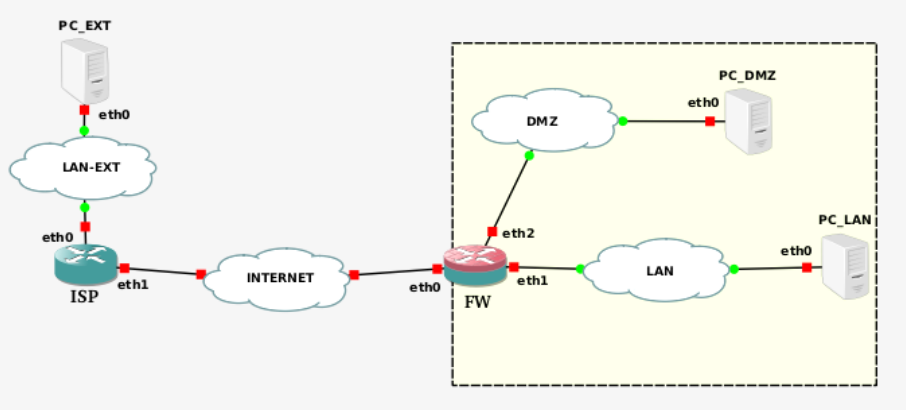
ACTIVIDAD: FIREWALL con `IPTABLES`

El objetivo es comprender el funcionamiento de un firewall en una red con una DMZ, estudiando las reglas de filtrado y traducción entre redes. Se trabajará con el software `Netfilter`, y en concreto, con el comando `iptables`.

Se realizará sobre una máquina virtual `Debian` donde se encuentra instalado el software `GNS3`[1] que es un simulador gráfico de redes. Esta herramienta permite definir una topología de red y poner en marcha todos su nodos (hosts, routers o switches) que, en este caso, están ejecutados como contenedores `Docker`[2].

Entorno de trabajo

1. El escenario representa la red de una entidad compuesta por una red local que se conecta a internet a través de un *triple-homed firewall* con una DMZ. También dispone de una red local en una sucursal remota. Para simplificar la actividad, en cada red hay una única máquina en la que solo está configurado el servicio web (`lighttpd`).



2. Para comenzar:

   a) Descargar contenedor actualizado desde *egela* (`iptermlog.tar.gz`) y cargarlo en la máquina virtual (`docker load < iptermlog.tar.gz`)

   b) Descomprimir el proyecto (`tar xvzf` *`fichero.tar.gz`* `-C` *`path/destino`*). Lanzar `GNS3` (no actualizar) y abrir el proyecto `srdsi-firewall`; poner en marcha todos los dispositivos y abrir las consolas de todos los PC's y del cortafuegos `FW`.

3. La única dirección pública de nuestra LAN se corresponde con la asignada a la interfaz `eth0` del firewall.

4. La política general a aplicar:

[1] https://www.gns3.com
[2] https://www.docker.com

   a) Permitir accesos desde Internet y la red local a los servidores de la DMZ
   b) No permitir acceso a los servidores de la red local excepto desde la propia red.
   c) Permitir acceso a Internet a los servidores de la red local.
   d) No permitir a los servidores de la DMZ establecer conexiones

5. Personalizar la página web (`/var/www/html/index.html`) en los tres servidores web para que incluya el nombre del grupo.

6. Realizar los ejercicios propuestos más adelante. En la máquina `FW` se encuentran disponibles (en `/root`) los ficheros requeridos.

Documentación a entregar

Soluciones documentadas (pantallazos, capturas wireshark...) de los ejercicios propuestos y análisis de los resultados obtenidos. Indicar tiempo dedicado a su realización.

