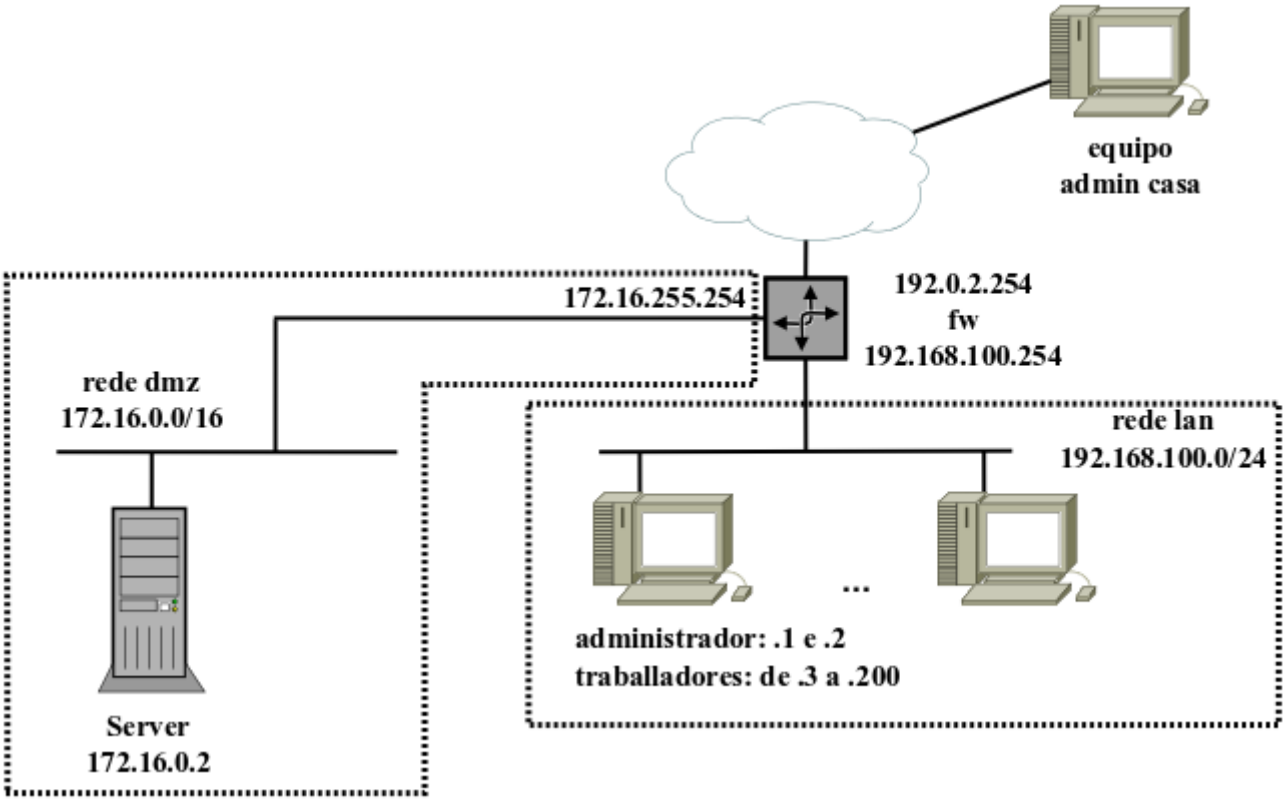
1. Escenario



As condicións a cumprir son:

- Permítese o acceso dende calquera lugar ós servizos http e https de Server.
- Permítese o acceso por ssh ó Server dende os equipos do administrador, tanto dende a LAN coma dende casa.
- Permítese o acceso por ssh ó FW dende o equipo de casa do administrador e dende os equipos admin da Lan.
- Tanto o Server coma o FW usarán para facer consultas DNS os servidores 8.8.8.8 e 8.8.4.4.
- Tanto o Server coma o FW poderán conectarse ós repositorios de ubuntu para poder actualizarse e instalar novos programas.
- Os equipos do administrador da lan e dos traballadores usarán como servidores DNS os servidores DNS 208.67.222.222 e 208.67.220.220.
- Os equipos do administrador da lan e dos traballadores poderán visitar páxinas web (http/https) - os traballadores únicamente de luns a venres das 7:00 ás 15:00 horas.
- As comunicacións dentro da organización faranse sen ningún tipo de NAT.
- O firewall non permitirá ningunha outra comunicación a maiores; é dicir, calquera outro equipo fóra dos rangos establecidos estará completamente bloqueado e calquera outro trafico fóra do indicado estará bloqueado.

Creación automática do Escenario

Descargamos o arquivo script_dmz_focal.zip no equipo anfitrión dos contedores, procedemos a descomprimilo, asegurarnos de que o script teña os permisos axeitados e lánzase escollendo a opción 1:

```
manuel@x99:~$ unzip script_dmz_focal.zip
Archive:  script_dmz_focal.zip
   creating: script_dmz_focal/
  inflating: script_dmz_focal/config.yml
  inflating: script_dmz_focal/config.yml_PLANTILLA
  inflating: script_dmz_focal/config.yml_PLANTILLA_DMZ
  inflating: script_dmz_focal/config.yml_PLANTILLA_FW
  inflating: script_dmz_focal/escenario_dmz.sh
  inflating: script_dmz_focal/profile_3NICs
  inflating: script_dmz_focal/profile_dmz
  inflating: script_dmz_focal/profile_lan
  inflating: script_dmz_focal/profile_wan
manuel@x99:~$ cd script_dmz_focal/
manuel@x99:~/script_dmz_focal$ ls -lhF
total 36K
-rw-r--r-- 1 manuel manuel 1,1K abr  1 12:10 config.yml
-rw-r--r-- 1 manuel manuel 1,1K abr  1 12:08 config.yml_PLANTILLA
-rw-r--r-- 1 manuel manuel 1,1K abr  1 12:08 config.yml_PLANTILLA_DMZ
-rw-r--r-- 1 manuel manuel 1,2K abr  1 12:08 config.yml_PLANTILLA_FW
-rwxr-xr-x 1 manuel manuel 3,8K abr  1 12:07 escenario_dmz.sh*
-rw-r--r-- 1 manuel manuel  394 abr  1 11:41 profile_3NICs
-rw-r--r-- 1 manuel manuel  212 abr  1 11:42 profile_dmz
-rw-r--r-- 1 manuel manuel  212 abr  1 11:42 profile_lan
-rw-r--r-- 1 manuel manuel  212 abr  1 11:42 profile_wan
manuel@x99:~/script_dmz_focal$ ./escenario_dmz.sh
vie 01 abr 2022 12:50:36 CEST
----- Escenario Firewall de red con DMZ -----
Seleccionar operación:
1. Crear escenario
2. Parar equipos
3. Arrancar equipos
4. Borrar escenario
------
```

Unha vez rematada a execución do script, agardar uns segundos e verificar que os contedores están creados e configurados:

```
manuel@x99:~/script_dmz_focal$ lxc ls -c n,4,s,S,P
+-----------+-----------------------+---------+-----------+----------+
|   NAME    |          IPV4         |  STATE  | SNAPSHOTS | PROFILES |
+-----------+-----------------------+---------+-----------+----------+
| admin2    | 192.168.100.170 (eth0) | RUNNING | 0         | lan      |
+-----------+-----------------------+---------+-----------+----------+
| fw        | 192.168.100.254 (eth1) | RUNNING | 0         | 3NICs    |
|           | 192.0.2.254 (eth0)     |         |           |          |
|           | 172.16.255.254 (eth2)  |         |           |          |
+-----------+-----------------------+---------+-----------+----------+
| serverdmz | 172.16.0.2 (eth0)      | RUNNING | 0         | dmz      |
+-----------+-----------------------+---------+-----------+----------+
```

Este script permite parar e arrincar os contedores así como destruir todo o escenario, borrando contedores e os *profiles*. Podemos verificar que hai comunicación entre o fw cos contedores admin2 e serverdmz e entre o fw e Internet:

```
manuel@x99:~/script_dmz_focal$ lxc exec fw -- ping -c 2 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=53 time=17.8 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=53 time=17.7 ms
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 17.719/17.762/17.806/0.140 ms
manuel@x99:~/script_dmz_focal$ lxc exec fw -- ping -c 2 192.168.100.2
PING 192.168.100.2 (192.168.100.2) 56(84) bytes of data.
64 bytes from 192.168.100.2: icmp_seq=1 ttl=64 time=0.096 ms
64 bytes from 192.168.100.2: icmp_seq=2 ttl=64 time=0.083 ms
--- 192.168.100.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
```

```
rtt min/avg/max/mdev = 0.083/0.089/0.096/0.011 ms
manuel@x99:~/script_dmz_focal$ lxc exec fw -- ping -c 2 172.16.0.2
PING 172.16.0.2 (172.16.0.2) 56(84) bytes of data.
64 bytes from 172.16.0.2: icmp_seq=1 ttl=64 time=0.111 ms
64 bytes from 172.16.0.2: icmp_seq=2 ttl=64 time=0.087 ms
--- 172.16.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.087/0.099/0.111/0.012 ms
```

Usuarios e acceso

En cada contedor creado polo script hai un usuario chamado sad con contrasinal magasix. Pódese acceder ós contedores como root con `lxc exec` *`contedor`* `bash` ou como usuario sad por ssh ou ben por `lxc exec` *`contedor`* `-- su -sad`

2. Enrutamento e tráfico de FW

Activar enrutamento

Para facer que un equipo Linux funcione como un router hai que asignar á variable do sistema `net.ipv4.ip_forward` o valor 1. Para asignarlle o valor 1 e facer o cambio persistente, resistindo reinicios da máquina, editaremos o arquivo de configuración `/etc/sysctl.conf` e descomentados a liña correspodente a `net.ipv4.ip_forward`:

```
sad@fw:~$ sudo nano /etc/sysctl.conf
...
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
...
```

Unha vez feito o cambio e gardado o cambio, procedemos a facer efectivo o novo valor:

```
sad@fw:~$ sudo sysctl -p
net.ipv4.ip_forward = 1
sad@fw:~$
```

Un punto a ter claro é que a partir deste momento, o equipo FW é un router; é dicir, pode reenviar paquetes. Trátase dun maneira moi rápida e sinxela de ter un router; non obstante reenviar paquetes non quere dicir facer NAT, xa que para iso hai que configurar netfilter.

Tráfico de FW

Neste punto abordaremos o tráfico propio de FW; é dicir, tráfico xerado/destinado por/hacia aplicacións correndo na máquina. FW está conectado directamente á Intranet, á DMZ e á Internet polo que non é preciso facer ningún tipo de NAT para o seu tráfico, sendo as cadeas INPUT e OUTPUT nas que teremos que centrarnos. Xestionaremos o tráfico de FW usando cadeas de usuario para ver o seu uso.

