`
- Especifica la meta de una cadena:
  `iptables -A <nombre-de-la-cadena> -j <nombre-de-la-cade`

Sintaxis de iptables (2/5)

- Se adiciona una regla a una cadena:
  `iptables -A <cadena> <especificacion_de_la_regla>`
- Se inserta una regla a una cadena:
  `iptables -I <cadena> [numero_de_regla] especificacion_d`
- Se borra una regla de una cadena:
  `iptables -D <cadena> [numero_de_regla]`
  `iptables -D <cadena> <especificacion_de_la_regla>`
- `-p` especifica el protocolo IP usado, puede ser TCP, ICMP, UDP o alguno de los protocolos menos usado.
- `--dport` especifica el puerto destino del paquete
- `--sport` especifica el puerto fuente del paquete. Se usa menos ya que las conexiones se originen de un puerto fuente aleatorio (arriba del 1024).

Sintaxis de iptables (3/5)

Comportamiento de la regla:

- `DROP` atrapa el paquete y lo manda al piso.
- `RETURN` para el paso del paquete en esa cadena y termina en la siguiente regla de la cadena previa (que la llama).
- `ACCEPT` deja pasar el paquete

Sintaxis de iptables (4/5)

Direcciones IP fuente y destino:

- `-s <direccion_ip>`
- `-d <direccion_ip>`
- Dirección de un huésped: `192.168.20.2/32`
- Dirección de una red: `192.168.20.0/24`
- Cualquier IP: `0.0.0.0/0`

Sintaxis de iptables (5/5)

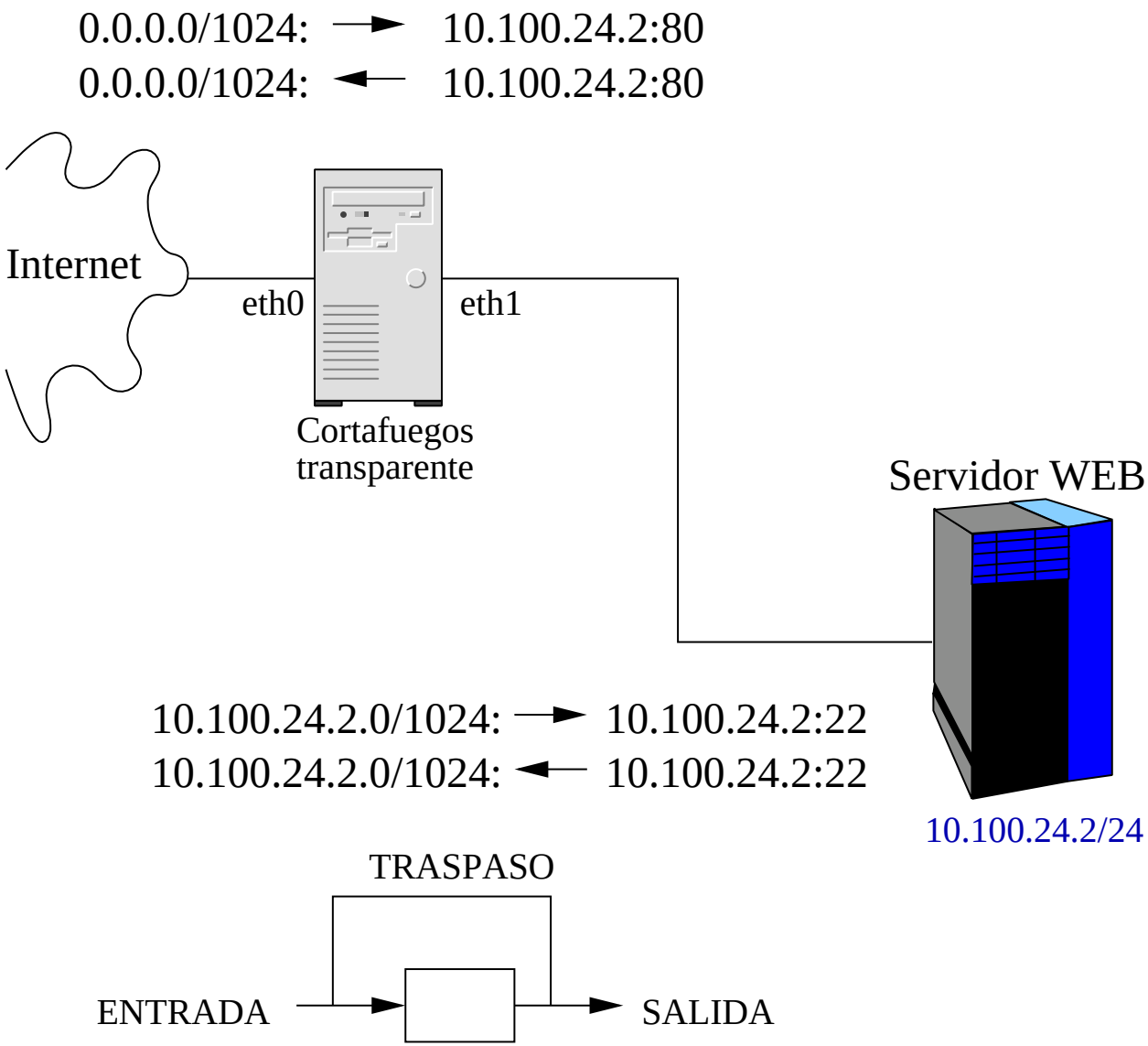
- Interfaz de entrada: `-i <nombre>`
- Interfaz de salida: `-o <nombre>`
- `-i` solo puede usarse con `INPUT`
- `-o` solo puede usarse con `OUTPUT`
- Pero `FORWARD` puede usar ambos.