EJERCICIOS

1. Puesta en marcha sin control de accesos.

   Todos los dispositivos tienen realizada su configuración IP [3]. Está activado el encaminamiento IP sobre el dispositivo `FW` (firewall) pero sin reglas de filtrado/traducción definidas.

   a) Cargar la configuración inicial de las reglas de `iptables` en `FW`:
   `FW# sh ./fw-rules.sh`

   b) Comprobar la configuración de las interfaces (`ifconfig`) y el contenido de las tablas de encaminamiento (`route -n`) de los PC's y de `FW`

   c) Verificar que desde cualquier PC[4] se puede acceder al resto de PC's (`ping/traceroute -I`)

   d) Probar acceso a los servidores web desde el PC en la red `LAN` (`wget`/`lynx`)

   e) Examinar acceso a otros puertos TCP/UDP (`nc`)

   f) Analizar tramas enviadas/recibidas por la red (`tcpdump` / `wireshark`)

2. En la máquina `FW` añadir el contenido de `fw-ejerc2.txt` al script `fw-rules.sh` (en la posición indicada):

```
$IPTABLES -P INPUT DROP
$IPTABLES -P OUTPUT DROP
$IPTABLES -P FORWARD DROP

[...]
#sh ./fw-rules.sh
```

   a) ¿En qué cambia el comportamiento del firewall?

   b) Comprobar la respuesta dada en el apartado anterior.

3. Añadir las nuevas reglas del fichero `fw-ejerc3.txt`

[3] Por comodidad en el fichero `/etc/hosts`, se han incorporado las resoluciones de los nombres de los PC's y de las páginas web, para no recurrir a DNS. Atención: desde Internet no podrían resolverse los nombres que se corresponden a @IP's privadas.

[4] En una situación real no es posible acceder desde Internet a una máquina que posea una @IP privada, ya que estas direcciones no son reenviadas por los routers en Internet.

```
$IPTABLES -A INPUT -i $LAN_IFACE -s $LAN_IP -d $LAN_IFACE_IP\
                -p icmp -m icmp --icmp-type echo-request \
                -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A OUTPUT -o $LAN_IFACE -d $LAN_IP -s $LAN_IFACE_IP \
                 -p icmp -m icmp --icmp-type echo-reply \
                 -m state --state RELATED,ESTABLISHED -j ACCEPT
$IPTABLES -A INPUT -i $DMZ_IFACE -s $DMZ_IP -d $DMZ_IFACE_IP\
                -p icmp -m icmp --icmp-type echo-request \
                -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A OUTPUT -o $DMZ_IFACE -d $DMZ_IP -s $DMZ_IFACE_IP\
                 -p icmp -m icmp --icmp-type echo-reply \
                 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

a) Las dos primeras reglas de filtrado permiten aceptar un comando `ping` al propio firewall (por la interfaz `eth1`) proveniente de la red `LAN`. Testar

b) ¿Y las otras dos?

c) Si se quiere permitir aceptar un comando `ping` de la red `LAN` a la `DMZ` ¿Qué cadena es necesario modificar? Efectuar la modificación y testar.

d) ¿Es necesario añadir alguna(s) otra(s) regla(s) respecto al comando `ping`?

4. Añadir las nuevas reglas del fichero `fw-ejerc4.txt`

```
$IPTABLES -A FORWARD -i $LAN_IFACE -s $LAN_IP \
                 -o $DMZ_IFACE -d $DMZ_WEB_IP \
                 -m state --state NEW,ESTABLISHED,RELATED \
                 -p tcp -m multiport --dport 80,443 -j ACCEPT
$IPTABLES -A FORWARD -o $LAN_IFACE -d $LAN_IP \
                 -i $DMZ_IFACE -s $DMZ_WEB_IP \
                 -m state --state ESTABLISHED,RELATED \
                 -p tcp -m multiport --sport 80,443 -j ACCEPT
```

a) ¿Cuál es la finalidad de estas reglas?

b) Comprobar la respuesta al apartado anterior

5. La `DMZ` es una red expuesta siendo conveniente aislar la red `LAN` mediante la traducción de todas las direcciones provenientes de la `LAN` y que van a la `DMZ` (no es estrictamente necesario si `DMZ` también trabaja con direcciones privadas).

a) Comprobar que las direcciones de la `LAN` son visibles en la `DMZ`[5]. Analizar tramas (`tcpdump` / `wireshark`).

b) Realizar la traducción de las direcciones entre la `LAN` y la `DMZ`.