O punto de partida é unha táboa de filtrado baleira:

```
sad@fw:~$ sudo iptables -L -n --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
```

Creamos as regras para aceptar todos os paquetes con destino/orixe FW que pertencen a unha conexión xa autorizada e establecida:

```
sad@fw:~$ sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
sad@fw:~$ sudo iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

Tráfico Rede de lazo pechado

A regra de loopback está pensada para permitir o tráfico do host dirixido ó propio host; e sen ela, pode haber problemas de comunicacións entre aplicaciones correndo no propio equipo. Para crear estas regras empregamos as opcións `-i lo` (os paquetes teñen que entrar pola interface de loopback) e `-o lo` (os paquetes teñen que saír pola interface de loopback):

```
sad@fw:~$ sudo iptables -A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
sad@fw:~$ sudo iptables -A OUTPUT -o lo -m conntrack --ctstate NEW -j ACCEPT
sad@fw:~$ sudo iptables -L -n --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state NEW
Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state NEW
sad@fw:~$
```

Para ter máis detalles e ver as interfaces podemos engadir a opción -v:

```
sad@fw:~$ sudo iptables -L -n -v --line-number
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1        0     0 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0            ctstate
RELATED,ESTABLISHED
2        0     0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0            state NEW
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1        0     0 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0            ctstate
RELATED,ESTABLISHED
2        0     0 ACCEPT     all  --  *      lo      0.0.0.0/0            0.0.0.0/0            state NEW
sad@fw:~$
```

Tráfico SSH de administración de FW

Entre as boas prácticas á hora de deseñar un firewall está a de restrinxir o acceso para administralo. Entre as primeiras regras do ruleset debe haber:

- Unha regra que garanta o acceso ó firewall dende equipos autorizados (regla anti-bloqueo ou anti-lockdown rule).
- Unhas regras que rexistran e prohiban o acceso dende o resto de equipos (regra de bloqueo ou Lockdown Rule ou Stealth Rule).

Traballaremos con cadeas de usuario que permitirán organizar as regras en base ó tipo de tráfico. En primeiro lugar creamos a cadea de usuario SSH e despois creamos unha regra en INPUT para que o tráfico ssh de administración sexa redirixido á cadea SSH:

```
sad@fw:~$ sudo iptables -N SSH
sad@fw:~$ sudo iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW -j SSH
sad@fw:~$ sudo iptables -L -n --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state NEW
3    SSH        tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:22 ctstate NEW
Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state NEW
Chain SSH (1 references)
```

```
num  target     prot opt source               destination
sad@fw:~$
```

Neste momento todo o tráfico de ssh de administración de FW é enviado á cadea SSH; e é ahí, onde crearemos as regras para permitir ós equipos dos admins e bloquear ó resto:

```
sad@fw:~$ sudo iptables -A SSH -s 192.0.2.1 -j ACCEPT
sad@fw:~$ sudo iptables -A SSH -s 192.168.100.1 -j ACCEPT
sad@fw:~$ sudo iptables -A SSH -s 192.168.100.2 -j ACCEPT
sad@fw:~$ sudo iptables -L -n --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state NEW
3    SSH        tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:22 ctstate NEW
Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state NEW
Chain SSH (1 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  192.0.2.1            0.0.0.0/0
2    ACCEPT     all  --  192.168.100.1        0.0.0.0/0
3    ACCEPT     all  --  192.168.100.2        0.0.0.0/0
sad@fw:~$
```

Para rexistrar os intentos de conexión por ssh dende equipos non autorizados, crearemos unha regra a tal efecto na cadea SSH usando unha acción de rexistro. Un punto moi importante a ter en conta é que acción de rexistro non é unha acción definitiva, xa que o paquete fará que se escriba unha entrada no arquivo de log e seguirá procesándose no ruleset do firewall. É dicir, rexistrar nun arquivo de log a aparición dun paquete non implica tomar ningunha decisión sobre o mesmo (descartar ou aceptar), eso decidirase noutra regra.

En iptables podemos rexistrar os paquetes de dúas formas:

- Usando `-j LOG` que rexistará as incidencias no arquivo `/var/log/kern.log`: `-j LOG e --log-prefix "iptables: SSH FW Bloqueo"` (`--log-prefix` úsase para facilitar a súa interpretación).
- Usando `-j NFLOG` que rexistrará as incidencias no arquivo `/var/log/ulog/syslogemu.log`: `-j NFLOG --nflog-prefix "iptables: SSH FW BLoqueo"`

En contedores e por motivos de seguridade non podemos escribir en `/var/log/kern.log` polo que usaremos o sistema NFLOG. De non usar contedores, podemos escoller calquera das dúas, tendo presente a ubicación por defecto onde se gardarán os rexistros. Nun apartado posterior explícase con máis detalle este punto.

Para poder usar NFLOG instalamos o paquete ulogd2:

```
sad@fw:~$ sudo apt-get update
sad@fw:~$ sudo apt-get install ulogd2
```

Procedemos a engadir a regra de rexistro:

```
sad@fw:~$ sudo iptables -A SSH -j NFLOG --nflog-prefix "iptables: SSH FW Bloqueo"
sad@fw:~$ sudo iptables -L -n --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state NEW
3    SSH        tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:22 ctstate NEW
Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state NEW
Chain SSH (1 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  192.0.2.1            0.0.0.0/0
2    ACCEPT     all  --  192.168.100.1        0.0.0.0/0
```

```
3    ACCEPT     all  --  192.168.100.2        0.0.0.0/0
4    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0           nflog-prefix  "iptables: SSH FW Bloqueo"
sad@fw:~$
```

O procesamento dos intentos de conexión por ssh a FW é o seguinte:

- Cando se intenta conectar por ssh con FW o paquete de inicio de conexión ssh chega a regra#3 da cadea INPUT e é desviado á cadea definida polo usuario SSH.
- Unha vez na cadea SSH compárase a IP orixe e se coincide con algunha das IPs dos admins (regra#1, regra#2 e regra#3) a conexión é aceptada.
- No caso de non proceder dun dos equipos dos admins, o paquete chega a regra#4 o que provoca que se rexistre a incidencia no arquivo de logs e despois prosigue a súa marcha chegando ó final da cadea.
- Cando o paquete chega ó final da cadea SSH aplícaselle a política por defecto que é RETURN, voltando a cadea principal INPUT onde seguirá procesándose.

Hai que indicar que as cadeas definidas polos usuarios teñen unha política implícita de RETURN que non se pode cambiar. Os paquetes que chegan ó final dunha cadea de usuario retornan á cadea de orixe de onde proceden e continúase o seu procesamento. No noso caso, os paquetes non procedentes de equipos autorizados queremos que sexan eliminados e podemos facelo de varias formas:

- Unha posible maneira de eliminalos é crear unha regra ó final da cadea SSH a tal efecto: `$ sudo iptables -A SSH -j DROP`
- Outra maneira de eliminalos é deixar que volten á cadea INPUT e matalos ben cunha regra de descarte ou definindo a política a DROP. Se optamos por esta solución hai que ter moi en conta que o paquete segue procesándose na cadea INPUT e se cumpre con algunha regra que o acepte, o paquete será aceptado e non chegará a descartarse.

Nos optaremos por descartalos na cadea SSH:

```
sad@fw:~$ sudo iptables -A SSH -j DROP
sad@fw:~$ sudo iptables -L -n --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state NEW
3    SSH        tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:22 ctstate NEW
Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state NEW
Chain SSH (1 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  192.0.2.1            0.0.0.0/0
2    ACCEPT     all  --  192.168.100.1        0.0.0.0/0
3    ACCEPT     all  --  192.168.100.2        0.0.0.0/0
4    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0           nflog-prefix  "iptables: SSH FW Bloqueo"
5    DROP       all  --  0.0.0.0/0            0.0.0.0/0
sad@fw:~$
```

Resolución DNS de FW

Repetimos o proceso seguido no punto anterior:

- Creamos unha cadea de usuario para o tráfico DNS de FW
- Creamos unha regra na cadea OUTPUT para enviar os paquetes dns á cadea de usuario DNS.
- Aceptamos as consultas enviadas ós servidores 8.8.8.8 e .8.8.4.4
- Rexistramos e denegamos o resto.

```
sad@fw:~$ sudo iptables -N DNS
sad@fw:~$ sudo iptables -A OUTPUT -p udp --dport 53 -m conntrack --ctstate NEW -j DNS
sad@fw:~$ sudo iptables -A DNS -d 8.8.8.8 -j ACCEPT
sad@fw:~$ sudo iptables -A DNS -d 8.8.4.4 -j ACCEPT
```

```
sad@fw:~$ sudo iptables -A DNS -j NFLOG --nflog-prefix "iptables: DNS FW Bloqueo"
sad@fw:~$ sudo iptables -A DNS -j DROP
sad@fw:~$ sudo iptables -L -n --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state NEW
3    SSH        tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:22 ctstate NEW
Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state NEW
3    DNS        udp  --  0.0.0.0/0            0.0.0.0/0            udp dpt:53 ctstate NEW
Chain DNS (1 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            8.8.8.8
2    ACCEPT     all  --  0.0.0.0/0            8.8.4.4
3    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0            nflog-prefix  "iptables: DNS FW Bloqueo"
4    DROP       all  --  0.0.0.0/0            0.0.0.0/0
Chain SSH (1 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  192.0.2.1            0.0.0.0/0
2    ACCEPT     all  --  192.168.100.1        0.0.0.0/0
3    ACCEPT     all  --  192.168.100.2        0.0.0.0/0
4    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0            nflog-prefix  "iptables: SSH FW Bloqueo"
5    DROP       all  --  0.0.0.0/0            0.0.0.0/0
sad@fw:~$
```

Acceso repositorios Ubuntu

Revisando o arquivo `/etc/apt/sources.list` podemos comprobar cales son os repositorios de paquetes que hai que autorizar (o arquivo que aparece a continuación está editado para desactivar os repositorios tipo deb-src e habilitar o repositorio partner):

```
sad@fw:~$ cat /etc/apt/sources.list
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://archive.ubuntu.com/ubuntu/ focal main restricted
# deb-src http://archive.ubuntu.com/ubuntu/ focal main restricted

## Major bug fix updates produced after the final release of the
## distribution.
deb http://archive.ubuntu.com/ubuntu/ focal-updates main restricted
# deb-src http://archive.ubuntu.com/ubuntu/ focal-updates main restricted

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://archive.ubuntu.com/ubuntu/ focal universe
# deb-src http://archive.ubuntu.com/ubuntu/ focal universe
deb http://archive.ubuntu.com/ubuntu/ focal-updates universe
# deb-src http://archive.ubuntu.com/ubuntu/ focal-updates universe

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu
## security team.
deb http://archive.ubuntu.com/ubuntu/ focal multiverse
# deb-src http://archive.ubuntu.com/ubuntu/ focal multiverse
deb http://archive.ubuntu.com/ubuntu/ focal-updates multiverse
# deb-src http://archive.ubuntu.com/ubuntu/ focal-updates multiverse

## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
deb http://archive.ubuntu.com/ubuntu/ focal-backports main restricted universe multiverse
# deb-src http://archive.ubuntu.com/ubuntu/ focal-backports main restricted universe multiverse
```

```
## Uncomment the following two lines to add software from Canonical's
## 'partner' repository.
## This software is not part of Ubuntu, but is offered by Canonical and the
## respective vendors as a service to Ubuntu users.
# deb http://archive.canonical.com/ubuntu focal partner
# deb-src http://archive.canonical.com/ubuntu focal partner

deb http://security.ubuntu.com/ubuntu/ focal-security main restricted
# deb-src http://security.ubuntu.com/ubuntu/ focal-security main restricted
deb http://security.ubuntu.com/ubuntu/ focal-security universe
# deb-src http://security.ubuntu.com/ubuntu/ focal-security universe
deb http://security.ubuntu.com/ubuntu/ focal-security multiverse
# deb-src http://security.ubuntu.com/ubuntu/ focal-security multiverse
sad@fw:~$
```

Polo tanto haberá que autorizar o acceso por http ós equipos archive.ubuntu.com e security.ubuntu.com:

```
sad@fw:~$ sudo iptables -N REPOS
sad@fw:~$ sudo iptables -A OUTPUT -p tcp --dport 80 -m conntrack --ctstate NEW -j REPOS
sad@fw:~$ sudo iptables -A REPOS -d archive.ubuntu.com -j ACCEPT
sad@fw:~$ sudo iptables -A REPOS -d security.ubuntu.com -j ACCEPT
sad@fw:~$ sudo iptables -A REPOS -j NFLOG --nflog-prefix "iptables: REPOS FW Bloqueo"
sad@fw:~$ sudo iptables -A REPOS -j DROP
sad@fw:~$ sudo iptables -L -n --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state NEW
3    SSH        tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:22 ctstate NEW
Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state NEW
3    DNS        udp  --  0.0.0.0/0            0.0.0.0/0            udp dpt:53 ctstate NEW
4    REPOS      tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:80 ctstate NEW
Chain DNS (1 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            8.8.8.8
2    ACCEPT     all  --  0.0.0.0/0            8.8.4.4
3    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0            nflog-prefix  "iptables: DNS FW Bloqueo"
4    DROP       all  --  0.0.0.0/0            0.0.0.0/0
Chain REPOS (1 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            91.189.88.152
2    ACCEPT     all  --  0.0.0.0/0            91.189.88.161
3    ACCEPT     all  --  0.0.0.0/0            91.189.91.23
4    ACCEPT     all  --  0.0.0.0/0            91.189.88.149
5    ACCEPT     all  --  0.0.0.0/0            91.189.88.162
6    ACCEPT     all  --  0.0.0.0/0            91.189.88.152
7    ACCEPT     all  --  0.0.0.0/0            91.189.88.149
8    ACCEPT     all  --  0.0.0.0/0            91.189.88.162
9    ACCEPT     all  --  0.0.0.0/0            91.189.91.26
10   ACCEPT     all  --  0.0.0.0/0            91.189.88.161
11   ACCEPT     all  --  0.0.0.0/0            91.189.91.23
12   ACCEPT     all  --  0.0.0.0/0            91.189.92.150
13   ACCEPT     all  --  0.0.0.0/0            91.189.92.191
14   ACCEPT     all  --  0.0.0.0/0            91.189.91.15
15   NFLOG      all  --  0.0.0.0/0            0.0.0.0/0            nflog-prefix  "iptables: REPOS FW
Bloqueo"
16   DROP       all  --  0.0.0.0/0            0.0.0.0/0
Chain SSH (1 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  192.0.2.1            0.0.0.0/0
2    ACCEPT     all  --  192.168.100.1        0.0.0.0/0
3    ACCEPT     all  --  192.168.100.2        0.0.0.0/0
4    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0            nflog-prefix  "iptables: SSH FW Bloqueo"
5    DROP       all  --  0.0.0.0/0            0.0.0.0/0
sad@fw:~$
```

Denegar por defecto o tráfico orixe/destino de FW: Clean Up rules

Para rematar definimos a política de denegar por defecto para as cadeas INPUT e OUPUT:

```
sad@fw:~$ sudo iptables -P INPUT DROP
sad@fw:~$ sudo iptables -P OUTPUT DROP
sad@fw:~$ sudo iptables -L -n --line-number
Chain INPUT (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state NEW
3    SSH        tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:22 ctstate NEW
Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
Chain OUTPUT (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state NEW
3    DNS        udp  --  0.0.0.0/0            0.0.0.0/0            udp dpt:53 ctstate NEW
4    REPOS      tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:80 ctstate NEW
Chain DNS (1 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            8.8.8.8
2    ACCEPT     all  --  0.0.0.0/0            8.8.4.4
3    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0            nflog-prefix  "iptables: DNS FW Bloqueo"
4    DROP       all  --  0.0.0.0/0            0.0.0.0/0
Chain REPOS (1 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            91.189.88.152
2    ACCEPT     all  --  0.0.0.0/0            91.189.88.161
3    ACCEPT     all  --  0.0.0.0/0            91.189.91.23
4    ACCEPT     all  --  0.0.0.0/0            91.189.88.149
5    ACCEPT     all  --  0.0.0.0/0            91.189.88.162
6    ACCEPT     all  --  0.0.0.0/0            91.189.88.152
7    ACCEPT     all  --  0.0.0.0/0            91.189.88.149
8    ACCEPT     all  --  0.0.0.0/0            91.189.88.162
9    ACCEPT     all  --  0.0.0.0/0            91.189.91.26
10   ACCEPT     all  --  0.0.0.0/0            91.189.88.161
11   ACCEPT     all  --  0.0.0.0/0            91.189.91.23
12   ACCEPT     all  --  0.0.0.0/0            91.189.92.150
13   ACCEPT     all  --  0.0.0.0/0            91.189.92.191
14   ACCEPT     all  --  0.0.0.0/0            91.189.91.15
15   NFLOG      all  --  0.0.0.0/0            0.0.0.0/0            nflog-prefix  "iptables: REPOS FW
Bloqueo"
16   DROP       all  --  0.0.0.0/0            0.0.0.0/0
Chain SSH (1 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  192.0.2.1            0.0.0.0/0
2    ACCEPT     all  --  192.168.100.1        0.0.0.0/0
3    ACCEPT     all  --  192.168.100.2        0.0.0.0/0
4    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0            nflog-prefix  "iptables: SSH FW Bloqueo"
5    DROP       all  --  0.0.0.0/0            0.0.0.0/0
sad@fw:~$
```

Tamén pode facerse creando unhas regras finais de descarte en INPUT e OUTPUT.

Log Denied Rule e No Logging Rules

Neste intre nas cadeas de usuario SSH, DNS e REPOS os paquetes non autorizados son rexistrados para despois ser inmediatamente descartados. En cambio, todos aqueles paquetes que non sexan procesados nesas cadeas serán descartados polas reglas CleanUP pero non rexistrados. É unha boa práctica rexistrar eses paquetes, pero podemos atoparmos con que rexistramos moitas incidencias debido a tráfico necesario nunha rede local pero moi ruidoso (por exemplo o tráfico broadcast). A solución a este problema é a seguinte:

- Definir unhas regras de tipo No Logging Rule onde descartamos (sen rexistralo) tráfico ruidoso pero necesario (broadcast e multicast).
- Definir unhas regras de tipo Log Denied Rule onde rexistramos o tráfico non autorizado antes de descartalo nas CleanUp Rules.

A modo de exemplo imos supoñer que a FW non lle interesa ningún tráfico broadcast ni multicast:

```
sad@fw:~$ sudo iptables -A INPUT -d 192.0.2.255 -j DROP
sad@fw:~$ sudo iptables -A INPUT -d 172.16.255.255 -j DROP
sad@fw:~$ sudo iptables -A INPUT -d 192.168.100.255 -j DROP
sad@fw:~$ sudo iptables -A INPUT -d 255.255.255.255 -j DROP
sad@fw:~$ sudo iptables -A INPUT -d 224.0.0.0/4 -j DROP
sad@fw:~$ sudo iptables -A INPUT -j NFLOG --nflog-prefix "iptables: CleanUP Rule "
sad@fw:~$ sudo iptables -A OUTPUT -j NFLOG --nflog-prefix "iptables: CleanUP Rule "
sad@fw:~$ sudo iptables -L -n --line-numbers
Chain INPUT (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state NEW
3    SSH        tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:22 ctstate NEW
4    DROP       all  --  0.0.0.0/0            192.0.2.255
5    DROP       all  --  0.0.0.0/0            172.16.255.255
6    DROP       all  --  0.0.0.0/0            192.168.100.255
7    DROP       all  --  0.0.0.0/0            255.255.255.255
8    DROP       all  --  0.0.0.0/0            224.0.0.0/4
9    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0           nflog-prefix  "iptables: CleanUP Rule "
Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
Chain OUTPUT (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state NEW
3    DNS        udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:53 ctstate NEW
4    REPOS      tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:80 ctstate NEW
5    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0           nflog-prefix  "iptables: CleanUP Rule "
Chain DNS (1 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            8.8.8.8
2    ACCEPT     all  --  0.0.0.0/0            8.8.4.4
3    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0           nflog-prefix  "iptables: DNS FW Bloqueo"
4    DROP       all  --  0.0.0.0/0            0.0.0.0/0
Chain REPOS (1 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            91.189.88.152
2    ACCEPT     all  --  0.0.0.0/0            91.189.88.161
3    ACCEPT     all  --  0.0.0.0/0            91.189.91.23
4    ACCEPT     all  --  0.0.0.0/0            91.189.88.149
5    ACCEPT     all  --  0.0.0.0/0            91.189.88.162
6    ACCEPT     all  --  0.0.0.0/0            91.189.88.152
7    ACCEPT     all  --  0.0.0.0/0            91.189.88.149
8    ACCEPT     all  --  0.0.0.0/0            91.189.88.162
9    ACCEPT     all  --  0.0.0.0/0            91.189.91.26
10   ACCEPT     all  --  0.0.0.0/0            91.189.88.161
11   ACCEPT     all  --  0.0.0.0/0            91.189.91.23
12   ACCEPT     all  --  0.0.0.0/0            91.189.92.150
13   ACCEPT     all  --  0.0.0.0/0            91.189.92.191
14   ACCEPT     all  --  0.0.0.0/0            91.189.91.15
15   NFLOG      all  --  0.0.0.0/0            0.0.0.0/0           nflog-prefix  "iptables: REPOS FW
Bloqueo"
16   DROP       all  --  0.0.0.0/0            0.0.0.0/0
Chain SSH (1 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  192.0.2.1            0.0.0.0/0
2    ACCEPT     all  --  192.168.100.1        0.0.0.0/0
3    ACCEPT     all  --  192.168.100.2        0.0.0.0/0
4    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0           nflog-prefix  "iptables: SSH FW Bloqueo"
5    DROP       all  --  0.0.0.0/0            0.0.0.0/0
sad@fw:~$
```

3. Tráfico da Intranet

Tráfico orixinado nos equipos da Intranet

Antes de comezar a facer regras imos analizar uns puntos relativos ó tráfico iniciado dende os equipos da Intranet:

- Os equipos da Intranet teñen IPs privadas e saen a Internet para facer resolucións dns e visitar sitios web; e polo tanto, haberá que facer regras NAT e regras de filtrado:
  - Regras SNAT (na cadea POSTROUTING) para cambiar a IP privada orixe dos paquetes pola IP pública de FW.
  - Regras de filtrado para autorizar os paquetes. Como SNAT faise na cadea POSTROUTING, na cadea FORWARD haberá que autorizar os paquetes orixinais (que aínda teñen como IP orixe a IP privada).
- O server é administrable por ssh dende os equipos dos administradores da Intranet: a comunicación entre os equipos dos admins da Intranet e o server ocorre pasando polo FW pero sen saír a Internet. Esto tradúcese en que non é necesario facer SNAT para ter IPs públicas nos paquetes con orixe a lan e destino o server da DMZ; polo tanto, teremos que asegurarnos que estes paquetes non sofren NAT.

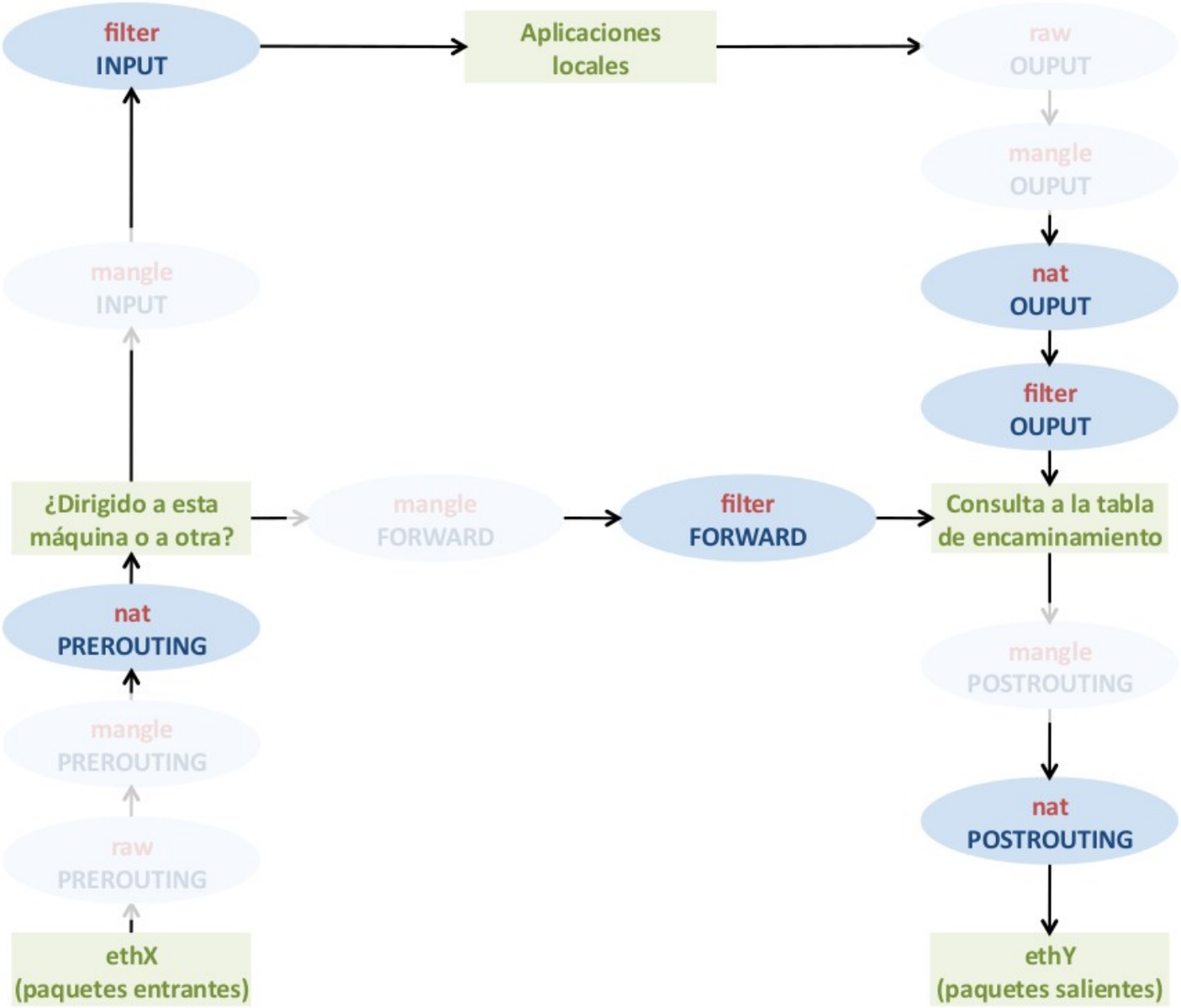


Fig. Táboas e cadeas predeterminadas máis usadas en netfilter. GSyC. CC BY-SA 2.1 ES

Comezamos creando as regras para aceptar todos os paquetes a enrutar por FW Linux que pertencen a unha conexión xa autorizada e establecida e a establecer a política da cadea FORWARD a DROP:

```
sad@fw:~$ sudo iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
sad@fw:~$ sudo iptables -P FORWARD DROP
sad@fw:~$ sudo iptables -L -n --line-numbers
```

```
Chain INPUT (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state NEW
3    SSH        tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:22 ctstate NEW
4    DROP       all  --  0.0.0.0/0            192.0.2.255
5    DROP       all  --  0.0.0.0/0            172.16.255.255
6    DROP       all  --  0.0.0.0/0            192.168.100.255
7    DROP       all  --  0.0.0.0/0            255.255.255.255
8    DROP       all  --  0.0.0.0/0            224.0.0.0/4
9    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0            nflog-prefix  "iptables: CleanUP Rule "
Chain FORWARD (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
Chain OUTPUT (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state NEW
3    DNS        udp  --  0.0.0.0/0            0.0.0.0/0            udp dpt:53 ctstate NEW
4    REPOS      tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:80 ctstate NEW
5    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0            nflog-prefix  "iptables: CleanUP Rule "
...
```

Resolución DNS

Para continuar practicando o uso das cadeas definidas polos usuarios, crearemos unha cadea chamada DNS_INTRA onde enviaremos as consultas dns. Dentro de DNS_INTRA permitiremos os paquetes en base á súa orixe e usaremos `iprange` para definir o rango de IPs autorizadas:

- `-m iprange:` permite especificar un rango de direccións IP
  - `--src-range from[-to]` --> IPs orixe.
  - `--dst-range from[-to]` --> IPs destino.

```
sad@fw:~$ sudo iptables -N DNS_INTRA
sad@fw:~$ sudo iptables -A FORWARD -p udp --dport 53 -d 208.67.222.222 -m conntrack --ctstate NEW -j
DNS_INTRA
sad@fw:~$ sudo iptables -A FORWARD -p udp --dport 53 -d 208.67.220.220 -m conntrack --ctstate NEW -j
DNS_INTRA
sad@fw:~$ sudo iptables -A DNS_INTRA -m iprange --src-range 192.168.100.1-192.168.100.200 -j ACCEPT
sad@fw:~$ sudo iptables -A DNS_INTRA -j NFLOG --nflog-prefix "iptables: DNS INTRA Bloqueo "
sad@fw:~$ sudo iptables -A DNS_INTRA -j DROP
sad@fw:~$ sudo iptables -L -n --line-numbers
Chain INPUT (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state NEW
3    SSH        tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:22 ctstate NEW
4    DROP       all  --  0.0.0.0/0            192.0.2.255
5    DROP       all  --  0.0.0.0/0            172.16.255.255
6    DROP       all  --  0.0.0.0/0            192.168.100.255
7    DROP       all  --  0.0.0.0/0            255.255.255.255
8    DROP       all  --  0.0.0.0/0            224.0.0.0/4
9    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0            nflog-prefix  "iptables: CleanUP Rule "
Chain FORWARD (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    DNS_INTRA  udp  --  0.0.0.0/0            208.67.222.222       udp dpt:53 ctstate NEW
3    DNS_INTRA  udp  --  0.0.0.0/0            208.67.220.220       udp dpt:53 ctstate NEW
...
Chain DNS_INTRA (2 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            source IP range 192.168.100.1-
192.168.100.200
2    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0            nflog-prefix  "iptables: DNS INTRA
Bloqueo "
3    DROP       all  --  0.0.0.0/0            0.0.0.0/0
...
```

Tráfico Web

A continuación controlaremos o tráfico web iniciado dende os clientes da Intranet. Pero para traballar novas condicións restrinxiremos tanto a interface de entrada dos paquetes coma o horario:

- A opción -i permite indicar que os paquetes a enviar á cadea WEB_INTRA teñen que entrar no FW pola interface conectada a rede lan (no meu caso esa interface chámase eth1); é dicir, proceder de equipos da Intranet.
- `-m time`: permite controlar se o paquete chega dentro dun intervalo temporal
  - `--timestart hh:mm[:ss]` e `--timestop hh:mm[:ss]` --> para definir controis en base á hora
  - `--weekdays day[,day...]` --> para definir controis en base ó día da semana.
  - `--kerneltz` --> para usar a hora do sistema como referencia.

```
sad@fw:~$ sudo iptables -N WEB_INTRA
sad@fw:~$ sudo iptables -A FORWARD -i eth1 -p tcp -m multiport --dports 80,443 -m conntrack --ctstate NEW -
j WEB_INTRA
sad@fw:~$ sudo iptables -A WEB_INTRA -s 192.168.100.1 -j ACCEPT
sad@fw:~$ sudo iptables -A WEB_INTRA -s 192.168.100.2 -j ACCEPT
sad@fw:~$ sudo iptables -A WEB_INTRA -m iprange --src-range 192.168.100.3-192.168.100.200 -m time --
timestart 07:00:00 --timestop 15:00:00 --weekdays Mon,Tue,Wed,Thu,Fri --kerneltz -j ACCEPT
sad@fw:~$ sudo iptables -A WEB_INTRA -j NFLOG --nflog-prefix "iptables: WEB INTRA Bloqueo "
sad@fw:~$ sudo iptables -A WEB_INTRA -j DROP
sad@fw:~$ sudo iptables -L -n --line-numbers
Chain INPUT (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state NEW
3    SSH        tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:22 ctstate NEW
4    DROP       all  --  0.0.0.0/0            192.0.2.255
5    DROP       all  --  0.0.0.0/0            172.16.255.255
6    DROP       all  --  0.0.0.0/0            192.168.100.255
7    DROP       all  --  0.0.0.0/0            255.255.255.255
8    DROP       all  --  0.0.0.0/0            224.0.0.0/4
9    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0            nflog-prefix  "iptables: CleanUP Rule "
Chain FORWARD (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    DNS_INTRA  udp  --  0.0.0.0/0            208.67.222.222       udp dpt:53 ctstate NEW
3    DNS_INTRA  udp  --  0.0.0.0/0            208.67.220.220       udp dpt:53 ctstate NEW
4    WEB_INTRA  tcp  --  0.0.0.0/0            0.0.0.0/0            multiport dports 80,443 ctstate NEW
...
Chain WEB_INTRA (1 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  192.168.100.1        0.0.0.0/0
2    ACCEPT     all  --  192.168.100.2        0.0.0.0/0
3    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            source IP range 192.168.100.3-
192.168.100.200 TIME from 07:00:00 to 15:00:00 on Mon,Tue,Wed,Thu,Fri
4    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0            nflog-prefix  "iptables: WEB INTRA
Bloqueo "
5    DROP       all  --  0.0.0.0/0            0.0.0.0/0
```

No listado das regras non aparece a interface na regra#4 da cadea FORWARD. Se facemos o listado coa opción -v si aparece:

```
sad@fw:~$ sudo iptables -L FORWARD -v -n --line-numbers
Chain FORWARD (policy DROP 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1        0     0 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           ctstate
RELATED,ESTABLISHED
2        0     0 DNS_INTRA  udp  --  *      *       0.0.0.0/0            208.67.222.222      udp dpt:53
ctstate NEW
3        0     0 DNS_INTRA  udp  --  *      *       0.0.0.0/0            208.67.220.220      udp dpt:53
ctstate NEW
4        0     0 WEB_INTRA  tcp  --  eth1   *       0.0.0.0/0            0.0.0.0/0           multiport
dports 80,443 ctstate NEW
```

Hai que sinalar que especificar interfaces nunha regra usando -i ou -o ten certas limitacións:

- -i pode usarse únicamente nas cadeas INPUT, FORWARD e PREROUTING.

- -o pode usarse únicamente nas cadeas FORWARD, OUTPUT e POSTROUTING

Tráfico ssh de administración para o server DMZ

Repetimos os pasos seguidos anteriormente.

```
sad@fw:~$ sudo iptables -N SSH_SERVER
sad@fw:~$ sudo iptables -A FORWARD -p tcp --dport 22 -d 172.16.0.2 -m conntrack --ctstate NEW -j SSH_SERVER
sad@fw:~$ sudo iptables -A SSH_SERVER -i eth1 -o eth2 -s 192.168.100.1 -j ACCEPT
sad@fw:~$ sudo iptables -A SSH_SERVER -i eth1 -o eth2 -s 192.168.100.2 -j ACCEPT
sad@fw:~$ sudo iptables -A SSH_SERVER -j NFLOG --nflog-prefix "iptables: SSH SERVER Bloqueo "
sad@fw:~$ sudo iptables -A SSH_SERVER -j DROP
sad@fw:~$ sudo iptables -L FORWARD -v -n --line-numbers
Chain FORWARD (policy DROP 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1        0     0 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0            ctstate
RELATED,ESTABLISHED
2        0     0 DNS_INTRA  udp  --  *      *       0.0.0.0/0            208.67.222.222       udp dpt:53
ctstate NEW
3        0     0 DNS_INTRA  udp  --  *      *       0.0.0.0/0            208.67.220.220       udp dpt:53
ctstate NEW
4        0     0 WEB_INTRA  tcp  --  eth1   *       0.0.0.0/0            0.0.0.0/0            multiport
dports 80,443 ctstate NEW
5        0     0 SSH_SERVER  tcp  --  *      *       0.0.0.0/0            172.16.0.2           tcp dpt:22
ctstate NEW
sad@fw:~$ sudo iptables -L SSH_SERVER -v -n --line-numbers
Chain SSH_SERVER (1 references)
num   pkts bytes target     prot opt in     out     source               destination
1        0     0 ACCEPT     all  --  eth1   eth2    192.168.100.1        0.0.0.0/0
2        0     0 ACCEPT     all  --  eth1   eth2    192.168.100.2        0.0.0.0/0
3        0     0 NFLOG      all  --  *      *       0.0.0.0/0            0.0.0.0/0            nflog-prefix
"iptables: SSH SERVER Bloqueo "
4        0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

A modo de exemplo engadimos a restricción de que os paquetes teñen que entrar dende a interface lan (eth1 no meu equipo) e saír pola interface da rede DMZ (eth2 no meu equipo).