Script para realizar una RM con IPtables

```
#!/bin/sh

PATH=/sbin

INTERFAZ_EXT=eth0
IPADDR=10.100.24.4
REDLOCAL=10.100.24.0/24
#
#
INTERFAZ_INT=eth1
REDINTERNA=192.168.36.0/24
#
# Limpiamos las reglas actuales
#
iptables -F
iptables -F -t nat

#---------------------------------------
# Establecer la política por defecto
#      Denegar la entrada
#      Denegar el transpaso
#      Denegar la salida
#---------------------------------------
iptables -P INPUT   DROP
iptables -P FORWARD DROP
iptables -P OUTPUT  DROP
```

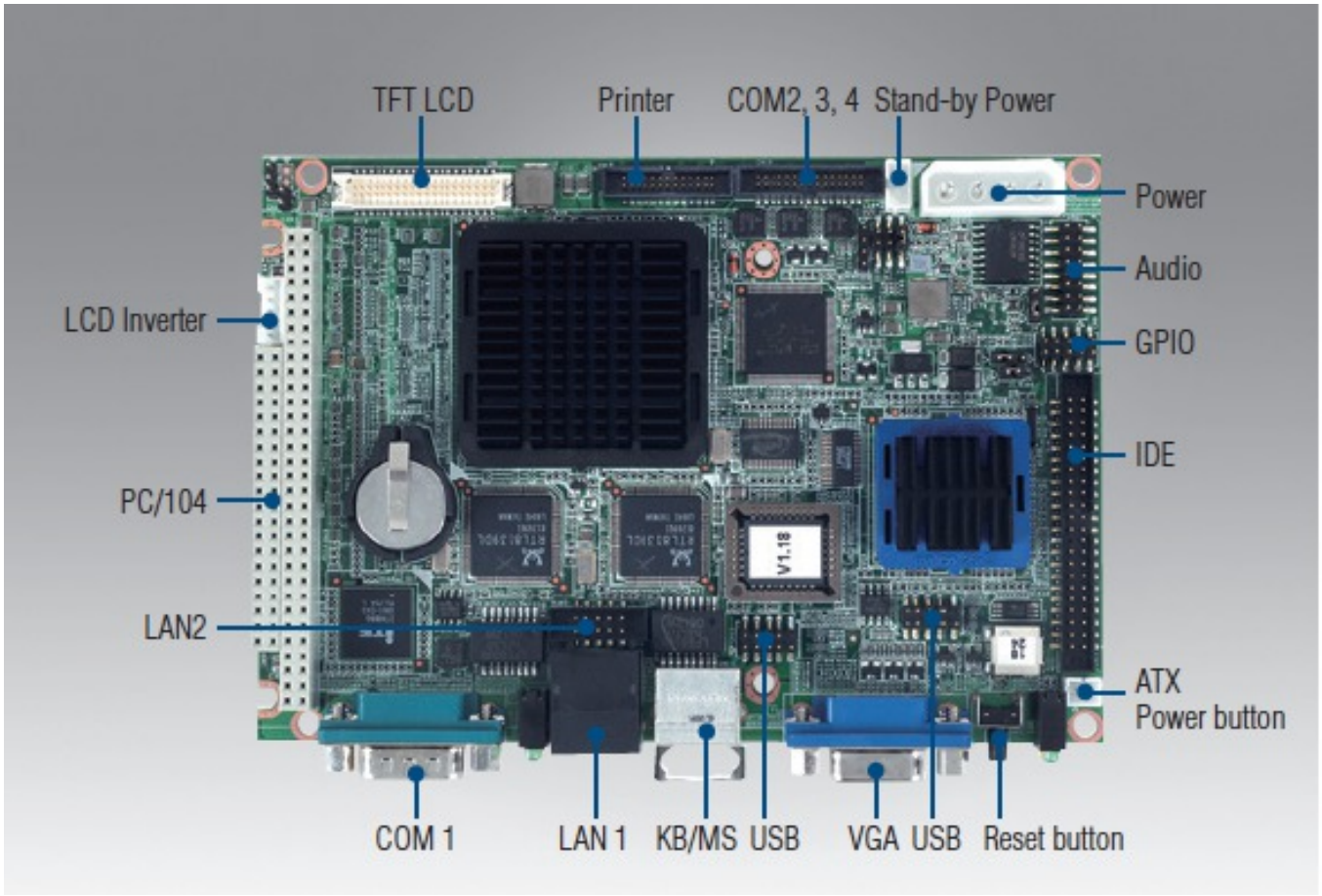
```
##############################################
# Permitimos la salida a la red interna
#
iptables -A FORWARD -m state --state NEW,ESTABLISHED \
    -i $INTERFAZ_INT -s $REDINTERNA -j ACCEPT

# Permitimos que regresen los paquetes asociados
# a estas conexiones
#
iptables -A FORWARD -m state --state ESTABLISHED,RELATED \
    -i $INTERFAZ_EXT -s ! $REDINTERNA -j ACCEPT

# Todo el tráfico interno es enmascarado externamente
#
iptables -A POSTROUTING -t nat -o $INTERFAZ_EXT -j MASQUERADE
```