```
$IPTABLES -t nat -A POSTROUTING -o $DMZ_IFACE -p all \
          -s $LAN_IP -d $DMZ_IP -j SNAT --to $DMZ_IFACE_IP
```

c) Constatar que las direcciones de la `LAN` ya no son visibles en la `DMZ`. Analizar tramas (`tcpdump` / `wireshark`).

d) Consultar el contenido de la tabla `NAT`

6. Las máquinas de la `LAN` no deben ser directamente visibles desde Internet

a) Añadir reglas que solo permitan a las máquinas de la `LAN` realizar: `ping` a máquinas en Internet y accesos a servidores web situados en Internet.

b) Realizar la traducción de la dirección para todo tráfico que tenga como dirección fuente la red `LAN` y como destino `INTERNET` (o cualquier dirección).

c) Verificar las reglas definidas comprobando accesos. Examinar las traducciones realizadas (`man conntrack`).

7. Filtrado entre la red `DMZ` e `INTERNET`

a) El servidor web de la `DMZ` debe tener acceso público desde Internet con URL `http://203.0.113.10`. Añadir las reglas necesarias para realizar la traducción de la dirección y el puerto (`fw-ejerc7.txt`).

```
$IPTABLES -t nat -A PREROUTING -i $INTERNET_IFACE \
                -d $INTERNET_IFACE_IP -p tcp --dport 80 \
                -j DNAT --to-destination $DMZ_WEB_IP:80
$IPTABLES -t nat -A PREROUTING -i $INTERNET_IFACE \
                -d $INTERNET_IFACE_IP -p tcp --dport 443 \
                -j DNAT --to-destination $DMZ_WEB_IP:443
```

b) Testar la configuración, accediendo desde el PC de la `LAN-EXT` a través de un navegador con la URL `http://203.0.113.10`

c) ¿Funciona? Solo hemos habilitado las reglas de traducción, es necesario añadir las reglas de filtrado correspondientes. Volver a testar b).

d) Analizar con `tcpdump/wireshark` las tramas obtenidas en las conexiones `HTTP` anteriores (b) y c)).

[5] Dado nuestro entorno de simulación, con resolución DNS local, las direcciones de PC-LAN son siempre visibles, lo que no ocurre en un entorno real.

8. El servidor web de la `DMZ` debe tener acceso público desde Internet, pero nunca directamente a su dirección privada `10.10.10.30`[6].

   a) Comprobar el acceso al servidor web de la `DMZ` desde Internet.

   b) Añadir las reglas necesarias para impedir el acceso desde Internet con direcciones privadas (`fw-ejerc8.txt`) y vuelve a testar a).

```
$IPTABLES -t nat -A PREROUTING -i $INTERNET_IFACE \
                -d $DMZ_IP -j DNAT --to $LO_IFACE_IP
$IPTABLES -t nat -A PREROUTING -i $INTERNET_IFACE \
                -d $LAN_IP -j DNAT --to $LO_IFACE_IP
```

9. Incorporar reglas que registren los intentos de accesos no permitidos incluyendo un comentario descriptivo.

   Para habilitar la recepción de registros (`/var/log/messages`) desde GNS3 ejecuta el siguiente comando en la máquina virtual:

   ```
   $ sudo sysctl -w net.netfilter.nf_log_all_netns=1
   ```

10. [OPCIONAL] Proponer nuevas reglas para bloquear ciertos tipos de paquetes asociados a un posible intento de ataque. [7]

[6] Internet no permite el reenvío de paquetes con IP de destino a direcciones privadas, pero se pueden producir errores (o ataques) que los remitan, es necesario filtrarlos.

[7] Por ejemplo: https://juncotic.com/ddos-mitigando-denegacion-iptables/

Reglas iniciales para `iptables`: `fw-rules.sh`

```
#!/bin/sh
# FW con tres patas (DMZ, LAN, INTERNET)
#
#
# Configuración
# Interfaces
LO_IFACE=lo
LAN_IFACE=eth1
DMZ_IFACE=eth2
INTERNET_IFACE=eth0
#
# @IP interfaces
LO_IFACE_IP=127.0.0.1
FW_IFACE_IP=127.0.1.1
LAN_IFACE_IP=192.168.1.1
DMZ_IFACE_IP=10.10.10.1
INTERNET_IFACE_IP=203.0.113.10
#
# @IP redes
LAN_IP=192.168.1.0/24
DMZ_IP=10.10.10.0/24
ANY_IP=0.0.0.0/0
#
# Servicios
LAN_WEB_IP=192.168.1.10
DMZ_WEB_IP=10.10.10.30
#
# Comando
IPTABLES="/sbin/iptables"
#
```

