Iptables: un cortafuegos TCP/IP

IES Gonzalo Nazareno
CONSEJERÍA DE EDUCACIÓN

Jesús Moreno León
Alberto Molina Coballes

Redes de Área Local

Noviembre 2009

Estas diapositivas son una obra derivada de la documentación de **Juan Céspedes**, profesor del Grupo de Sistemas y Comunicaciones de la Universidad Rey Juan Carlos, para la asignatura Diseño y Administración de Sistemas y Redes

1. Introducción

- El kernel de Linux incorpora varias funciones integradas que le permiten trabajar como un firewall TCP/IP
  - Incluye código (el framework Netfilter) que permite interceptar y manipular paquetes de red
  - Presenta una herramienta en el espacio de usuario (iptables) que el administrador puede usar para establecer las reglas del firewall

2. iptables

- iptables se usa para crear, mantener y revisar las tablas de filtrado de paquetes en el kernel de Linux
  - Existen diferentes tablas (tables) dentro de las cuales puede haber varias cadenas (chains)
  - Dentro de cada tabla, existen varias cadenas predefinidas y puede haber otras definidas por el usuario
  - Cada cadena consiste en una lista de reglas con las que se comparan los paquetes. Las reglas especifican qué se hace con los paquetes que se ajustan a ellas (target)

2.1. targets

- Cuando un paquete no se ajusta a una regla, se continúa examinando la siguiente hasta que se ajusta con alguna. En ese momento se examina el target:
  - DROP: el paquete se descarta, no puede pasar
  - ACCEPT: el paquete continúa su camino normal
  - ...

- Si se llega al final de una chain predefinida se ejecuta un target por defecto, llamado chain policy

2.2. Tablas

- Existen tres tablas (filter, nat y mangle)
  - filter: es la tabla por defecto y se utiliza para especificar filtros de paquetes. Contiene 3 chains predefinidas
    - INPUT: Se consulta para los paquetes cuyo destino es la propia máquina
    - FORWARD: La atraviesan los paquetes rutados a través de esta máquina (tanto origen como destino son externos)
    - OUTPUT: Para paquetes generados localmente