Regras NAT para tráfico iniciado na Intranet

É moi importante lembrar que unha cousa é transformar os paquetes (NAT) e outra é autorizalos/denegalos (filtrado). Unha regra NAT sen a súa regra de filtrado correspondente (e viceversa) non é capaz de facer que os paquetes saian a Internet. Como xa creamos as regras de filtrado que controlan o tráfico con orixe os equipos da Intranet, únicamente falta configurar:

- Unhas regras No NAT para que as comunicacións lan <--> dmz.
- Unhas regras SNAT na cadea POSTROUTING para cambiar a IP orixe privada pola IP pública de FW. Por comodidade, especificamos na regra SNAT coma orixe toda a Intranet sen preocuparnos de incluír máis equipos dos indicados no enunciado; xa que o tráfico iniciado por eses equipos será bloqueado nas regras de filtrado, e xa non chegará a cadea POSTROUTING.

Regras NO NAT

As comunicacións iniciadas na LAN (rede 192.168.100.0/24) hacia a DMZ (rede 172.16.0.0/16) non sufrirán cambios. Crear as seguintes regras garantirán que as comunicación lan --> dmz nunca sufrirán NAT con independencia das regras que creemos máis tarde:

```
sad@fw:~$ sudo iptables -t nat -A PREROUTING -s 192.168.100.0/24 -d 172.16.0.0/16 -j ACCEPT
sad@fw:~$ sudo iptables -t nat -A POSTROUTING -s 192.168.100.0/24 -d 172.16.0.0/16 -j ACCEPT
sad@fw:~$ sudo iptables -t nat -L -n --line-numbers
Chain PREROUTING (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  192.168.100.0/24     172.16.0.0/16
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
```

```
Chain POSTROUTING (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  192.168.100.0/24     172.16.0.0/16
sad@fw:~$
```

Regra SNAT

```
sad@fw:~$ sudo iptables -t nat -A POSTROUTING -s 192.168.100.0/24 -j SNAT --to-source 192.0.2.254
sad@fw:~$ sudo iptables -t nat -L -n --line-numbers
Chain PREROUTING (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  192.168.100.0/24     172.16.0.0/16
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
Chain POSTROUTING (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  192.168.100.0/24     172.16.0.0/16
2    SNAT       all  --  192.168.100.0/24     0.0.0.0/0            to:192.0.2.254
```

Fixarse que na cadea POSTROUTING a primeira regra é unha No NAT e se os equipos da LAN acceden ó server DMZ os seus paquetes non se modifican ó aceptarse na regra#1; en cambio, se o destino é outro equipo (Internet) a regra#1 non se lle aplica e ó chegar á regra#2 cambiase a IP orixe pola IP externa de FW (192.0.2.254).

4. Tráfico do Server

Tráfico do server da DMZ

Antes de comezar a facer regras imos analizar uns puntos relativos ó tráfico do server da DMZ e en xeral de calquera equipo ubicado na DMZ:

- Os equipos da DMZ teñen IPs privadas (172.16.0.0/16) e saen a Internet para facer resolucións dns e acceder ós repositorios; e polo tanto, haberá que facer regras NAT e regras de filtrado:
    - Regras SNAT (na cadea POSTROUTING) para cambiar a IP privada orixe dos paquetes pola IP pública de FW.
    - Regras de filtrado para autorizar os paquetes. Como SNAT faise na cadea POSTROUTING, na cadea FORWARD haberá que autorizar os paquetes orixinais (que aínda teñen como IP orixe a IP privada).
- Hai tráfico procedente de Internet:
    - O server é administrable por ssh dende o equipo da casa do administrador.
    - O servizo http/https é de acceso público.

  Este tipo de tráfico obriga a facer regras NAT e de filtrado:
    - Regras DNAT na cadea PREROUTING para mapear un socket 'público' (`<IP pública de FW, tcp, porto>`) cun socket do server dmz. Na seguinte táboa aparece un posible mapeo, pero no caso do ssh teremos que pararnos a analizar o que queremos e tomar unha decisión sobre o número de porto a mapear.
    - Regras de filtrado en FORWARD que permitan que os paquetes transformados nas regras DNAT cheguen ó server. Fixarse que en primeiro lugar os paquetes sofren DNAT e polo tanto nas regras de filtrado hai que autorizar os paquetes con IP destino a IP privada do server (172.16.0.2).

IP pública	Porto público	IP privada	Porto privado
192.0.2.254	tcp/80	172.16.0.2	tcp/80
192.0.2.254	tcp/443	172.16.0.2	tcp/443
192.0.2.254	tcp/22	172.16.0.2	tcp/22*

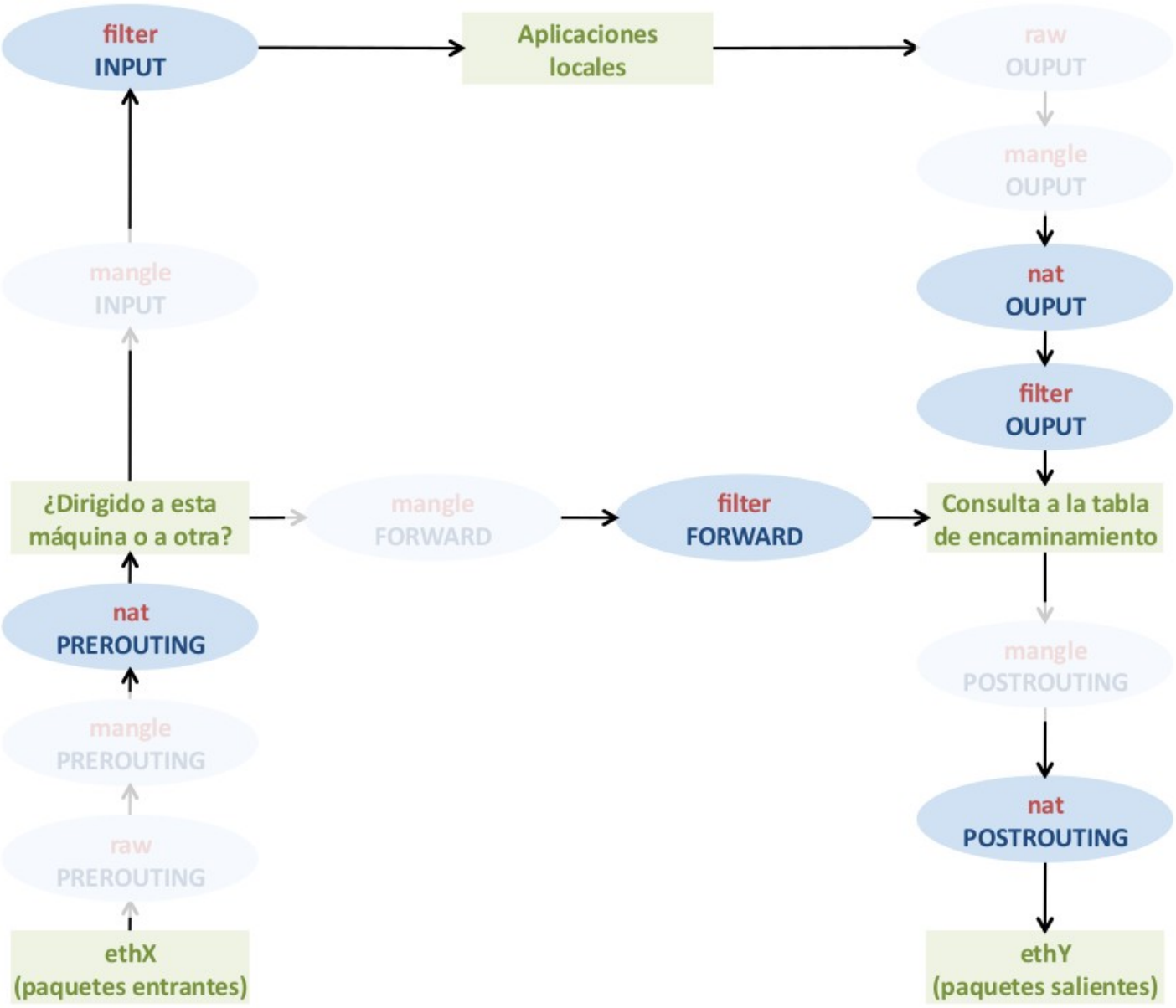


Fig. Táboas e cadeas predeterminadas máis usadas en netfilter. GSyC. CC BY-SA 2.1 ES

Resolución DNS

Seguimos o mesmo procedemento que en puntos anteriores:

```
sad@fw:~$ sudo iptables -N DNS_DMZ
sad@fw:~$ sudo iptables -A FORWARD -i eth2 -p udp --dport 53 -d 8.8.8.8 -m conntrack --ctstate NEW -j
DNS_DMZ
sad@fw:~$ sudo iptables -A FORWARD -i eth2 -p udp --dport 53 -d 8.8.4.4 -m conntrack --ctstate NEW -j
DNS_DMZ
sad@fw:~$ sudo iptables -A DNS_DMZ -s 172.16.0.2 -j ACCEPT
sad@fw:~$ sudo iptables -A DNS_DMZ -j NFLOG --nflog-prefix "iptables: DNS DMZ Bloqueo "
sad@fw:~$ sudo iptables -A DNS_DMZ -j DROP
sad@fw:~$ sudo iptables -L FORWARD -v -n --line-numbers
Chain FORWARD (policy DROP 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1        0     0 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           ctstate
RELATED,ESTABLISHED
2        0     0 DNS_INTRA  udp  --  *      *       0.0.0.0/0            208.67.222.222      udp dpt:53
ctstate NEW
3        0     0 DNS_INTRA  udp  --  *      *       0.0.0.0/0            208.67.220.220      udp dpt:53
ctstate NEW
4        0     0 WEB_INTRA  tcp  --  eth1   *       0.0.0.0/0            0.0.0.0/0           multiport
dports 80,443 ctstate NEW
```

```
5        0     0 SSH_SERVER  tcp  --  *      *       0.0.0.0/0            172.16.0.2           tcp dpt:22
ctstate NEW
6        0     0 DNS_DMZ     udp  --  eth2   *       0.0.0.0/0            8.8.8.8              udp dpt:53
ctstate NEW
7        0     0 DNS_DMZ     udp  --  eth2   *       0.0.0.0/0            8.8.4.4              udp dpt:53
ctstate NEW
sad@fw:~$ sudo iptables -L DNS_DMZ -n --line-numbers
Chain DNS_DMZ (2 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  172.16.0.2           0.0.0.0/0
2    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0            nflog-prefix  "iptables: DNS DMZ Bloqueo
"
3    DROP       all  --  0.0.0.0/0            0.0.0.0/0
```

Acceso repositorios

Revisando os arquivo de repositorios haberá que autorizar o acceso por http ós equipos archive.ubuntu.com e security.ubuntu.com. Nesta ocación usamos unha cadea de usuario e derivamos o tráfico hacia ela cando se cumpran á vez que:

- Entra no FW pola interfaz eth2 (dende a rede DMZ) e sairá pola interfaz eth0 (hacia Internet).
- Procede do server DMZ.
- É un intento de conexión dirixido ó porto tcp/80.