```
######  Reglas iptables
#
### FILTER
# Vaciar reglas
$IPTABLES -F
$IPTABLES -X
$IPTABLES -Z
#
# INSERTAR AQUÍ fw-ejerc2.txt
#
#
## INSERTAR AQUÍ reglas ejercicio 6a
#
#
### NAT
# Vaciar reglas
$IPTABLES -t nat -F
$IPTABLES -t nat -X
$IPTABLES -t nat -Z
#
# INSERTAR AQUÍ reglas auxiliares ejercicio 7a
#
# Reglas
#
# INSERTAR AQUÍ regla ejercicio 6b
#
# INSERTAR AQUÍ regla ejercicio 5
#
# INSERTAR AQUÍ fw-ejerc7.txt
```

PACKET-FILTERING FIREWALL RULES

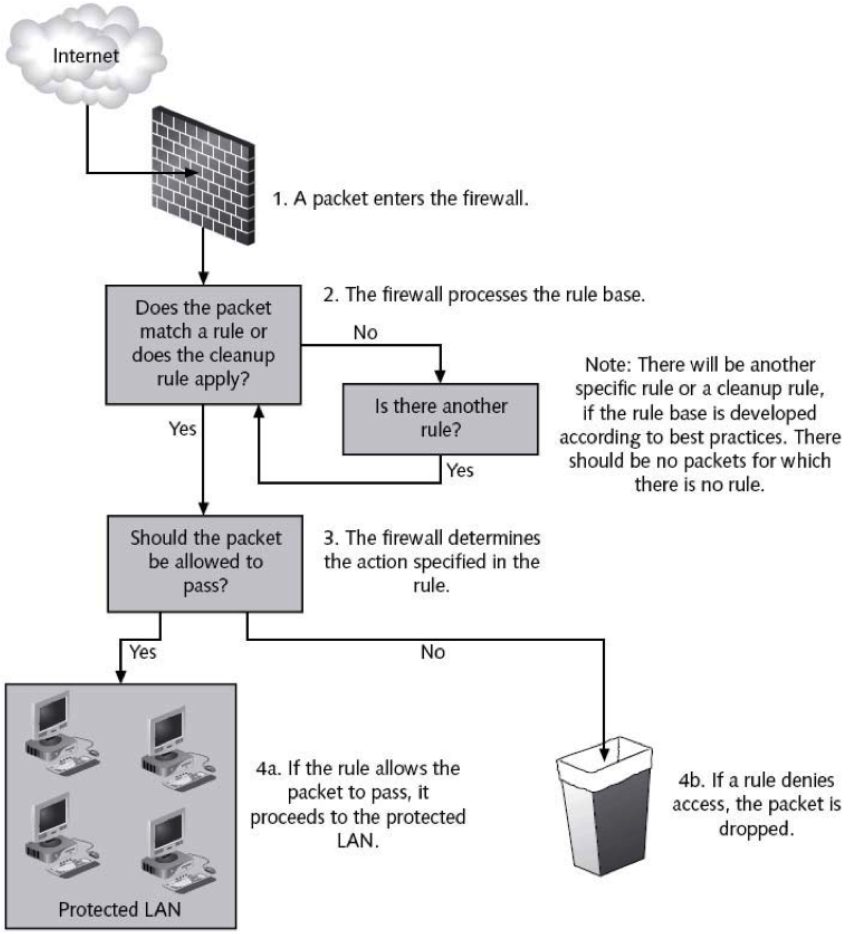


Figure 9-11 Firewalls process rules in order until a match is found

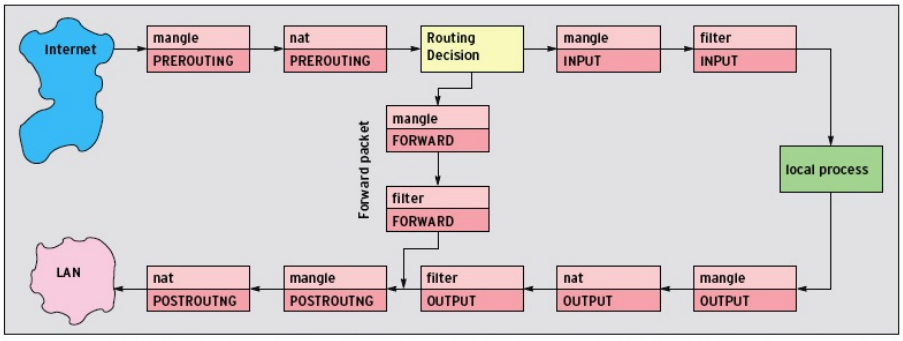


Figure 3: With Iptables, incoming packages first pass through the mangle table in the PREROUTING chain; the NAT table then follows. If the packet is destined for another host, the packet is simply passed to the mangle table in the POSTROUTING chain.

- Tabla **Filter**, cadenas:
  - **INPUT** - Todos los paquetes **dirigidos** a un proceso local de la máquina.
  - **OUTPUT** - Todos los paquetes **originados** desde un proceso local en la máquina.
  - **FORWARD** - Todos los paquetes que no son originados o dirigidos a un proceso local de la máquina, pero pasan a través de ella (**encaminados**). Esta cadena es usada cuando la máquina actúa como un encaminador.
- Tabla **NAT**, cadenas:
  - **POSTROUTING** - Cambiar la dirección de origen de las conexiones. Se hace justo antes de que sea enviado, lo que significa que cualquier otro servicio de la máquina Linux (encaminamiento, filtrado de paquetes) verá el paquete sin cambiar.
  - **PREROUTING** - Cambiar la dirección de destino de las conexiones. Se hace según entra el paquete; esto significa que cualquier otro servicio de la máquina Linux (encaminamiento, filtrado de paquetes) verá el paquete cambiado con su destino «real» (el definitivo).
