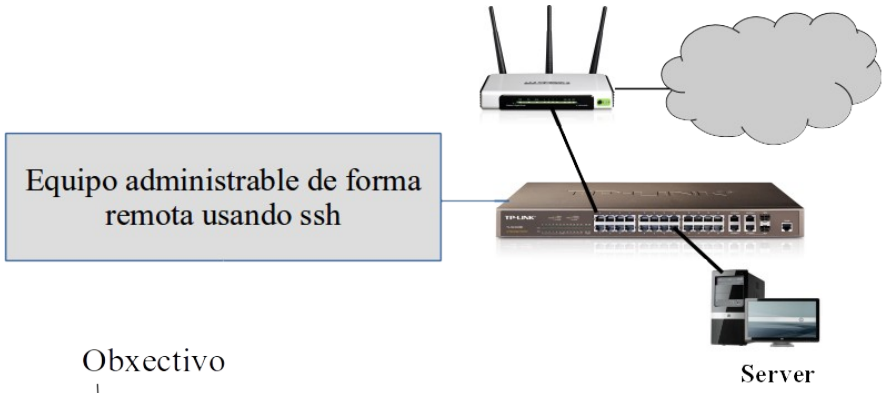
Escenario: Firewall de host con netfilter/iptables (regras con estados e persistentes)

Nesta práctica configurarase un firewall de host nunha máquina Linux Ubuntu para permitir únicamente o tráfico autorizado, denegando por defecto o resto do tráfico. Traballarase con **estados** (**seguimento de conexións**) e faranse as **regras de filtrado persistentes** (non se perden ó reiniciar a máquina).



Obxectivo

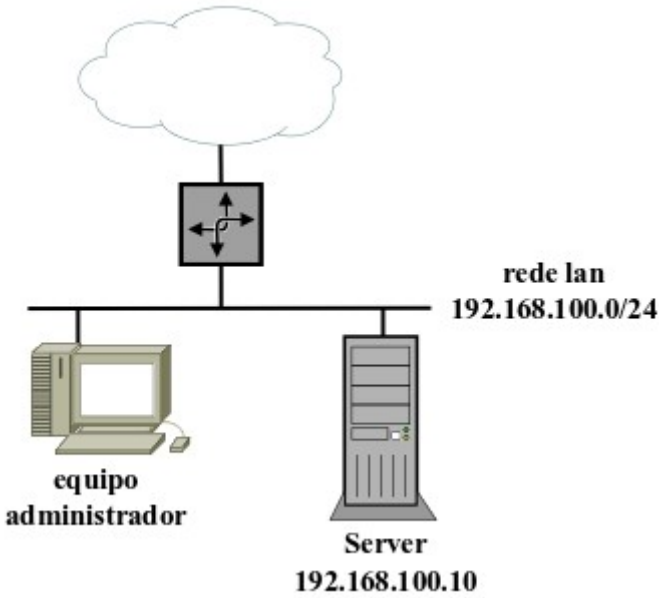
Configurar o firewall a nivel de host no equipo Server permitindo únicamente o tráfico autorizado

Tráfico autorizado:

- Poderá realizar consultas DNS ós servidores 8.8.8.8 y 8.8.4.4.
- Poderá visitar sitios web (http/https).
- Pode ser xestionado dende calquera equipo a través dun servidor ssh correndo no porto por defecto (tcp/22).

Configuración do escenario

Esquema de rede do escenario



- A máquina Server ten un adaptador de rede en modo ponte, configuración de rede estática e saída a Internet.
- Adaptar ás direccións IP á vosa casa.

Configuración do server para o escenario 3B:

- Server ten instalado o servizo ssh:

```
manuel@server:~$ sudo apt-get update
manuel@server:~$ sudo apt-get install openssh-server
```