Unha vez dentro da cadea, únicamente autorízanse os accesos ós repositorios oficiais. Fixarse en como estamos a definir condicións máis estrictas á hora de cumprir coa regra; aínda que nos apuntamentos primamos a aprendizaxe e polo tanto ás veces simplificamos e non optimizamos, ser o máis estrictos posible e optimizar na medida do posible debería ser o noso comportamento habitual.

```
sad@fw:~$ sudo iptables -N REPOS_DMZ
sad@fw:~$ sudo iptables -A FORWARD -i eth2 -o eth0 -s 172.16.0.2 -p tcp --dport 80 -m conntrack --ctstate
NEW -j REPOS_DMZ
sad@fw:~$ sudo iptables -A REPOS_DMZ -d archive.ubuntu.com -j ACCEPT
sad@fw:~$ sudo iptables -A REPOS_DMZ -d security.ubuntu.com -j ACCEPT
sad@fw:~$ sudo iptables -A REPOS_DMZ -j NFLOG --nflog-prefix "iptables: REPOS DMZ Bloqueo "
sad@fw:~$ sudo iptables -A REPOS_DMZ -j DROP
sad@fw:~$ sudo iptables -L FORWARD -v -n --line-numbers
Chain FORWARD (policy DROP 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1        0     0 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0            ctstate
RELATED,ESTABLISHED
2        0     0 DNS_INTRA  udp  --  *      *       0.0.0.0/0            208.67.222.222       udp dpt:53
ctstate NEW
3        0     0 DNS_INTRA  udp  --  *      *       0.0.0.0/0            208.67.220.220       udp dpt:53
ctstate NEW
4        0     0 WEB_INTRA  tcp  --  eth1   *       0.0.0.0/0            0.0.0.0/0            multiport
dports 80,443 ctstate NEW
5        0     0 SSH_SERVER  tcp  --  *      *       0.0.0.0/0            172.16.0.2           tcp dpt:22
ctstate NEW
6        0     0 DNS_DMZ    udp  --  eth2   *       0.0.0.0/0            8.8.8.8              udp dpt:53
ctstate NEW
7        0     0 DNS_DMZ    udp  --  eth2   *       0.0.0.0/0            8.8.4.4              udp dpt:53
ctstate NEW
8        0     0 REPOS_DMZ  tcp  --  eth2   eth0    172.16.0.2           0.0.0.0/0            tcp dpt:80
ctstate NEW
sad@fw:~$ sudo iptables -L REPOS_DMZ -n --line-numbers
Chain REPOS_DMZ (1 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            91.189.88.161
2    ACCEPT     all  --  0.0.0.0/0            91.189.88.149
3    ACCEPT     all  --  0.0.0.0/0            91.189.88.152
4    ACCEPT     all  --  0.0.0.0/0            91.189.91.23
5    ACCEPT     all  --  0.0.0.0/0            91.189.88.162
6    ACCEPT     all  --  0.0.0.0/0            91.189.91.26
7    ACCEPT     all  --  0.0.0.0/0            91.189.88.162
8    ACCEPT     all  --  0.0.0.0/0            91.189.88.152
9    ACCEPT     all  --  0.0.0.0/0            91.189.91.23
10   ACCEPT     all  --  0.0.0.0/0            91.189.88.161
11   ACCEPT     all  --  0.0.0.0/0            91.189.88.149
```

```
12   ACCEPT     all  --  0.0.0.0/0            91.189.92.150
13   ACCEPT     all  --  0.0.0.0/0            91.189.91.15
14   ACCEPT     all  --  0.0.0.0/0            91.189.92.191
15   NFLOG      all  --  0.0.0.0/0            0.0.0.0/0           nflog-prefix  "iptables: REPOS DMZ
Bloqueo "
16   DROP       all  --  0.0.0.0/0            0.0.0.0/0
sad@fw:~$
```

Acceso por ssh dende a casa do administrador

Nun punto anterior xa creamos unha cadea de usuario para o tráfico ssh hacia o server:

```
sad@fw:~$ sudo iptables -L FORWARD -v -n --line-numbers
Chain FORWARD (policy DROP 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1        0     0 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0            ctstate
RELATED,ESTABLISHED
2        0     0 DNS_INTRA  udp  --  *      *       0.0.0.0/0            208.67.222.222       udp dpt:53
ctstate NEW
3        0     0 DNS_INTRA  udp  --  *      *       0.0.0.0/0            208.67.220.220       udp dpt:53
ctstate NEW
4        0     0 WEB_INTRA  tcp  --  eth1   *       0.0.0.0/0            0.0.0.0/0            multiport
dports 80,443 ctstate NEW
5        0     0 SSH_SERVER  tcp  --  *      *       0.0.0.0/0            172.16.0.2           tcp dpt:22
ctstate NEW
6        0     0 DNS_DMZ    udp  --  eth2   *       0.0.0.0/0            8.8.8.8              udp dpt:53
ctstate NEW
7        0     0 DNS_DMZ    udp  --  eth2   *       0.0.0.0/0            8.8.4.4              udp dpt:53
ctstate NEW
8        0     0 REPOS_DMZ  tcp  --  eth2   eth0    172.16.0.2           0.0.0.0/0            tcp dpt:80
ctstate NEW
sad@fw:~$ sudo iptables -L SSH_SERVER -v -n --line-numbers
Chain SSH_SERVER (1 references)
num   pkts bytes target     prot opt in     out     source               destination
1        0     0 ACCEPT     all  --  eth1   eth2    192.168.100.1        0.0.0.0/0
2        0     0 ACCEPT     all  --  eth1   eth2    192.168.100.2        0.0.0.0/0
3        0     0 NFLOG      all  --  *      *       0.0.0.0/0            0.0.0.0/0            nflog-prefix
"iptables: SSH SERVER Bloqueo "
4        0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
sad@fw:~$
```

Engadimos unha nova regra autorizando as conexións dende a casa do administrador na posición 3 da cadea SSH_SERVER:

```
sad@fw:~$ sudo iptables -I SSH_SERVER 3 -i eth0 -o eth2 -s 192.0.2.1 -j ACCEPT
sad@fw:~$ sudo iptables -L SSH_SERVER -v -n --line-numbers
Chain SSH_SERVER (1 references)
num   pkts bytes target     prot opt in     out     source               destination
1        0     0 ACCEPT     all  --  eth1   eth2    192.168.100.1        0.0.0.0/0
2        0     0 ACCEPT     all  --  eth1   eth2    192.168.100.2        0.0.0.0/0
3        0     0 ACCEPT     all  --  eth0   eth2    192.0.2.1            0.0.0.0/0
4        0     0 NFLOG      all  --  *      *       0.0.0.0/0            0.0.0.0/0            nflog-prefix
"iptables: SSH SERVER Bloqueo "
5        0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
sad@fw:~$
```

Acceso ó servizo web http/https

Engadimos unha regra para autorizar o tráfico web hacia o servidor:

```
sad@fw:~$ sudo iptables -A FORWARD -p tcp -m multiport --dports 80,443 -d 172.16.0.2 -j ACCEPT
sad@fw:~$ sudo iptables -L FORWARD -n --line-numbers
Chain FORWARD (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
2    DNS_INTRA  udp  --  0.0.0.0/0            208.67.222.222       udp dpt:53 ctstate NEW
3    DNS_INTRA  udp  --  0.0.0.0/0            208.67.220.220       udp dpt:53 ctstate NEW
4    WEB_INTRA  tcp  --  0.0.0.0/0            0.0.0.0/0            multiport dports 80,443 ctstate NEW
5    SSH_SERVER  tcp  --  0.0.0.0/0            172.16.0.2           tcp dpt:22 ctstate NEW
6    DNS_DMZ    udp  --  0.0.0.0/0            8.8.8.8              udp dpt:53 ctstate NEW
7    DNS_DMZ    udp  --  0.0.0.0/0            8.8.4.4              udp dpt:53 ctstate NEW
8    REPOS_DMZ  tcp  --  172.16.0.2           0.0.0.0/0            tcp dpt:80 ctstate NEW
9    ACCEPT     tcp  --  0.0.0.0/0            172.16.0.2           multiport dports 80,443
```

Sneaky Rules

Debido ás particularidades dos equipos que se ubican nunha DMZ debería denegarse o tráfico de saída dos servidores da DMZ (conexións iniciadas polos servidores). A misión dos servidores é recibir peticións dos clientes e responder co resultado; é dicir, a conexión é iniciada por un cliente e autorizada polo firewall, polo que as respostas dos servidores estarán autorizadas para saír (stateful firewall-firewall de estado). O que se está a impedir con esta regra é que os servidores inicien conexións hacia fóra (que normalmente non son necesarias para desempeñar as súas funcións). No caso de ser necesario, deberían permitirse únicamente as conexións creando as regras correspondentes da forma máis restritiva posible. Xa creamos as regras que autorizan o tráfico que pode iniciar o server e procedemos a rexistrar e bloquear o resto:

```
sad@fw:~$ sudo iptables -N SNEAKY
sad@fw:~$ sudo iptables -I INPUT 9 -i eth2 -j SNEAKY
sad@fw:~$ sudo iptables -A FORWARD -i eth2 -j SNEAKY
sad@fw:~$ sudo iptables -A SNEAKY -j NFLOG --nflog-prefix "iptables: Sneaky Rule"
sad@fw:~$ sudo iptables -A SNEAKY -j DROP
sad@fw:~$ sudo iptables -L INPUT -n -v --line-numbers
Chain INPUT (policy DROP 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1        3   420 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           ctstate
RELATED,ESTABLISHED
2        0     0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0           state NEW
3        0     0 SSH        tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
ctstate NEW
4        0     0 DROP       all  --  *      *       0.0.0.0/0            192.0.2.255
5        0     0 DROP       all  --  *      *       0.0.0.0/0            172.16.255.255
6        0     0 DROP       all  --  *      *       0.0.0.0/0            192.168.100.255
7        0     0 DROP       all  --  *      *       0.0.0.0/0            255.255.255.255
8        0     0 DROP       all  --  *      *       0.0.0.0/0            224.0.0.0/4
9        0     0 SNEAKY     all  --  eth2   *       0.0.0.0/0            0.0.0.0/0
10       0     0 NFLOG      all  --  *      *       0.0.0.0/0            0.0.0.0/0           nflog-prefix
"iptables: CleanUP Rule "
sad@fw:~$ sudo iptables -L FORWARD -n -v --line-numbers
Chain FORWARD (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           ctstate RELATED,ESTABLISHED
2    DNS_INTRA  udp  --  0.0.0.0/0            208.67.222.222      udp dpt:53 ctstate NEW
3    DNS_INTRA  udp  --  0.0.0.0/0            208.67.220.220      udp dpt:53 ctstate NEW
4    WEB_INTRA  tcp  --  0.0.0.0/0            0.0.0.0/0           multiport dports 80,443 ctstate NEW
5    SSH_SERVER  tcp  --  0.0.0.0/0            172.16.0.2          tcp dpt:22 ctstate NEW
6    DNS_DMZ    udp  --  0.0.0.0/0            8.8.8.8             udp dpt:53 ctstate NEW
7    DNS_DMZ    udp  --  0.0.0.0/0            8.8.4.4             udp dpt:53 ctstate NEW
8    REPOS_DMZ  tcp  --  172.16.0.2           0.0.0.0/0           tcp dpt:80 ctstate NEW
9    ACCEPT     tcp  --  0.0.0.0/0            172.16.0.2          multiport dports 80,443
10   SNEAKY     all  --  0.0.0.0/0            0.0.0.0/0
sad@fw:~$ sudo iptables -L SNEAKY -n --line-numbers
Chain SNEAKY (2 references)
num  target     prot opt source               destination
1    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0           nflog-prefix  "iptables: Sneaky Rule"
2    DROP       all  --  0.0.0.0/0            0.0.0.0/0
sad@fw:~$
```

Para rematar engadimos na cadea FORWARD unha regra final para rexistrar todo o resto do tráfico non autorizado antes de eliminalo coa policy DROP:

```
sad@fw:~$ sudo iptables -A FORWARD -j NFLOG --nflog-prefix "iptables: CleanUP Rule "
```

Regras NAT

Regras NAT para tráfico iniciado dende o server

Xa temos as regras de filtrado preparadas e engadimos unha regra SNAT:

```
sad@fw:~$ sudo iptables -t nat -A POSTROUTING -s 172.16.0.0/16 -j SNAT --to-source 192.0.2.254
sad@fw:~$ sudo iptables -t nat -L -n --line-numbers
Chain PREROUTING (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  192.168.100.0/24     172.16.0.0/16
```

```
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
Chain POSTROUTING (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  192.168.100.0/24     172.16.0.0/16
2    SNAT       all  --  192.168.100.0/24     0.0.0.0/0            to:192.0.2.254
3    SNAT       all  --  172.16.0.0/16        0.0.0.0/0            to:192.0.2.254
sad@fw:~$
```

Igual que antes a regra SNAT coma orixe toda a DMZ sen preocuparnos de incluír máis equipos dos indicados no enunciado; xa que o tráfico iniciado por eses equipos será bloqueado nas regras de filtrado, e xa non chegará a cadea POSTROUTING.