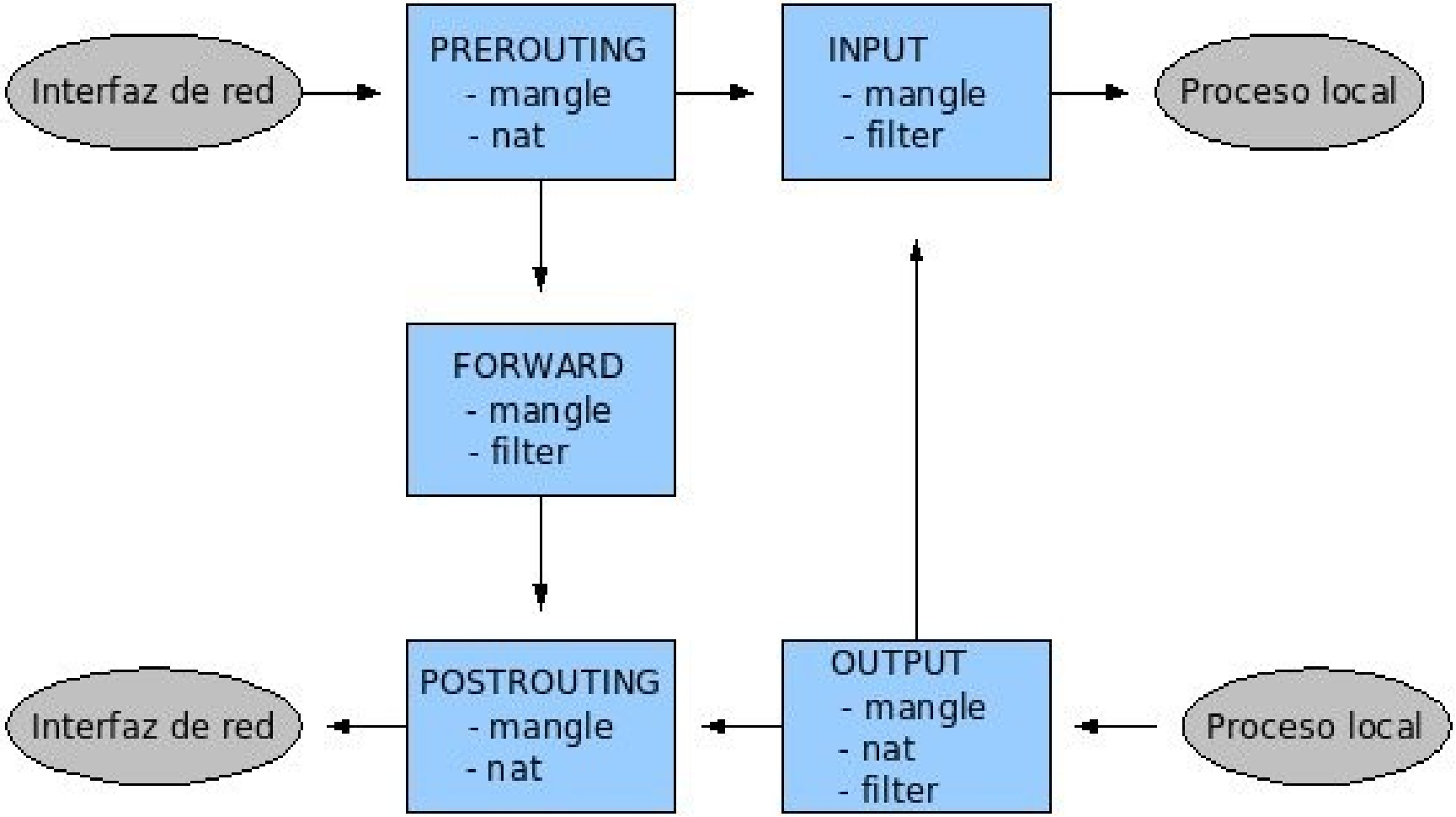
2.2. Tablas

- nat: esta tabla se consulta cada vez que se ve un paquete que inicia una nueva conexión, con el objetivo de alterar algún parámetro de esa conexión. Tiene 3 chains predefinidas:
  - PREROUTING: se consulta con los paquetes que entran en la máquina, tan pronto como llegan, antes de decidir qué hacer con ellos
  - OUTPUT: se utiliza para alterar paquetes generados localmente, antes de rutarlos
  - POSTROUTING: para alterar los paquetes que están a punto de salir de la máquina

2.2. Tablas

- mangle: tabla especial, destinada a alterar determinados parámetros de los paquetes (TOS, TTL ...)
  - PREROUTING
  - INPUT
  - FORWARD
  - OUTPUT
  - POSTROUTING

2.3. Flujo de paquetes



2.4. Listado de reglas

```
#iptables [-t tabla] [opciones] -L [chain]
```

- Muestra un listado de todas las reglas de una chain, o de todas ellas
- Opciones disponibles:
  - -v: información detallada
  - -n: salida numérica
  - -x: valor exacto de cada número

2.5. Borrado de contadores

```
#iptables [-t tabla] -Z [chain]
```

- Borra los contadores de una determinada chain, o de todas ellas
- Se puede combinar con la opción -L para mostrar la información justo antes de borrarla

2.6. Borrado de reglas

```
#iptables [-t tabla] -F [chain]
```

- Borra las reglas de una determinada chain, o de todas ellas

2.7. Creado y borrado de reglas

```
#iptables [-t table] -A chain rule-spec
#iptables [-t table] -I chain [rulenum] rule-spec
#iptables [-t table] -R chain rulenum rule-spec
#iptables [-t table] -D chain rule-spec
#iptables [-t table] -D chain rulenum
```

- -A: añade una regla al final de la lista
- -I: inserta una regla al comienzo de la lista o en el punto especificado
- -R: reemplaza una regla (especificada por su número de regla) por otra
- -D: borra una regla determinada

2.8. Especificación de reglas

- Con las opciones -A, -I, -R, y -D de iptables se tiene que especificar una regla:
  - `-p [!] protocolo`: el protocolo del paquete a comprobar. Puede ser 'tcp', 'udp', 'icmp' o 'all'
  - `-[sd] [!] dirección[/máscara]`: dirección ip origen (s) o destino (d) del paquete
  - `-[io] [!] iface`: nombre de la interfaz de entrada (i) o de salida (o) del paquete
  - `-j target`: especifica el target de dicha regla. Puede ser una acción predefinida (ACCEPT, DROP), una extensión o el nombre de una chain