Estado inicial

Co server correctamente configurado, pode comprobarse que é posible enviar e recibir paquetes sen limitacións. A modo de exemplo, fanse probas ping, navegación web, acceso ftp e resolucións dns para verificar que é posible iniciar conexións dende o server hacia fóra e recibir as respostas.

```
manuel@server:~$ ping -c 2 www.google.com
PING www.google.com (172.217.168.164) 56(84) bytes of data.
64 bytes from mad07s10-in-f4.1e100.net (172.217.168.164): icmp_seq=1 ttl=53 time=17.8 ms
64 bytes from mad07s10-in-f4.1e100.net (172.217.168.164): icmp_seq=2 ttl=53 time=17.1 ms
--- www.google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 17.163/17.512/17.861/0.349 ms
```

```
manuel@server:~$ wget http://www.xunta.gal
--2018-10-14 16:28:15--  http://www.xunta.gal/
Resolving www.xunta.gal (www.xunta.gal)... 85.91.64.109
Connecting to www.xunta.gal (www.xunta.gal)|85.91.64.109|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://www.xunta.gal/ [following]
--2018-10-14 16:28:15--  https://www.xunta.gal/
Connecting to www.xunta.gal (www.xunta.gal)|85.91.64.109|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: /portada [following]
--2018-10-14 16:28:16--  https://www.xunta.gal/portada
Connecting to www.xunta.gal (www.xunta.gal)|85.91.64.109|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 136563 (133K) [text/html]
Saving to: ‘index.html’
index.html      100%
[=======================================================================================>]
133.36K   757KB/s    in 0.2s
2018-10-14 16:28:17 (757 KB/s) - ‘index.html’ saved [136563/136563]
```

```
manuel@server:~$ ftp ftp.rediris.es
Connected to ftp.rediris.es.
220-  Bienvenido al servicio de replicas de RedIRIS.
220-     Welcome to the RedIRIS mirror service.
220 Only anonymous FTP is allowed here
Name (ftp.rediris.es:root): anonymous
230-            RedIRIS - Red Académica y de Investigación Española
230-                RedIRIS - Spanish National Research Network
230-
230-           ftp://ftp.rediris.es  -=-  http://ftp.rediris.es
230 Anonymous user logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> exit
```

```
manuel@server:~$ host www.google.com
www.google.com has address 172.217.16.228
www.google.com has IPv6 address 2a00:1450:4003:809::2004
```

Tameń é posible iniciar conexións dende outros equipos hacia o server. A modo de exemplo faise ping e accédese por ssh dende o equipo do administrador (poemos usar o equipo anfitrión como equipo administrador) ó server:

```
manuel@pc:~$ ping -c 2 192.168.100.10
PING 192.168.100.10 (192.168.100.10) 56(84) bytes of data.
64 bytes from 192.168.100.10: icmp_seq=1 ttl=64 time=0.060 ms
64 bytes from 192.168.100.10: icmp_seq=2 ttl=64 time=0.066 ms
--- 192.168.100.10 ping statistics ---
```

```
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.060/0.063/0.066/0.003 ms
manuel@pc:~$ ssh ubuntu@192.168.100.10
The authenticity of host '192.168.100.10 (192.168.100.10)' can't be established.
ECDSA key fingerprint is SHA256:P4KEMPwUPsUeji0l7xq4d2UJhTL9GK33wBj3oZEoy0o.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.100.10' (ECDSA) to the list of known hosts.
ubuntu@192.168.100.10's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 5.4.0-65-generic x86_64)
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
0 packages can be updated.
0 of these 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.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
manuel@server:~$
```

O anterior é posible xa que **a configuración por defecto de netfilter en Server é permitir todo o tráfico**, tanto entrante como saínte. Esta configuración de permitir todo por defecto pode verificarse facendo un listado das regras da táboa filter:

```
manuel@server:~$ sudo iptables -t filter -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
manuel@server:~$
```

Como pode verse, non hai regras en ningunha das cadeas (INPUT, FORWARD e OUTPUT) da táboa filter; polo que a un paquete procesado nestas cadeas aplicaráselle a **Policy** (acción por defecto da cadea), que é ACCEPT en todas elas.