Regra NAT para tráfico iniciado dende Internet hacia o servizo http/https server

Para permitir o acceso ó servizo web http/https creamos unha regra DNAT que asocie a IP pública de FW portos 80/tcp e 443/tcp coa IP do server portos 80/tcp e 443/tcp:

```
sad@fw:~$ sudo iptables -t nat -A PREROUTING -p tcp -m multiport --dports 80,443 -d 192.0.2.254 -j DNAT --
to-destination 172.16.0.2
sad@fw:~$ sudo iptables -t nat -L -n --line-numbers
Chain PREROUTING (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  192.168.100.0/24     172.16.0.0/16
2    DNAT       tcp  --  0.0.0.0/0            192.0.2.254           multiport dports 80,443 to:172.16.0.2
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
Chain POSTROUTING (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  192.168.100.0/24     172.16.0.0/16
2    SNAT       all  --  192.168.100.0/24     0.0.0.0/0            to:192.0.2.254
3    SNAT       all  --  172.16.0.0/16        0.0.0.0/0            to:192.0.2.254
sad@fw:~$
```

Regra NAT para tráfico iniciado dende Internet hacia o server

Hai que ter cuidado coa forma en que se fai NAT para este tráfico; xa que, podemos crear un conflicto de portos. A tendencia natural sería facer a seguinte regra:

```
sad@fw:~$ sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -d 192.0.2.254 -j DNAT --to-destination
172.16.0.2
```

O problema xurde se temos en conta as regras de filtrados que permiten o acceso por ssh ó FW:

```
sad@fw:~$ sudo iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW -j SSH
sad@fw:~$ sudo iptables -A SSH -s 192.0.2.1 -j ACCEPT
```

O socket `<IP WAN, tcp, 22>` estase a usar para dúas cousas á vez: acceder por ssh ó propio FW e acceder por ssh ó server da DMZ. Qué pasa se quero administrar de forma remota FW dende casa por ssh? O que sucede cando se faga ssh á IP WAN vese claramente se compilamos as regras e ollamos como funcionar netfilter:

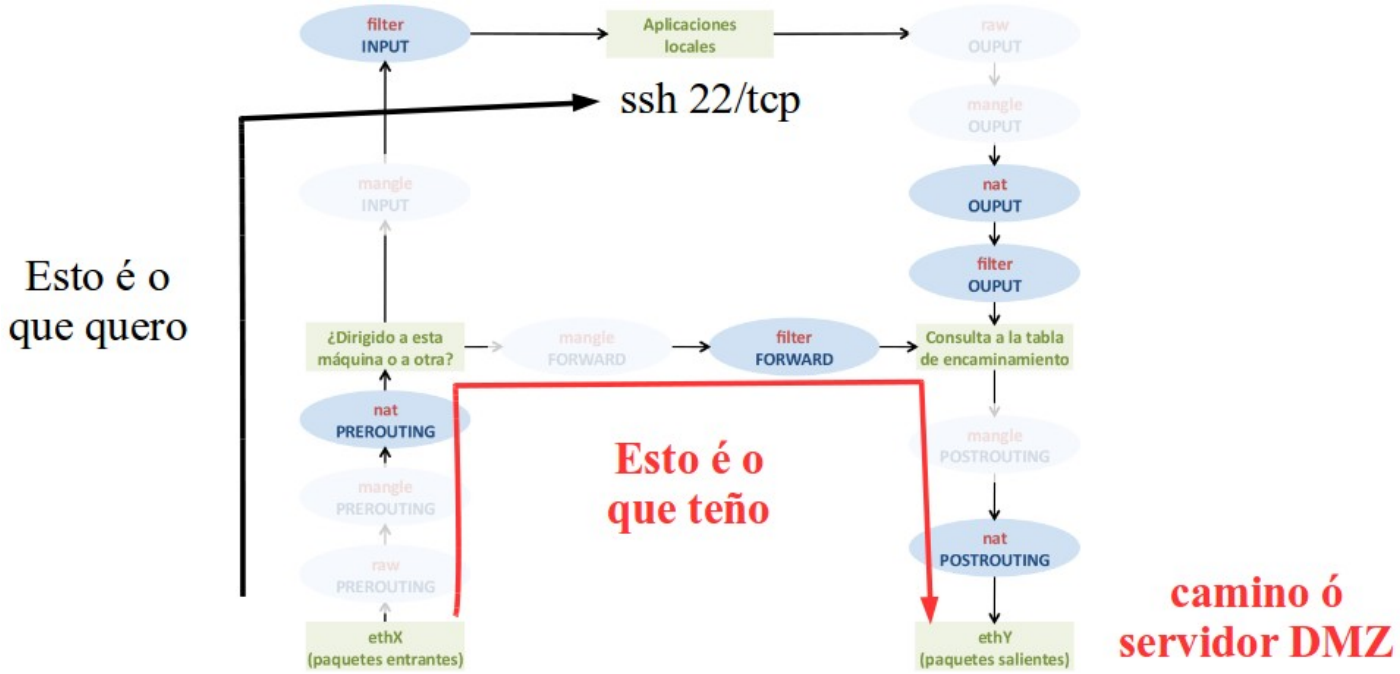


Fig. Conflicto de portos e procesamento dos paquetes en netfiltter

O conflicto de portos pódese solucionar de diversas maneiras. A continuación presentamos tres posibles solucións:

- Sol. nº 1: cambiar o porto de traballo do ssh en FW e arranxar regras antilockdown e stealth. Por exemplo:
  - ssh en FW traballando en porto 22000/tcp
  - En netfilter permitir acceso á IP Pública de FW – porto 22000/tcp e facer a regra NAT como se indicou.
- Sol. nº 2: cambiar o porto de traballo do ssh no servidor e arranxar regras NAT e de filtrado asociadas. Por exemplo:
  - ssh en servidor traballando no porto 22000/tcp
  - En netfilter:
    - regra NAT mapeando: IP Pública FW – porto 22000/tcp con IP Servidor – porto 22000/tcp
    - regra de filtrado permitindo acceso a IP Servidor – porto 22000/tcp
- Sol. Nº 3: SSH en FW e servidor traballando no porto 22/tcp
  - En netfilter:
    - regra NAT mapeando: IP Pública FW – porto 22000/tcp con IP Servidor – porto 22/tcp
    - regra de filtrado permitindo acceso a IP Servidor – porto 22/tcp

Nos faremos a solución nº 3:

```
sad@fw:~$ sudo iptables -t nat -A PREROUTING -p tcp --dport 22000 -d 192.0.2.254 -j DNAT --to-destination
172.16.0.2:22
sad@fw:~$ sudo iptables -t nat -L -n --line-numbers
Chain PREROUTING (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  192.168.100.0/24     172.16.0.0/16
2    DNAT       tcp  --  0.0.0.0/0            192.0.2.254          multiport dports 80,443 to:172.16.0.2
3    DNAT       tcp  --  0.0.0.0/0            192.0.2.254          tcp dpt:22000 to:172.16.0.2:22
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
Chain POSTROUTING (policy ACCEPT)
```

```
num  target     prot opt source               destination
1    ACCEPT     all  --  192.168.100.0/24     172.16.0.0/16
2    SNAT       all  --  192.168.100.0/24     0.0.0.0/0            to:192.0.2.254
3    SNAT       all  --  172.16.0.0/16        0.0.0.0/0            to:192.0.2.254
sad@fw:~$
```

Polo tanto:

- A regra de filtrado autorizando o acceso ó porto 22/tcp do server non hai que tocala nin tampouco hai que cambiar o porto de traballo do servidor ssh correndo en server.
- A regra de filtrado autorizando o acceso ó porto 22/tcp de FW non hai que tocala nin tampouco hai que cambiar o porto de traballo do servidor ssh correndo en FW Linux

Para administrar por ssh FW habería que facer:

- ssh usuario@192.0.2.254 --> dende Internet.
- ssh usuario@192.168.100.254 --> dende a intranet.

Para administrar por ssh o server habería que facer:

- ssh usuario@192.0.2.254 -p 22000 --> dende Internet.
- ssh usuario@172.16.0.2 --> dende a intranet (lembrade que para as comunicacións internas non hai NAT e pódese acceder ó porto tcp/22 do ssh correndo en server).

5. Rexistros

Durante a creación das regras na cadea de usuario SSH vimos como rexistrar paquetes con LOG e NFLOG.

```
sad@fw:~$ sudo iptables -L SSH -n --line-numbers
Chain SSH (1 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  192.0.2.1            0.0.0.0/0
2    ACCEPT     all  --  192.168.100.1        0.0.0.0/0
3    ACCEPT     all  --  192.168.100.2        0.0.0.0/0
4    NFLOG      all  --  0.0.0.0/0            0.0.0.0/0            nflog-prefix  "iptables: SSH Bloqueo "
5    DROP       all  --  0.0.0.0/0            0.0.0.0/0
```

As incidencias rexistraranse no arquivo `/var/log/ulog/syslogemu.log` :

```
sad@fw:~$ sudo ls -lahF /var/log/ulog/
total 14K
drwxr-xr-x 2 root root      3 Oct 16 11:37 ./
drwxrwxr-x 8 root syslog   17 Oct 16 11:37 ../
-rw-r--r-- 1 root root   1.1K Oct 16 15:25 syslogemu.log
sad@fw:~$ sudo cat /var/log/ulog/syslogemu.log
Oct 16 15:25:26 fw iptables: SSH Bloqueo  IN=eth1 OUT= MAC=00:16:3e:70:97:3b:00:16:3e:17:18:d1:08:00
SRC=192.168.100.20 DST=192.168.100.254 LEN=60 TOS=00 PREC=0x00 TTL=64 ID=45794 DF PROTO=TCP SPT=53424
DPT=22 SEQ=4076628094 ACK=0 WINDOW=29200 SYN URGP=0 MARK=0
Oct 16 15:25:28 fw iptables: SSH Bloqueo  IN=eth1 OUT= MAC=00:16:3e:70:97:3b:00:16:3e:17:18:d1:08:00
SRC=192.168.100.20 DST=192.168.100.254 LEN=60 TOS=00 PREC=0x00 TTL=64 ID=45795 DF PROTO=TCP SPT=53424
DPT=22 SEQ=4076628094 ACK=0 WINDOW=29200 SYN URGP=0 MARK=0
Oct 16 15:25:30 fw iptables: SSH Bloqueo  IN=eth1 OUT= MAC=00:16:3e:70:97:3b:00:16:3e:17:18:d1:08:00
SRC=192.168.100.20 DST=192.168.100.254 LEN=60 TOS=00 PREC=0x00 TTL=64 ID=45796 DF PROTO=TCP SPT=53424
DPT=22 SEQ=4076628094 ACK=0 WINDOW=29200 SYN URGP=0 MARK=0
Oct 16 15:25:34 fw iptables: SSH Bloqueo  IN=eth1 OUT= MAC=00:16:3e:70:97:3b:00:16:3e:17:18:d1:08:00
SRC=192.168.100.20 DST=192.168.100.254 LEN=60 TOS=00 PREC=0x00 TTL=64 ID=45797 DF PROTO=TCP SPT=53424
DPT=22 SEQ=4076628094 ACK=0 WINDOW=29200 SYN URGP=0 MARK=0
sad@fw:~$
```

6. Probas de funcionamento

A continuación faremos unhas sinxelas probas para verificar o correcto funcionamento do sistema:

Resolución dns e acceso ós repositorios en FW:

```
sad@fw:~$ host www.xunta.gal
www.xunta.gal has address 85.91.64.109
sad@fw:~$ sudo apt-get install net-tools
pt install net-tools
```

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfreetype6
Use 'apt autoremove' to remove it.
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 35 not upgraded.
Need to get 196 kB of archives.
After this operation, 864 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 net-tools amd64 1.60+git20180626.aebd88e-1ubuntu1
[196 kB]
Fetched 196 kB in 0s (7844 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 31686 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20180626.aebd88e-1ubuntu1_amd64.deb ...
Unpacking net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Setting up net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
```

Acceso por ssh dende un equipo admin da Intranet a FW:

```
sad@admin2:~$ ssh sad@192.168.100.254
sad@192.168.100.254's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-137-generic x86_64)
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud
7 packages can be updated.
0 updates are security updates.
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
sad@fw:~$
```

Acceso por ssh dende o equipo admin de Internet a FW:

```
magasix@lxd:~⟫ ssh sad@192.0.2.254
sad@192.0.2.254's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-137-generic x86_64)
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud
7 packages can be updated.
0 updates are security updates.
Last login: Sun Oct 21 20:19:15 2018 from 192.168.100.2
sad@fw:~$
```

Acceso por ssh ó server da DMZ dende equipo admin en Internet :

```
magasix@lxd:~⟫ ssh sad@192.0.2.254 -p 22000
ubuntu@192.0.2.254's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-137-generic x86_64)
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud
7 packages can be updated.
0 updates are security updates.
Last login: Sat Oct 20 20:06:39 2018 from 192.0.2.1
sad@serverdmz:~$
```

Acceso a FW por ssh dende un equipo non autorizado dende Internet e entradas no log de FW: fixarse que ó usar un descarte silencioso con DROP prodúcese un time out.

