

Charla de redes

Administración de redes sobre GNU/Linux

Carlos Hernando
chernando@acm.org

ACM Facultad de Informática
Universidad Politécnica de Madrid

19 de noviembre de 2007
Noviembre Linuxero 07



Contenido

1 Netfilter

- Conceptos generales
- Filtros
- NAT

2 Herramientas

- netstat
- netcat
- nmap



Motivación

Queremos conseguir:

- Control
Determinar el comportamiento de nuestro sistema.
- Seguridad
Proteger nuestra red.
- Vigilancia
Saber lo que pasa.

Lo aplicamos en:

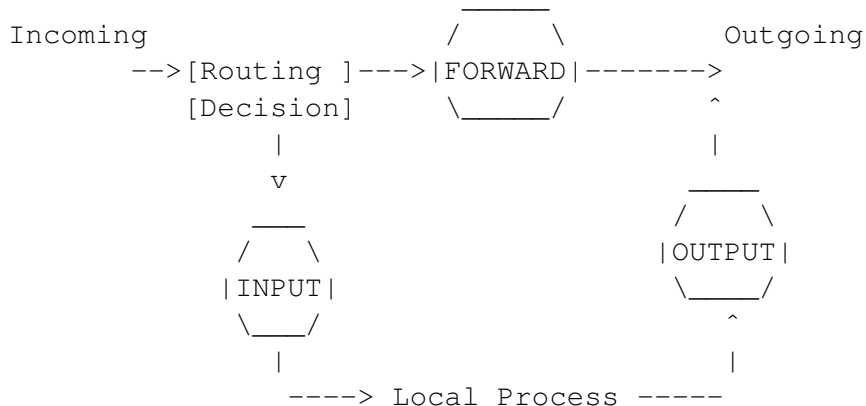
- Máquinas solitarias.
- Encaminadores de redes locales.
- En DMZs.
- ...

Exige un conocimiento profundo de nuestra red y sus servicios.



Conceptos generales

Recorrido de un paquete



Conceptos generales (2)

Targets (Objetivos)

Acción a realizar con un paquete:

ACCEPT Aceptar el paquete.

DROP Ignorar el paquete.

REJECT Rechazar el paquete (**icmp: port unreachable**).

LOG Registrar el paquete.

cadena Enviar a otra cadena (definida por nosotros).

RETURN Volver a la cadena anterior.

...



Conceptos generales (y 3)

Trabajando con Netfilter

- Trabajamos sobre cadenas.
INPUT, OUTPUT, FORWARD...
- Herramientas (zona de usuario):
 - `iptables` Programa principal de gestión de filtros.
 - `iptables-save` Guarda la tabla de filtros actual.
 - `iptables-restore` Carga una tabla de filtros.



Lo más básico

Manejo de *chains*

Operaciones sobre cadenas:

- N Crear una nueva cadena.
- X Borrar una cadena (**sin reglas**).
- P Definir política por defecto (**cadenas base**).
- L Listar reglas asignadas a cadena cadena.
- F Limpiar la cadena.
- Z Reiniciar contadores.

Example

```
iptables -N sospechoso
```

```
iptables -P FORWARD DROP
```



Lo más básico (y 2)

Manejo de *rules*

Sintaxis

```
iptables comando cadena filtros accion
```

- A Añadir una regla (al final de la cadena).
- D Borrar una regla.
- I Insertar una regla **en una posición**.
- R Remplazar una regla por otra.

Example

```
iptables -A sospechoso -j DROP
```



Consejo

Hay que tener en cuenta:

- Las reglas se aplican de forma consecutiva hasta que se cumple una especificación: el orden **influye**.
- La aplicación de una regla es inmediata.
- Si enviamos un paquete a una cadena en la que no se determina si se acepta o se rechaza el paquete vuelve a la cadena de origen.

Ojo con no pillarse los dedos!



Especificación básica

-s Origen

-s 192.168.0.0/24

-d Destino

-d ! 192.168.0.0/255.255.255.0

-p Protocolo

-p icmp

-i Interfaz de origen (*in-interface* **INPUT**)

-i eth0

-o Interfaz de salida (*out-interface* **OUTPUT**)

-o ppp+

Example

```
iptables -A INPUT -i eth0 -d 10.0.0.0/24 -j DROP
```

```
iptables -A OUTPUT -o eth1 -p icmp -d !192.168.0.1 -j DROP
```

Jugando con TCP

- `-sport` Puerto de origen.
- `-dport` Puerto de destino.
- `-tcp-flags` Flags del paquete:
SYN, ACK, FIN, RST, URG, PSH, NONE, ALL
- `-syn` Equivale a: *SYN,RST,ACK SYN*.