A opción `-t` do comando iptables permite especificar a táboa na que se está interesado; e no caso de non especificar nada, cóllese a táboa filter por defecto. Polo tanto, o mesmo resultado obtense ó executar:

```
manuel@server:~$ sudo iptables  -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Regras con estados

Netfilter permite traballar con estados e na seguinte regra acéptanse todos os paquetes con destino server que pertencen a unha conexión xa autorizada e establecida:

```
manuel@server:~$ sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

Faise o mesmo para permitir a saída do server de todos os paquetes que pertencen a unha conexión xa autorizada e establecida:

```
manuel@server:~$ sudo iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

Non facemos nada na cadea FORWARD por que estamos a traballar nun firewall de host, que unicamente ten que preocuparse polo tráfico xerado polas súas aplicacións e o tráfico destinado ás súas aplicacións. O tráfico que iría a FORWARD sería o tráfico non xerado polo equipo e non destinado ó equipo; e o noso equipo non é un router, polo que coa policy DROP en FORWARD xa descartamos este tráfico (ademais de non ter activado o enrutamento de pquetes).

Creamos as regras del lazo pechado pensadas para permitir o tráfico do host dirixido ó propio host. Sen elas 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):

```
manuel@server:~$ sudo iptables -A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
manuel@server:~$ sudo iptables -A OUTPUT -o lo -m conntrack --ctstate NEW -j ACCEPT
```

Agora o que falta é autorizar o establecemento de conexións aceptando o primeiro paquete dunha conexión. A seguinte regra permite que o server lance consultas dns ó servidor 8.8.8.8.

```
manuel@server:~$ sudo iptables -A OUTPUT -p udp --dport 53 -d 8.8.8.8 -m conntrack --ctstate NEW -j ACCEPT
```

Unha vez autorizado o primeiro paquete, calquera paquete da conexión será aceptado grazas ás dúas regras anteriores:

- A regra "`OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED`" permitirá a saída dos paquetes necesarios para que o equipo fale co servidor 8.8.8.8.
- A regra "`INPUT -m conntrack --ctstate ESTABLISHED,RELATED`" permitirá a entrada das mensaxes de resposta do servidor 8.8.8.8 á consulta efectuada polo equipo.

A continuación están as regras para autorizar a resolución DNS e a navegación web:

```
manuel@server:~$ sudo iptables -A OUTPUT -p udp --dport 53 -d 8.8.4.4 -m conntrack --ctstate NEW -j ACCEPT
```

```
manuel@server:~$ sudo iptables -A OUTPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate NEW -j ACCEPT
```

Para aprender novas opcións, fixarse que para o tráfico web emprégase **`-m multiport --dports 80,443`** que permite definir varios portos e incluso rangos deles. Tamén pódese crear dúas regras, unha para autorizar as conexión ó porto 80 e outra ó port 443 usando `--dport`.

Para autorizar as conexión por ssh ó server:

```
manuel@server:~$ sudo iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT
```

Por último aplícase unha política de denegar todo o que non foi explícitamente aprobado:

```
manuel@server:~$ sudo iptables -P INPUT DROP
manuel@server:~$ sudo iptables -P OUTPUT DROP
manuel@server:~$ sudo iptables -P FORWARD DROP
```

O ruleset quedaría:

```
manuel@server:~$ sudo iptables -L -n
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           ctstate RELATED,ESTABLISHED
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           ctstate NEW

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           ctstate RELATED,ESTABLISHED
```

```
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           ctstate NEW
ACCEPT     udp  --  0.0.0.0/0            8.8.8.8             udp dpt:53 ctstate NEW
ACCEPT     udp  --  0.0.0.0/0            8.8.4.4             udp dpt:53 ctstate NEW
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           multiport dports 80,443 ctstate NEW
manuel@server:~$
```

Pódese obter máis información empregando as opcións `--line-numbers` e `-v`

```
manuel@server:~$ iptables -L -n --line-numbers -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1      931 2848K ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           ctstate
RELATED,ESTABLISHED
2       10   689 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0           ctstate
NEW

Chain FORWARD (policy DROP 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy DROP 10 packets, 1005 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1      779 52144 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           ctstate
RELATED,ESTABLISHED
2       10   689 ACCEPT     all  --  *      lo      0.0.0.0/0            0.0.0.0/0           ctstate
NEW
3        0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            8.8.8.8             udp
dpt:53 ctstate NEW
4        0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            8.8.4.4             udp
dpt:53 ctstate NEW
5           5    300 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
multiport dports 80,443 ctstate NEW
```

Ademais do número de liña pódese ver a interface de entrada/saída (de estar especificada coma nas regras de loopback) e o nº de pqeutes procesados por cada regra.

A continuación realízanse algunhas probas para verificar o funcionamento do firewall:

```
manuel@server:~$ host www.xunta.es 8.8.4.4
Using domain server:
Name: 8.8.4.4
Address: 8.8.4.4#53
Aliases:
www.xunta.es is an alias for proxygold.xunta.es.
proxygold.xunta.es has address 85.91.64.240
manuel@server:~$ ping -c 2 8.8.4.4
PING 8.8.4.4 (8.8.4.4) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
--- 8.8.4.4 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 999ms
manuel@server:~$  wget http://www.edu.xunta.gal
--2018-10-14 18:16:16--  http://www.edu.xunta.gal/
Resolving www.edu.xunta.gal (www.edu.xunta.gal)... 85.91.64.110
Connecting to www.edu.xunta.gal (www.edu.xunta.gal)|85.91.64.110|:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://www.edu.xunta.gal/portal/ [following]
--2018-10-14 18:16:16--  http://www.edu.xunta.gal/portal/
Connecting to www.edu.xunta.gal (www.edu.xunta.gal)|85.91.64.110|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 75392 (74K) [text/html]
Saving to: ‘index.html.1’
index.html.1                                        100%
[=============================================================================================>]
73.62K  --.-KB/s    in 0.07s
2018-10-14 18:16:16 (1.03 MB/s) - ‘index.html.1’ saved [75392/75392]
manuel@server:~$
```

É posible facer resolucións DNS contra 8.8.4.4 pero non facer ping, xa que o ping non está autorizado.

Dende o equipo anfitrión:

```
manuel@pc:~$ ping -c 2 192.168.100.10
PING 192.168.1.254 (192.168.1.254) 56(84) bytes of data.
--- 192.168.1.254 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1008ms
manuel@pc:~$ ssh ubuntu@192.168.100.10
ubuntu@192.168.100.10'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
0 packages can be updated.
0 updates are security updates.
New release '18.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
Last login: Sun Oct 14 16:31:23 2018 from 192.168.100.1
manuel@server:~$
```

Regras persistentes

As regras creadas con iptables son efímeras e pérdense ó apagar o equipo. Pódese comprobar fácilmente:

```
manuel@server:~$ sudo reboot
```

E tras arricar verifícase que se perderon todas as regras incluíndo a política de denegar todo:

```
manuel@server:~$ sudo iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
manuel@server:~$
```

As regras creadas con iptables **son efímeras** e para non perdelas pódense empregar varios métodos:

- Crear un script cos comandos iptables correspondentes que se carga ó inicio do sistema.
- Usar o servizo `iptables-persistent` que se encarga de recuperar as regras iptables configuradas nos reinicios.

Neste escenario empregarase `iptables-persistent` para facer as regras persistentes. O primeiro paso será recuperar as regras borradas:

```
manuel@server:~$ sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
manuel@server:~$ sudo iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
manuel@server:~$ sudo iptables -A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
manuel@server:~$ sudo iptables -A OUTPUT -o lo -m conntrack --ctstate NEW -j ACCEPT
manuel@server:~$ sudo iptables -A OUTPUT -p udp --dport 53 -d 8.8.8.8 -m conntrack --ctstate NEW -j
ACCEPT
manuel@server:~$ sudo iptables -A OUTPUT -p udp --dport 53 -d 8.8.4.4 -m conntrack --ctstate NEW -j
ACCEPT
manuel@server:~$ sudo iptables -A OUTPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate
NEW -j ACCEPT
manuel@server:~$ sudo iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT
manuel@server:~$ sudo iptables -P INPUT DROP
manuel@server:~$ sudo iptables -P OUTPUT DROP
manuel@server:~$ sudo iptables -P FORWARD DROP
manuel@server:~$ iptables -L -n --line-numbers -v
Chain INPUT (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 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0           ctstate
NEW
3        0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp
dpt:22 ctstate NEW

Chain FORWARD (policy DROP 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (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 ACCEPT     all  --  *      lo      0.0.0.0/0            0.0.0.0/0           ctstate
NEW
3        0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            8.8.8.8             udp
dpt:53 ctstate NEW
4        0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            8.8.4.4             udp
dpt:53 ctstate NEW
5        0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
multiport dports 80,443 ctstate NEW
```

En segundo lugar instalarase o paquete `iptables-persistent`:

```
manuel@server:~$ sudo apt-get update
manuel@server:~$ sudo apt-get install iptables-persistent
```

Durante a instalación pregúntase se se queren gardar as regras actuais (tanto para IPv4 coma para IPv6):

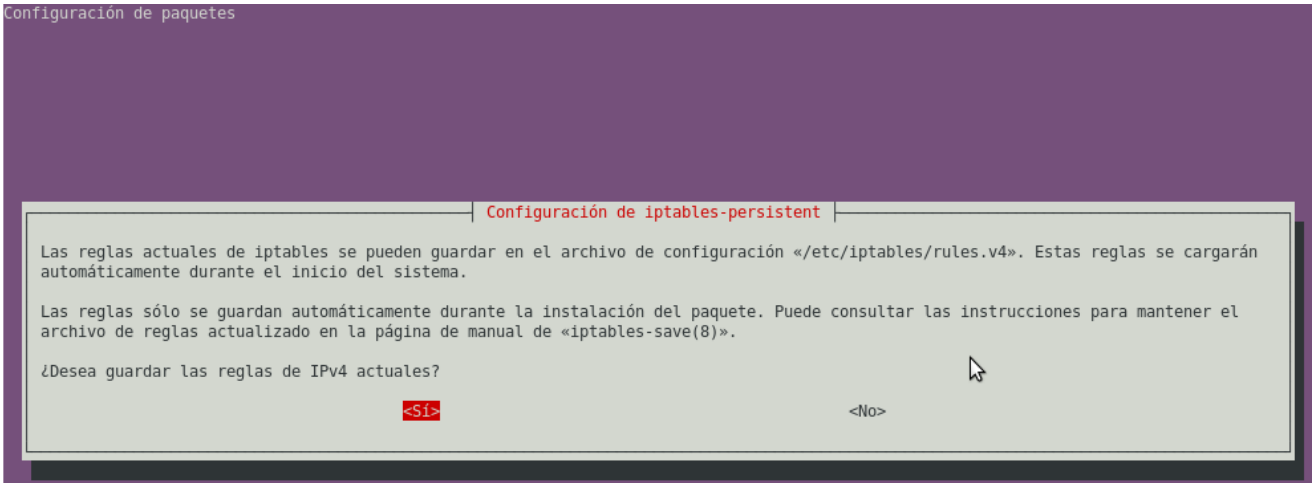


Fig. Configuración de `iptables-persistent` durante a súa instalación

Gardando as regras actuais e unha vez rematada a instalación, no directorio `/etc/iptables` aparecerán dous arquivos coas regras, un para IPv4 e outro para IPv6:

```
manuel@server:~$ ls -lhF /etc/iptables/
total 2.0K
-rw-r----- 1 root root 806 Feb 6 19:36 rules.v4
-rw-r----- 1 root root 183 Feb 6 19:33 rules.v6
```

No arquivo `/etc/iptables/rules.ipv4` atoparanse as regras anteriormente introducidas e gardadas durante a instalación. Podemos ver as táboas, cadeas, policy e regras:

```
manuel@server:~$ cat /etc/iptables/rules.v4
# Generated by iptables-save v1.8.4 on Tue Feb 22 09:13:33 2022
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [1:86]
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -o lo -m conntrack --ctstate NEW -j ACCEPT
```

```
-A OUTPUT -d 8.8.8.8/32 -p udp -m udp --dport 53 -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -d 8.8.4.4/32 -p udp -m udp --dport 53 -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate NEW -j ACCEPT
COMMIT
# Completed on Tue Feb 22 09:13:33 2022
```

Cando o equipo se reinicie, `iptables-persistent` lerá os arquivos de regras e procederá a cargalas:

```
manuel@server:~$ sudo reboot
manuel@server:~$ 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            ctstate NEW
3    ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:22 ctstate NEW

