Firewall en GNU/Linux

NETFILTER/IPTABLES

SEGURIDAD EN SISTEMAS INFORMATICOS
4º Grado en Ing. Informática

http://ccia.ei.uvigo.es/docencia/SSI-grado/

6 de noviembre de 2012

1. Introducción a `netfilter/iptables`

- NETFILTER (`http://www.netfilter.org`) es un componente del núcleo Linux (desde la versión 2.4) encargado de la manipulación de paquetes de red, que permite:
  - filtrado de paquetes
  - traducción de direcciones (NAT)
  - modificación de paquetes
- `iptables` es una herramienta/aplicación (forma parte del proyecto NETFILTER) que hace uso de la infraestructura que ofrece NETFILTER para construir y configurar *firewalls*
  - permite definir políticas de filtrado, de NAT y realizar logs
  - remplaza a herramientas anteriores: IFWADMIN, IPCHAINS
  - puede usar las capacidades de seguimiento de conexiones NETFILTER para definir *firewalls con estado*
- Las posibles tareas a realizar sobre los paquetes (filtrado, NAT, modificación) se controlan mediante distintos conjuntos de reglas, en función de la situación/momento en la que se encuentre un paquete durante su procesamiento dentro de NETFILTER.
  - Las listas de reglas y demás datos residen en el espacio de memoria del kernel
  - La herramienta de nivel de usuario `iptables` permite al administrador configurar las listas de reglas que usa el kernel para decidir qué hacer con los paquetes de red que maneja.
  - En la práctica un *"firewall"* IPTABLES consistirá en un script de shell conteniendo los comandos `iptables` para configurar convenientemente las listas de reglas.
    - tipicamente ese script residirá en el directorio `'/etc/init.d'` ó en `'/etc/rc.d'` para que sea ejecutado cada vez que arranca el sistema
  - Otras utilidades (guardar/recuperar reglas en memoria): `iptables-save`, `iptable-restore`

(a) Elementos

TABLAS. Se corresponden con los distintos **tipos de procesamiento** que se pueden aplicar sobre los paquetes.

Tablas disponibles:

filter. Controla decisiones de filtrado de paquetes (aceptar/denegar)
Cadenas: INPUT, OUTPUT, FORWARD

nat. Controla traducción de direcciones (NAT:*network address translation*)
Cadenas: PREROUTING, POSTROUTING (opc. OUTPUT)

mangle. Controla los procesos de modificación del contenido y las opciones de los paquetes.
Cadenas: INPUT, OUTPUT, FORWARD, PREROUTING, POSTROUTING

Las reglas de cada *tabla* se organizan en *cadenas*, que se consultarán en momentos concretos del flujo de los paquetes..

CADENAS. Contienen las **listas de reglas** a aplicar sobre los paquetes
Cadenas predeterminadas: (asociadas a momentos concretos del flujo de los paquetes)

INPUT reglas a aplicar sobre los paquetes destinados a la propia máquina, (justo antes de pasarlos a las aplicaciones)
Usada para controlar las entradas al propio equipo/cortafuegos

OUTPUT reglas a aplicar sobre los paquetes originados en la propia máquina, (justo después de recibirlas desde las aplicaciones)
Usada para controlar las salidas del propio equipo/cortafuegos