Reglas en los cortafuegos

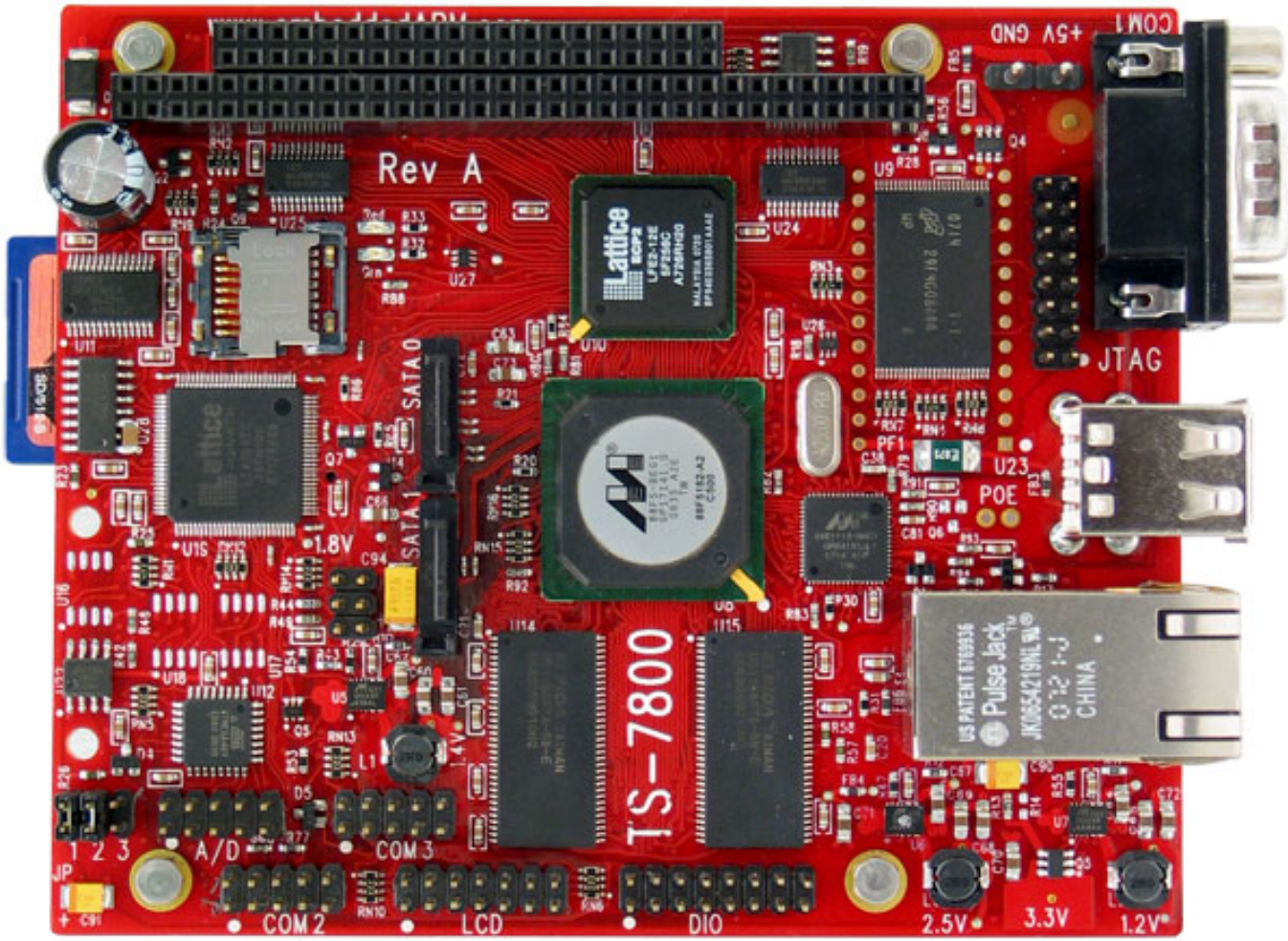


Parte 3. Una receta para realizar un cortafuegos

1. Se necesita una computadora con dos tarjetas de red
2. Debe tener el mínimo software necesario para trabajar
3. En vez de quitar software, pondremos solo el software necesario para que se ejecute el cortafuegos.



http://www.advantech.com/products/1-2JKD1I/PCM-9375/mod_E17B5F1F-B52D-4574-940C-A4F9F6892BA3.aspx



http://www.embeddedarm.com/products/board-detail.php?product=TS-7800

Necesitamos para un sistema mínimo:

1. Algún núcleo de Linux
2. Algunos módulos del mismo núcleo (no necesitamos recompilarlo)
3. Un sistema de archivos mínimo
4. Los comandos del sistema (proveídos por busybox)
5. Las bibliotecas compartidas del sistema GNU/Linux donde compilamos busybox
6. Un esfuerzo para configurar todo
7. Tenemos que arrancar nuestro nuevo sistema

Ingredientes:

- Se tiene una máquina anfitrión con alguna distribución de GNU/Linux ya instalada
- Ya se probó el cortafuegos en esa máquina anfitrión
- Se quiere realizar el sistema mínimo para que funcione el cortafuegos en ese anfitrión
- Realizar el sistema mínimo para una Computadora en usa Sola Tarjeta (CST), usando el anfitrión, también es posible.

¿Donde está en núcleo?