  - **OUTPUT** - Cambiar la dirección de origen de las conexiones **originadas** desde un proceso local en la máquina.
- Reglas: sintaxis general:

  `iptables [-t tabla] operación [cadena] [filtros] [-j acción]`

Referencias:

- *http://www.ticarte.com/contenido/iptables-**conceptos-generales**-para-configurar-un-cortafuegos*
- *http://es.tldp.org/Manuales-LuCAS/doc-iptables-firewall/doc-iptables-firewall.pdf*
- *https://www.digitalocean.com/community/tutorials/how-to-choose-an-effective-firewall-policy-to-secure-your-servers*
- *https://wiki.archilinux.org/tittle/Simple-stateful-firewall*

ANEXOS

Comandos (más información en `man`)

- `nc(netcat)`
  - para establecer conexiones TCP cliente/servidor (UDP con opción `-u`)
    Servidor: `nc -l -p <puerto>`
    Cliente: `nc <servidor> <puerto>`
  - Escanear un rango de puertos:
    `netcat -zv <servidor> 20-25`
- `tcpdump`, analizador de tramas

  `tcpdump -i <interface>`

  `tcpdump host PC_LAN`
- `wget`/`lynx` son clientes web, el primero descarga los ficheros, el segundo es un navegador en modo texto:

  `wget http://servidor-web --connect-timeout=10`

  `lynx http://servidor-web`

IPTABLES

- Cuando se utiliza un cortafuegos con política por defecto `DROP`, las reglas de `iptables` de la tabla `filter` suelen ir por parejas para permitir las solicitudes y las respuestas (INPUT/OUTPUT para procesos locales y FORWARD/FORWARD para reenviar)
- Bajo una política «denegar por defecto», para que un cortafuegos sea efectivo, las reglas permisivas deben ser lo más restrictivas posibles. Así, es mejor

  `$ iptables -A FORWARD -i eth1 -s 10.0.0.1 -o eth0 -d 8.8.8.8 -p udp --dport 53 -j ACCEPT`

  que una regla como:

  `$ iptables -A FORWARD -p udp --dport 53 -j ACCEPT`
- `man iptables, man iptables-extensions`