2.9. Establecer política por defecto

```
#iptables [-t table] -P chain target
```

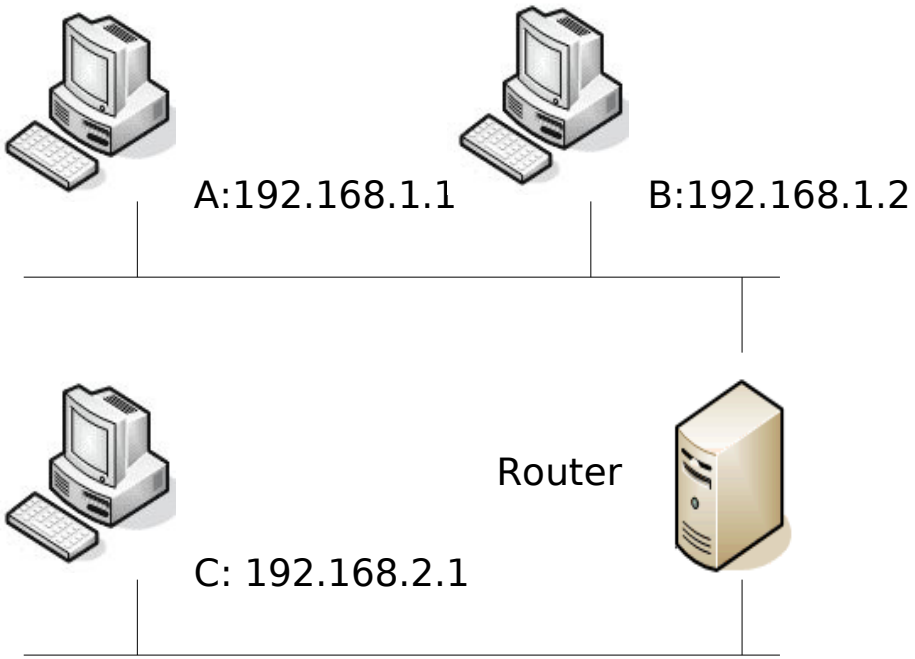
- Establece el taget que se ejecutará para los paquetes que no cumplan con ninguna regla de la chain especificada

2.10. Obtener ayuda

```
#iptables -h
#man iptables
```

2.11. Un ejemplo sencillo

- Queremos que sólo la máquina B pueda hablar con C, y sólo usando el protocolo TCP
- Los paquetes que no cumplan estas condiciones serán descartados



2.11. Un ejemplo sencillo

```
#iptables -F FORWARD
#iptables -P FORWARD DROP
#iptables -t filter -A FORWARD -p tcp -s 192.168.1.2
          -d 192.168.2.1 -j ACCEPT
#iptables -t filter -A FORWARD -p tcp -s 192.168.2.1
          -d 192.168.1.2 -j ACCEPT
```

Ejemplo. ¿Cómo se lee la tercera línea ?

Componente	Descripción
-t filter	Vamos a trabajar con la tabla filter (tabla por defecto)...
-A FORWARD	... añadiendo la siguiente regla a su cadena FORWARD.
-p tcp	Selecciona los paquetes cuyo protocolo sea TCP...
-s 192.168.1.2	...cuya dirección origen sea 192.168.1.2...
-d 192.168.2.1	... y cuya dirección destino sea 192.168.2.1.
-j ACCEPT	Acepta esos paquetes para su reenvío.