- El comando para conocer qué núcleo se está ejecutando:
- `uname -a`

```
Linux pegaso 2.6.43.8-1.fc15.x86_64 #1 SMP
 Mon Jun 4 20:33:44 UTC 2012 x86_64 x86_64 x86_64 GNU/Linux
```

- `ls -l /boot`

```
$ ls -l /boot
-rw-r--r--  1 root root   118373 Jun  4  2012 config-2.6.43.8-1.fc15.x86_64
-rw-r--r--  1 root root 15719477 Apr 10  2013 initramfs-2.6.43.8-1.fc15.x86_64.
-rw-------  1 root root  2447575 Jun  4  2012 System.map-2.6.43.8-1.fc15.x86_64
-rwxr-xr-x  1 root root  4753184 Jun  4  2012 vmlinuz-2.6.43.8-1.fc15.x86_64
```

- `file /boot/vmlinuz-2.6.43.8-1.fc15.x86_64`

```
/boot/vmlinuz-2.6.43.8-1.fc15.x86_64: Linux kernel
   x86 boot executable bzImage, version 2.6.43.8-1.fc15.x86_64
  (mockbuild@x86-02.phx2.fedoraproject.org, RO-rootFS,
   swap_dev 0x4, Normal VGA
```

Construimos nuestro sistema de archivos mínimo

Aquí se muestra un conjunto mínimo de directorios para el sistema de archivos raíz:

- /dev Archivos de dispositivos, requeridos para E/S
- /proc Directorio requerido para el sistema de archivos /proc (variables de estado del núcleo)
- /etc Archivos de configuración del sistema
- /sbin Binarios críticos del sistema
- /bin Binarios del sistema (busybox)
- /mnt Un punto de montaje para otros discos
- /usr Utilerias adicionales y aplicaciones