```
root@gui1604:~# ssh sad@192.0.2.254
ssh: connect to host 192.0.2.254 port 22: Connection timed out
root@gui1604:~#
```

```
Oct 21 20:26:35 fw iptables: SSH FW Bloqueo  IN=eth0 OUT= MAC=00:16:3e:bd:b5:42:00:16:3e:87:0d:e1:08:00
SRC=192.0.2.130 DST=192.0.2.254 LEN=60 TOS=00 PREC=0x00 TTL=64 ID=35676 DF PROTO=TCP SPT=45766 DPT=22
SEQ=243301880 ACK=0 WINDOW=29200 SYN URGP=0 MARK=0
Oct 21 20:26:51 fw iptables: SSH FW Bloqueo  IN=eth0 OUT= MAC=00:16:3e:bd:b5:42:00:16:3e:87:0d:e1:08:00
SRC=192.0.2.130 DST=192.0.2.254 LEN=60 TOS=00 PREC=0x00 TTL=64 ID=35677 DF PROTO=TCP SPT=45766 DPT=22
SEQ=243301880 ACK=0 WINDOW=29200 SYN URGP=0 MARK=0
Oct 21 20:27:23 fw iptables: SSH FW Bloqueo  IN=eth0 OUT= MAC=00:16:3e:bd:b5:42:00:16:3e:87:0d:e1:08:00
SRC=192.0.2.130 DST=192.0.2.254 LEN=60 TOS=00 PREC=0x00 TTL=64 ID=35678 DF PROTO=TCP SPT=45766 DPT=22
SEQ=243301880 ACK=0 WINDOW=29200 SYN URGP=0 MARK=0
```

Acceso a server da DMZ por ssh dende un equipo non autorizado dende Internet e entradas no log de FW:

```
root@gui1604:~# ssh sad@192.0.2.254 -p 22000
ssh: connect to host 192.0.2.254 port 22000: Connection timed out
root@gui1604:~#
Oct 21 20:29:04 fw iptables: SSH SERVER Bloqueo  IN=eth0 OUT=eth2
MAC=00:16:3e:bd:b5:42:00:16:3e:87:0d:e1:08:00 SRC=192.0.2.130 DST=172.16.0.2 LEN=60 TOS=00 PREC=0x00 TTL=63
ID=1086 DF PROTO=TCP SPT=52720 DPT=22 SEQ=1986499080 ACK=0 WINDOW=29200 SYN URGP=0 MARK=0
Oct 21 20:29:05 fw iptables: SSH SERVER Bloqueo  IN=eth0 OUT=eth2
MAC=00:16:3e:bd:b5:42:00:16:3e:87:0d:e1:08:00 SRC=192.0.2.130 DST=172.16.0.2 LEN=60 TOS=00 PREC=0x00 TTL=63
ID=1087 DF PROTO=TCP SPT=52720 DPT=22 SEQ=1986499080 ACK=0 WINDOW=29200 SYN URGP=0 MARK=0
Oct 21 20:29:07 fw iptables: SSH SERVER Bloqueo  IN=eth0 OUT=eth2
MAC=00:16:3e:bd:b5:42:00:16:3e:87:0d:e1:08:00 SRC=192.0.2.130 DST=172.16.0.2 LEN=60 TOS=00 PREC=0x00 TTL=63
ID=1088 DF PROTO=TCP SPT=52720 DPT=22 SEQ=1986499080 ACK=0 WINDOW=29200 SYN URGP=0 MARK=0
Oct 21 20:29:11 fw iptables: SSH SERVER Bloqueo  IN=eth0 OUT=eth2
MAC=00:16:3e:bd:b5:42:00:16:3e:87:0d:e1:08:00 SRC=192.0.2.130 DST=172.16.0.2 LEN=60 TOS=00 PREC=0x00 TTL=63
ID=1089 DF PROTO=TCP SPT=52720 DPT=22 SEQ=1986499080 ACK=0 WINDOW=29200 SYN URGP=0 MARK=0
```

Instalación de Apache no servidor dmz e habilitación do servizo https:

```
sad@serverdmz:~$ sudo apt-get update
sad@serverdmz:~$ sudo apt-get install apache2
sad@serverdmz:~$ sudo a2enmod ssl
sad@serverdmz:~$ sudo a2ensite default-ssl.conf
sad@serverdmz:~$ sudo service apache2 reload
sad@serverdmz:~$ sudo netstat -putan
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      317/sshd
tcp6       0      0 :::80                   :::*                    LISTEN      417/apache2
tcp6       0      0 :::22                   :::*                    LISTEN      317/sshd
tcp6       0      0 :::443                  :::*                    LISTEN      417/apache2
sad@serverdmz:~$
```

Acceso dende Internet ó servidor Web por https e entradas no conntrack:

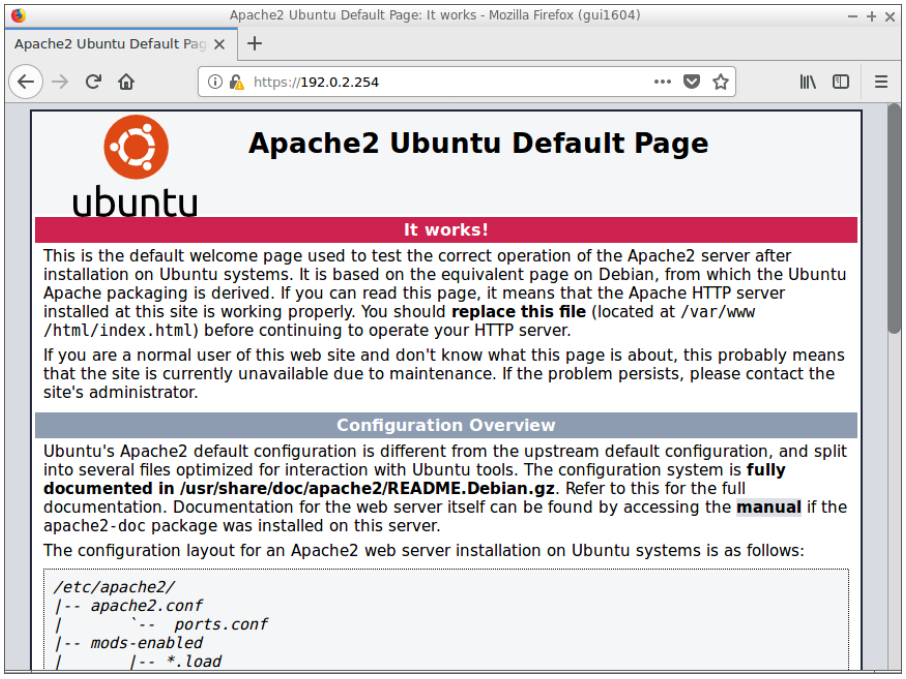


Fig. Acceso por https ó servizo web do servidor da DMZ

Fixarse que na saída de conntrack aparece en primeiro lugar información da conexión no sentido no que se orixinou (192.0.2.1 --> 192.0.2.254) e despois aparece no outro sentido (172.16.0.2 --> 192.0.2.1).

```
sad@fw:~$ sudo apt-get install conntrack
sad@fw:~$ sudo conntrack -L
tcp      6 3 CLOSE src=192.0.2.1 dst=192.0.2.254 sport=41776 dport=443 src=172.16.0.2 dst=192.0.2.1
sport=443 dport=41776 [ASSURED] mark=0 use=1
tcp      6 3 CLOSE src=192.0.2.1 dst=192.0.2.254 sport=41774 dport=443 src=172.16.0.2 dst=192.0.2.1
sport=443 dport=41774 [ASSURED] mark=0 use=1
udp      17 18 src=172.16.0.2 dst=8.8.8.8 sport=43693 dport=53 src=8.8.8.8 dst=192.0.2.254 sport=53
dport=43693 [ASSURED] mark=0 use=1
udp      17 18 src=172.16.0.2 dst=8.8.8.8 sport=58601 dport=53 src=8.8.8.8 dst=192.0.2.254 sport=53
dport=58601 [ASSURED] mark=0 use=1
tcp      6 431997 ESTABLISHED src=192.0.2.1 dst=192.0.2.254 sport=41780 dport=443
src=172.16.0.2 dst=192.0.2.1 sport=443 dport=41780 [ASSURED] mark=0 use=1
conntrack v1.4.3 (conntrack-tools): 5 flow entries have been shown.
```

Se probamos a xerar tráfico, tanto dende Internet hacia ó server como dende a intranet hacia Internet, e usamos os opcións de filtrado -n e -g de conntrack poderemos filtrar as conexións con SNAT e DNAT respectivamente:

```
sad@fw:~$ sudo conntrack -L -n
udp      17 165 src=172.16.0.2 dst=8.8.8.8 sport=38318 dport=53 src=8.8.8.8 dst=192.0.2.254 sport=53
dport=38318 [ASSURED] mark=0 use=1
udp      17 15 src=172.16.0.2 dst=8.8.8.8 sport=59680 dport=53 src=8.8.8.8 dst=192.0.2.254 sport=53
dport=59680 mark=0 use=1
tcp      6 106 TIME_WAIT src=172.16.0.2 dst=91.189.88.149 sport=52956 dport=80 src=91.189.88.149
dst=192.0.2.254 sport=80 dport=52956 [ASSURED] mark=0 use=1
tcp      6 106 TIME_WAIT src=172.16.0.2 dst=91.189.91.26 sport=56994 dport=80
src=91.189.91.26 dst=192.0.2.254 sport=80 dport=56994 [ASSURED] mark=0 use=1
udp      17 165 src=172.16.0.2 dst=8.8.8.8 sport=59112 dport=53 src=8.8.8.8 dst=192.0.2.254 sport=53
dport=59112 [ASSURED] mark=0 use=1
udp      17 15 src=172.16.0.2 dst=8.8.8.8 sport=47979 dport=53 src=8.8.8.8 dst=192.0.2.254 sport=53
dport=47979 mark=0 use=1
conntrack v1.4.3 (conntrack-tools): 6 flow entries have been shown.
sad@fw:~$ sudo conntrack -L -g
tcp      6 104 TIME_WAIT src=192.0.2.1 dst=192.0.2.254 sport=41786 dport=443 src=172.16.0.2 dst=192.0.2.1
sport=443 dport=41786 [ASSURED] mark=0 use=1
tcp      6 431995 ESTABLISHED src=192.0.2.1 dst=192.0.2.254 sport=37600 dport=22000
src=172.16.0.2 dst=192.0.2.1 sport=22 dport=37600 [ASSURED] mark=0 use=1
conntrack v1.4.3 (conntrack-tools): 2 flow entries have been shown.
sad@fw:~$
```

Para dar por válidas as regras deberiamos realizar probas estrictas e completas:

- Verificando que todo o tráfico que queremos autorizar é permitido.
- Comprobando que tráfico non desexado é bloqueado e rexistrado se fose o caso.

Se non o fixeches antes, unha vez probado o firewall non esquecerse de gardar as regras para facelas persistentes :

```
sad@fw:~$ sudo apt-get install iptables-persistent
sad@fw:~$ sudo netfilter-persistent save
```