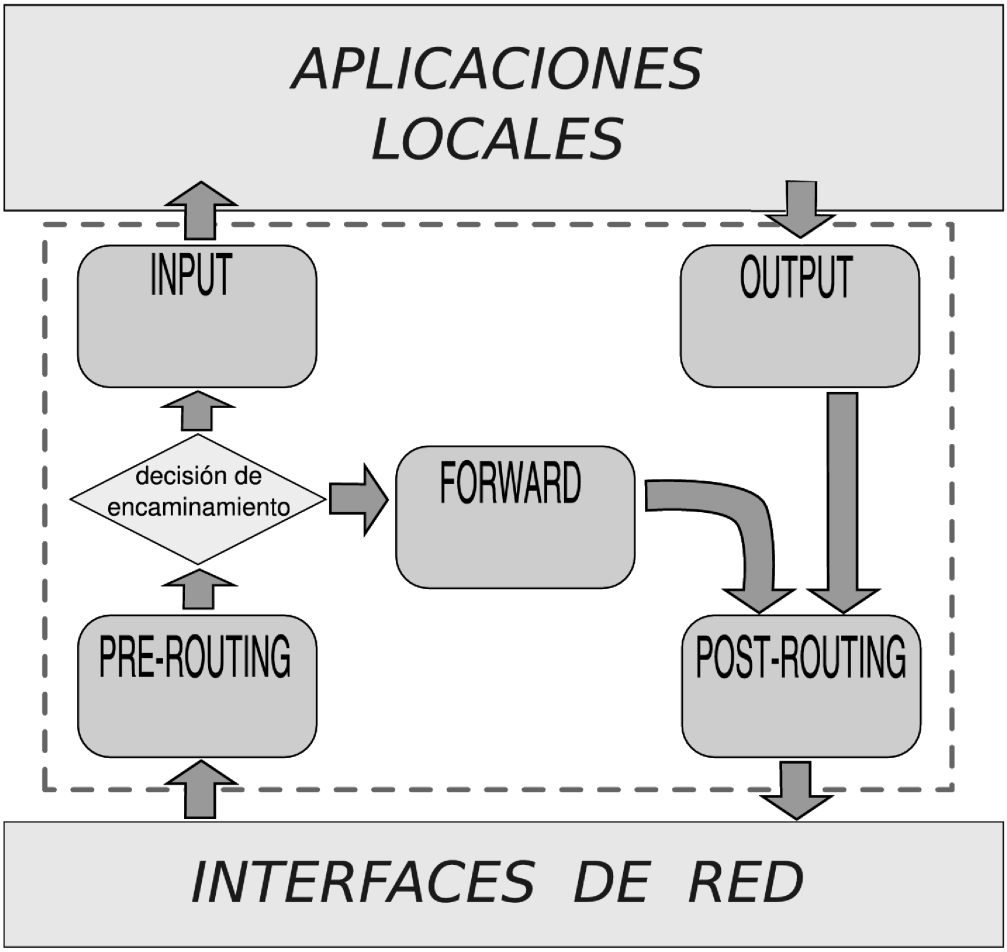
FORWARD reglas a aplicar sobre los paquetes que atraviesan la máquina con destino a otras (paquetes en tránsito reenviados)
Usadas en Cortafuegos de borde (protección red interna)

PRE-ROUTING reglas a aplicar sobre paquetes justo antes de enviarlos a la red
Usada para DNAT (*destination NAT*) [redirección de puertos]

POST-ROUTING reglas aplicar sobre paquetes (propios o ajenos) recibidos de la red (antes de decidir a dónde encaminarlos [local o reenvío])
Usada para SNAT (*source NAT*) [enmascaramiento]

Se pueden crear cadenas definidas por el usuario (comando "`iptables -N cadena`"), a las que se accederá desde reglas incluidas en alguna de las cadenas predeterminadas ($\approx$ subrutinas)

(b) Flujo de paquetes a través de NETFILTER



(c) Funcionamiento

- Para cada paquete, en función del procesamiento que vaya a sufrir, se consulta la *cadena* que corresponda a su *situción* dentro de NETFILTER.
- Dentro de cada *cadena* las reglas se inspeccionan secuencialmente [orden de reglas importante]
- Si el paquete encaja con las condiciones de una regla, se ejecuta la acción correspondiente y se abandona la *cadena*
- Si el paquete no encaja con ninguna regla, se le aplica la **política por defecto** que se haya asignado a esa *cadena*.
  - por defecto las cadenas predeterminadas están inicializadas con una política ACCEPT
  - al agotar las cadenas definidas por el usuario, se retorna a la cadena predeterminada que la activó

2. Reglas `iptables`

(a) Esquema

iptables *-t tabla* COMANDO CONDICIONES OBJETIVO

Nota: Si no se indica una tabla, se usa por defecto *-t filter*

(b) Comandos más relevantes

`-L/--list cadena` lista las reglas actualmente en uso en una cadena

`-F/--flush cadena` vacía una cadena

`-Z/--zero cadena` reinicia los contadores de una cadena

`-P/--policy cadena DROP/ACCEPT` establece la política por defecto

`-A/--append cadena` añade una regla (condiciones+objetivos) a una cadena

`-D/--delete cadena` borra una regla de una cadena

`-R/--replace cadena` reemplaza una regla de una cadena

`-I/--insert cadena` inserta una regla de una cadena

(c) Condiciones

dirección IP { origen: `-s/--source` / destino: `-d/--destination`

Puede ser una dirección (`-s 193.147.87.47`) o un rango (`-s 193.147.87.0/24`)

interfaz { de salida: `-i/--in-interface` (en INPUT,FORWARD,PREROUTING) / de entrada: `-o/--out-interface` (en OUTPUT,FORWARD,POSTROUTING)