Example

```
iptables -A INPUT -p tcp -dport 22 -j ACCEPT
iptables -P INPUT DROP
```



Otras extensiones

Cargados simplemente con **-p** **protocol**:

- **udp**: *sport, dport*.
- **icmp**: *icmp-type*: `icmp-type echo-request`.

Necesitan **-m** **modulo**:

- **state**: *state* NEW, ESTABLISHED, RELATED, INVALID.
- **mac**: *mac-source*.
- **limit**: *limit* 1/s.
- **owner**: *uid-owner, gid-owner, pid-owner*.

Example

```
iptables -A INPUT -m mac --mac-source 00:00:00:11:22:33 -j DROP
iptables -A OUTPUT -m owner !--uid-owner 1001 -j DROP
```



Plantilla base

Una buena base

*filter

:INPUT DROP

:FORWARD DROP

:OUTPUT ACCEPT

-A INPUT -i lo -j ACCEPT

-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

COMMIT



NAT

SNAT y DNAT

Situación

- Utilizar una ip externa para toda una red interna.
- Hacer uso de un proxy transparente.

Acciones

- Enmascarar nuestra red.
- Enviar los paquetes al proxy.

Necesitamos modificar paquetes → utilizamos otras tablas.



POSTROUTING

SNAT

- Queremos enmascar nuestra red interna.
- Utilizamos el objetivo *MASQUERADE*.
- Automáticamente se encargará de gestionar las conexiones.

Example

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE  
iptables -A FORWARD -i eth1 -j ACCEPT  
echo 1 > /proc/sys/net/ipv4/ip_forward
```



PREROUTING

DNAT

- Queremos enviar transparentemente un servicio a un proxy, por ejemplo el tráfico web.
- Modificamos los paquetes nada más llegar al interfaz de red.
- Utilizamos el objetivo *DNAT*

Example

```
iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j DNAT --to 192.168.0.1:8080
```

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to 192.168.0.10:80
```



netstat

Visualizando la red

1 Protocolo

t TCP

u UDP

2 Estado

l Solamente en escucha.

a Todos.

3 Sin resolución de nombres: *-n*.

4 Programa asociado: *-p*.

Example

```
netstat -putan
```



netcat

La navaja suiza

Conexiones de red de forma rápida y fácil.

Sintaxis

```
nc OPCIONES DESTINO PUERTO
```

Destino

```
nc -l -p 4000 | tar zcv
```

Origen

```
tar zcv home | nc -q 0 destino 4000
```

- l Quedarse a la escucha.
- p Escuchar en un puerto concreto.
- q Esperar después de un EOF.



netcat (y 2)

Ejemplos

- Replicar una partición:

```
dd if=/dev/hda2 | nc destino puerto  
nc -l -p puerto | dd of=/dev/hda2
```

- Enviar datos entre varios:

```
entrada | nc destino1 puerto1  
nc -l -p puerto1 | nc destino2 puerto2  
nc -l -p puertoN | salida
```

- Interprete de comandos:

```
nc -l -p puerto -e /bin/sh
```

- Interprete de comandos inverso:

```
nc -l -p puerto  
nc -e /bin/sh destino puerto
```



nmap

El escaneador de puertos

- Opciones de escaneador:
 - sP Búsqueda de máquinas mediante PING.
 - sU Servicios UDP.
 - sT Establecer una conexión completa.
 - sS Establece solamente un SYN.
 - sF,-sX,-sN Conexiones FIN, Xmas y NULL.
 - sV Descubrir versiones.
- Puertos:
 - Rango de puertos: *-p N-M,X*
 - Puertos en /etc/services: *-F*
- Identificar sistema operativo: *-O*
- Tiempo entre paquetes: *-T [0-5]*



Resumen

Conocimientos adquiridos:

- Conceptos del manejo de paquetes de Netfilter.
- Manejo básico de `iptables`.
- Uso de las herramientas de redes (`netstat`, `netcat...`).

Otras herramientas interesantes:

- Nessus: <http://www.nessus.org/>
- Snort: <http://www.snort.org/>
- tcpdump: <http://www.tcpdump.org/>
- ethereal: <http://www.ethereal.com/>
- Kismet: <http://www.kismetwireless.net/>
- hping: <http://www.hping.org/>
- ettercap: <http://ettercap.sourceforge.net/>