Chain FORWARD (policy DROP)
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            ctstate NEW
3    ACCEPT     udp  --  0.0.0.0/0            8.8.8.8              udp dpt:53 ctstate NEW
4    ACCEPT     udp  --  0.0.0.0/0            8.8.4.4              udp dpt:53 ctstate NEW
5    ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            multiport dports 80,443 ctstate NEW
```

A pregunta que xurde agora é como facer para cambiar o ruleset e que `iptables-persistent` recoñeza eses cambios. Pode facerse de varias formas:

- Editando directamente o arquivo `/etc/iptables/rules.v4`:
  - Hai que introducir as novas regras na súa posición, respectando non escribir debaixo de COMMIT.
  - Unha vez modificado o `/etc/iptables/rules.v4` hai que reiniciar o servizo con **`$ sudo systemctl restart netfilter-persistent.service`** ou **`$ sudo netfilter-persistent restart`**
  - No caso de traballar con IPv6 sería semellante pero modificando o arquivo `/etc/iptables/rules.v6`
- Executando directamente comandos iptables:
  - Executaríanse os comandos iptables correspondentes para crear as novas regras.
  - Gárdanse os cambios no ruleset mediante: **`$ sudo sh -c "iptables-save > /etc/iptables/rules.v4"`** ou **`sudo netfilter-persistent save.`** Desta forma, a próxima vez que se reinicie o sistema (ou o servizo con **`$ sudo netfilter-persistent restart`**) cargaranse as regras gardadas no arquivo **`/etc/iptables/rules.v4`.**
  - No caso de traballar con IPv6 sería semellante pero co comando **`ip6tables-save`** e o arquivo `/etc/iptables/rules.v6`

A modo de exemplo crease unha nova regra:

```
manuel@server:~$ sudo iptables -A INPUT -p tcp --dport 33000 -m conntrack --ctstate NEW -j ACCEPT
```

Pode comprobarse que no arquivo `/etc/iptables/rules.v4` non aparece a entrada correspondente:

```
manuel@server:~$ sudo cat /etc/iptables/rules.v4
# Generated by iptables-save v1.8.4 on Tue Feb 22 09:16:53 2022
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [1:66]
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -o lo -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -d 8.8.8.8/32 -p udp -m udp --dport 53 -m conntrack --ctstate NEW -j ACCEPT
```

```
-A OUTPUT -d 8.8.4.4/32 -p udp -m udp --dport 53 -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate NEW -j ACCEPT
COMMIT
# Completed on Tue Feb 22 09:16:53 2022
```

Para actualizar `/etc/iptables/rules.v4` execútase:

```
manuel@server:~$ sudo sh -c "iptables-save > /etc/iptables/rules.v4"
manuel@server:~$ sudo cat /etc/iptables/rules.v4
# Generated by iptables-save v1.8.4 on Tue Feb 22 09:23:25 2022
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -p tcp -m tcp --dport 33000 -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -o lo -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -d 8.8.8.8/32 -p udp -m udp --dport 53 -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -d 8.8.4.4/32 -p udp -m udp --dport 53 -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate NEW -j ACCEPT
COMMIT
# Completed on Tue Feb 22 09:23:25 2022
```

Agora, a próxima vez que se reinicie o sistema (ou `iptables-persistent`) cargarase a nova regra.

Policy DROP

Como se indicou no apartado de Cadeas (*chains*) e ó falar das CleanUp rules nos apuntamentos:

- Un paquete vai pasando por cada regra dunha cadea ata que casa cunha regra, momento no que procédese a executar a acción indicada na regra.
- Se non casa con ningunha regra, chega ata a Policy ou política da cadea, que ven sendo a acción por defecto a executar.
- A política máis segura a usar nun firewall é a de denegar por defecto, onde é permitido unicamente o tráfico requirido polas necesidades da rede e rexeitase o resto.

Estos puntos levan dun xeito natural a crear as pertinentes regras para autorizar o tráfico desexado e a aplicar unha política DROP nas cadeas a través dos comandos:

```
$sudo iptables -P INPUT DROP
$sudo iptables -P OUTPUT DROP
$sudo iptables -P FORWARD DROP
```

Todo correcto, pero ó aplicar unha política DROP poden xurdir '*problemas'* inesperados.

Problemática

Hai que entender que con Policy DROP bloqueamos todo o tráfico, agás o explícitamente autorizado; e tamén hai que darse conta que, en canto escribimos un comando iptables e prememos `Enter`, a orde execútase inmediatamente. Netfilter/iptables non é como outros firewalls onde hai que gardar os cambios e despois aplicalos; aquí, a execución é inmediata.

Esto é especialmente problemático cando traballamos cunha máquina remota e polo motivo que sexa, temos que facer unha limpeza das regras do firewall. A modo de exemplo, no noso ruleset o acceso por ssh ó equipo remoto está asegurado grazas a regra 3 da cadea INPUT (iniciar conexións ssh) e as regras 1 das cadeas INPUT e OUTPUT (manter as conexións). Supoñamos que queremos borrar as regras e procedemos a facer un `iptables -F`:

```
-F, --flush [chain]
Flush the selected chain (all the chains in the table if none is given). This is equivalent to deleting all the rules one by one.
```

```
manuel@server:~$ sudo iptables -F
```

O ruleset resultante é:

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

Limpamos as regras pero o ruleset resultante fai que perdamos a conexión por ssh ó server (fixarse que xa non hai regras ESTABLISHED). Non só perdemos a conexión ssh actual, tamén bloqueamos totalmente o acceso ó equipo!!!! Habería que acceder físicamente ó server para solventar o problema.

Solucións

O que está claro é que sempre, sempre, sempre, hai que ser consciente dos cambios a executar e as súas consecuencias. A política DROP non é un problema se facemos as cousas ben; sen embargo, podemos adoptar certas medidas sinxelas que nos protexan de despistes:

Protección#1

Limpar as regras do ruleset empregando a opción flush de `netfilter-persistent` en vez de con `iptables -F`. Conseguimos limpar as regras e ademais as políticas de filtrado configúranse a ACCEPT, polo que temos garantido o acceso:

```
manuel@server:~$ 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            ctstate NEW
3    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate NEW
4    ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:22 ctstate NEW
5    ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:33000 ctstate NEW

Chain FORWARD (policy DROP)
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            ctstate NEW
3    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            ctstate NEW
4    ACCEPT     udp  --  0.0.0.0/0            8.8.8.8              udp dpt:53 ctstate NEW
5    ACCEPT     udp  --  0.0.0.0/0            8.8.4.4              udp dpt:53 ctstate NEW
6    ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            multiport dports 80,443 ctstate NEW
manuel@server:~$ sudo netfilter-persistent flush
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables flush
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables flush
manuel@server:~$ 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
manuel@server:~$
```

Protección#2:

Neste caso non definimos a Policy a DROP senon que:

1. Definimos a Policy da táboa filter a ACCEPT.

2. Creamos nas cadeas unha última regra de CleanUP, onde explícitamente procedemos a eliminar todo o tráfico.

Deste xeito, a política de tráfico do firewall segue sendo denegar todo por defecto (última regra); pero se borramos as regras con `iptables -F`, como a Policy está a ACCEPT non quedamos bloqueados. A continuación podemos ver como se crean as regras finais de CleanUp pero conservando a Policy a ACCEPT e o que sucede ó facer un `iptables -F`:

```
manuel@server:~$ sudo iptables -P INPUT ACCEPT
manuel@server:~$ sudo iptables -P OUTPUT ACCEPT
manuel@server:~$ sudo iptables -P FORWARD ACCEPT
manuel@server:~$ sudo iptables -L -n –line-numbers
manuel@server:~$ sudo iptables -A INPUT -j DROP
manuel@server:~$ sudo iptables -A OUTPUT -j DROP
manuel@server:~$ sudo iptables -A FORWARD -j DROP
CChain 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           ctstate NEW
3    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           ctstate NEW
4    ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:22 ctstate NEW
5    ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:33000 ctstate NEW
6    DROP       all  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       all  --  0.0.0.0/0            0.0.0.0/0

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           ctstate NEW
3    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           ctstate NEW
4    ACCEPT     udp  --  0.0.0.0/0            8.8.8.8             udp dpt:53 ctstate NEW
5    ACCEPT     udp  --  0.0.0.0/0            8.8.4.4             udp dpt:53 ctstate NEW
6    ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           multiport dports 80,443 ctstate NEW
7    DROP       all  --  0.0.0.0/0            0.0.0.0/0
manuel@server:~$ sudo iptables -F
manuel@server:~$ 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
manuel@server:~$
```