2.11. Un ejemplo sencillo

- En vez de escribir los comandos uno tras otro, podríamos crear un script de shell

```
#!/bin/bash
##Script de iptables - Un ejemplo sencillo

##Borramos las reglas de la chain FORWARD de la tabla
filter
iptables -F FORWARD

##Establecemos la política por defecto -> DROP
iptables -P FORWARD DROP

##Aceptamos los paquetes TCP entre B y C
iptables -t filter -A FORWARD -p tcp -s 192.168.1.2
         -d 192.168.2.1 -j ACCEPT
iptables -t filter -A FORWARD -p tcp -s 192.168.2.1
         -d 192.168.1.2 -j ACCEPT
```

2.12. Extensiones de las reglas

- Con las opciones vistas podemos controlar los parámetros más básicos de la cabecera IP del paquete
- Puede ser necesario un control más estricto, por ello existen una serie de extensiones que permiten utilizar opciones nuevas
- `-m extensión`: activa una extensión para poder especificar los parámetros del paquete

2.12. Extensiones de las reglas

- `tcp`: añade las siguientes opciones
  - `--sport [!] port[:port]`: especifica el puerto o rango de puertos origen
  - `--dport [!] port[:port]`: especifica el puerto o rango de puertos destino
  - `[!] --syn`: la regla concordará sólo con los paquetes cuyo bit SYN=1 y los flags ACK y FIN valgan 0. Los datagramas con estos valores se utilizan para abrir las conexiones TCP

2.12. Extensiones de las reglas

- `udp`: añade las siguientes opciones
  - `--sport [!] port[:port]`: especifica el puerto o rango de puertos origen
  - `--dport [!] port[:port]`: especifica el puerto o rango de puertos destino

2.12. Extensiones de las reglas

- `icmp`: añade la opción `--icmp-type tipo`, que indica qué tipo ICMP debe tener el paquete (echo-request, echo-reply, network-unreachable...)
- `mac`: añade la opción `--mac-source [!] dir_mac`, que especifica la dirección MAC que debe tener el paquete

2.12. Extensiones de las reglas

- `state`: Añade la opcion `--state valor`, que indica el estado en el que debe estar la conexion correspondiente a dicho paquete. Los tipos de estado principales son:
  - `NEW`: el paquete corresponde a una conexión nueva
  - `ESTABLISHED`: el paquete está asociado a una conexión ya establecida
  - `RELATED`: el paquete corresponde a una conexión nueva, pero relacionada con una que ya está establecida (como un canal de datos FTP o un error de ICMP)

2.13. Extensiones de target

- Aparte de las target predefinidas se pueden usar otras. Por ejemplo, las siguientes sólo pueden usarse en la tabla nat:
  - `MASQUERADE`: sólo es válida en la chain POSTROUTING. Indica que la dirección origen del paquete (y de todos los de esa conexión) será cambiada por la IP local de esta máquina. Muy útil para IP dinámica (es lo que se conoce como NAT)

//Ejemplo de Masquerade (NAT)

```
# iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

2.13. Extensiones de target

- `SNAT`: sólo es válida en la chain POSTROUTING. Indica que la dirección y puerto origen del paquete (y de todos los de esa conexión) sea modificada según se especifica con la opción `--to-source`
- El TARGET será SNAT (en lugar de MASQUERADE) cuando tengamos una IP fija (no dinámica)

//Ejemplo de SNAT

```
# iptables -t nat -A POSTROUTING -o eth1 -j SNAT
           --to-source 189.29.35.15
```

2.13. Extensiones de target

- `DNAT`: sólo es válida en las chains PREROUTING y OUTPUT. Cambia la dirección IP destino (y de todos los futuros de esta misma conexión) por el especificado con la opción --`to-destination`
- Es lo que se conoce como 'abrir el puerto'

//Ejemplo de DNAT

```
# iptables -t nat -A PREROUTING -i eth0 -p tcp
--dport 80 -j DNAT --to-destination 192.168.1.2:80
```

3. Conclusión

- Con iptables podemos implementar un firewall que nos permite realizar filtro de paquetes, traducción de direcciones y mantener registros de log
- iptables trabaja a nivel 3 y 4, por tanto los criterios para filtrar un paquete no pueden incluir condiciones de nivel de aplicación. Sería necesario usar un servidor proxy para, por ejemplo, filtrar cierto contenido web

4. Referencias

- Linux Network Administrator's Guide. Tony Bautts, Terry Dawson & Gregor N. Purdy (O'REILLY)
- Existe mucha documentación en Internet
  - En este enlace hay muchos scripts para diferentes configuraciones (FW de perímetro, DMZ...) http://www.linuxguruz.com/iptables/
  - Os recomiendo el manual que podéis encontrar en http://www.pello.info