Busybox

- Nos provee los comandos mínimos para el cortafuegos
- Todos los comandos de Linux están empotrados en un solo archivo ejecutable
- Está disponible en `http://www.busybox.net`
- Hay que bajarlo, compilarlo e instalarlo sobre nuestro sistema de archivos mínimo

Las bibliotecas compartidas:

```
$ LD_DEBUG=libs /bin/ls
      26794: find library=libselinux.so.1 [0]; searching
      26794:  search cache=/etc/ld.so.cache
      26794:   trying file=/lib64/libselinux.so.1
      26794:
      26794: find library=librt.so.1 [0]; searching
      26794:  search cache=/etc/ld.so.cache
      26794:   trying file=/lib64/librt.so.1
      26794:
      26794: find library=libcap.so.2 [0]; searching
      26794:  search cache=/etc/ld.so.cache
      26794:   trying file=/lib64/libcap.so.2
```

Las bibliotecas compartidas:

```
$ ldd /bin/ls
linux-vdso.so.1 =>  (0x00007ffff1d98000)
libselinux.so.1 => /lib64/libselinux.so.1 (0x00000035e9c00000)
librt.so.1 => /lib64/librt.so.1 (0x00000035e8c00000)
libcap.so.2 => /lib64/libcap.so.2 (0x00000035ea000000)
libacl.so.1 => /lib64/libacl.so.1 (0x00000035f6400000)
libc.so.6 => /lib64/libc.so.6 (0x00000035e7c00000)
libdl.so.2 => /lib64/libdl.so.2 (0x00000035e8000000)
/lib64/ld-linux-x86-64.so.2 (0x00000035e7800000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00000035e8400000)
libattr.so.1 => /lib64/libattr.so.1 (0x00000035f3400000)
```

- Con el comando ldd o con la bandera de debug habilitada (LD_DEBUG) sabemos las bibliotecas compartidas que se pondrán en el directorio /lib del sistema de archivo mínimo

- Hay que cambiar la siguiente liga en busybox `./linuxrc` a `init`
- Hay que encender la bandera del setuid de busybox `chmod +s ./bin/busybox`

Las siguientes dispositivos son necesarios:

```
/dev/null
/dev/mem
/dev/ram0
/dev/ram1
/dev/tty1
/dev/tty2
/dev/tty3
/dev/console
/dev/loop0
/dev/ptmx
/dev/systty
/dev/zero
/dev/initctl
/dev/pts
/dev/shm
/dev/stderr
/dev/stdin
/dev/stdout
```

```
ls /etc
fstab
group
inittab
passwd
protocols
rc
securetty
shells
```

El contenido del archivo /etc/inittab

```
tty1::respawn:/sbin/getty 38400 tty1
::sysinit:/etc/rc

# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

Más configuraciones:

- Levantar el sistema de archivos /proc
- Configurar el traspaso de paquetes:
  echo 1 > /proc/sys/net/ipv4/ip_forward
- Copiar los módulos necesarios para activar las tarjetas de red
- Configurar las tarjetas de red (ifconfig)
- Arrancar el cortafuegos

El sistema de archivos para que pueda sobrevivir Linux lo creamos como:

```
cd SisMinimo
find . | cpio -o -H newc > SistemaMin.cpio
gzip Sistema.cpio
```

En el directorio `SisMinimo` se encuentra la estructura de directorios de nuestro sistema mínimo.

En este punto ya tenemos:

- Un núcleo seleccionado
- Busybox instalado junto con las bibliotecas compartidas (dentro del archivo Sistema.cpio.gz)
- ¡Solo falta arrancar!

Para configurar el arranque desde el disco duro (una prueba) con grub:

```
title Mi Sistema Minimo
    kernel /vmlinuz-2.6.17.2 rw root=/dev/ram0 init=/sbin/init
    initrd /miImagen.gz
```

El proceso de arranque es el siguiente:

- Se ejecuta el núcleo
- El núcleo configura todo el hardware
- Al final, el sistema de archivo proveído se descomprime en memoria
- Ahora se está ejecutando el núcleo con el sistema de archivos (y todo su contenido) en memoria

Podríamos necesitar configurar también:

- dnsmasq (un servidor DNS y también DHCP)
- dropbear (un servidor SSH)

Conclusiones

- Vimos las inseguridades de TCP/IP
- Vimos como crear un cortafuegos con iptables
- Vimos como contruir un cortafuegos con el software mínimo

¡Gracias!