Dispositivo (*eth0,eth1,...,lo*) por el que se ha recibido el paquete o por el que saldrá el paquete.

tipo protocolo `-p/--protocol`

Protocolo de nivel de transporte: `tcp,udp,icmp,all`

puertos { origen: `-sport/--source-port` / destino: `-dport/--destination-port`

Puede ser un número de puerto (`-sport 80`), un nombre de servicio (`-sport http`) o un rango de puertos (`-sport 1024:65535`)

control estado conexión `-m state --state` ESTADOS

Soporte básico para reglas de filtrado *"con estado"*.

Situación del paquete respecto a la conexión a la que pertenece:

INVALID paquete no asociado a una conexión conocida

ESTABLISHED paquete que pertenece a una conexión válida ya establecida

NEW paquete mediante el cual se está creando una nueva conexión

RELATED paquetes que inician una nueva conexión que está asociada con otra ya establecida

Filtros con estado más detallados: `-m conntrack --ctstate` ESTADOS

Nota: pueden inspeccionarse las tablas de seguimiento de conexiones en `/proc/net/ip_conntrack`

otros
- `--tcp-flag` (bits de paquetes TCP)
- `-m --mac-source` (direcciones MAC [en redes ethernet])
- `-m limit (--limit | --limit-burst` (límites pqts./seg.)

Para expresar la negación de una restricción se antepone un "`!`".

```
-sport ! 80        -p ! icmp        -s !193.147.87.47
```

(d) Objetivos

Objetivos predefinidos (tabla *-t filter*) [en cadenas INPUT,OUTPUT,FORWARD]

`-j` ACCEPT. el paquete se acepta y se deja de recorrer la cadena

`-j` DROP. se rechaza el paquete (sin informar al origen)

`-j` REJECT `--reject-with`. se rechaza el paquete, informando al origen con el mensaje ICMP indicado

Objetivos predefinidos (tabla *-t nat*)

`-j` SNAT `--to-source`. Realiza SNAT(*source*-NAT) sobre los paquetes salientes *[enmascaramiento de direcciones]*

Cambia dir. IP (opc. puerto) de origen del paquete (sólo disponible en POSTROUTING)

`-j` MASQUERADE. IDEM que SNAT, pero usando la dir. IP del propio equipo (útil en conexiones volátiles [ADSL, modem]) (sólo disponible en POSTROUTING)

`-j` DNAT `--to-destination`. Realiza DNAT(*destination*-NAT) sobre los paquetes entrantes *[redireccionamiento de puertos]*

Cambia dir. IP (opc. puerto) de destino del paquete (sólo disponible en PREROUTING y opc. en OUTPUT)

Objetivos de log (no detienen el recorrido por la cadena)

`-j` LOG. crea entrada en el log del sistema [`/var/log/sysylog`]

`-j` ULOG. crea entrada en un log definido por usuario

3. Interfaces, herramientas e información

Interfaces gráficos

- Firestarter: `http://www.fs-security.com/`
- Firewall Builder: `http://www.fwbuilder.org/`
- KMyFirewall: `http://www.kmyfirewall.org/`
- Guardog: `http://www.simonzone.com/software/guarddog/`

Herramientas en modo texto

- Shorewall (Shoreline Firewall):
- FireHOL: `http://firehol.sourceforge.net/`
- Dwall: `http://dag.wieers.com/home-made/dwall/`

Fuentes de información

- **Más detalles:** `iptables --help`, `man iptables`
- Página del proyecto: `http://www.netfilter.org/documentation/`
- Tutorial detallado: `http://iptables-tutorial.frozentux.net/`
- Tutorial práctico en español: `http://www.pello.info/filez/firewall`

Módulos

NETFILTER tiene una arquitectura modular que permite la inclusión de nuevos componentes

- Es posible incluir nuevas condiciones y objetivos que permitan hacer distintos tipos de controles/procesamiento sobre los paquetes.
- `http://xtables-addons.sourceforge.net/`
- Ejemplos

  geoip permite identificar el origen geográfico de un paquete en base a su dirección IP
  lpp2p aporta capacidades para reconocer algunos tipos de tráfico P2P (kazaa, edk, etc)
  string permite emplear reglas que analicen cadenas de texto presentes en la carga útil de los paquetes procesados