Servei Nftables

"Netfilter" és el sistema tallafocs integrat a Linux. Per gestionar-lo ens podem trobar amb diferents eines, com ara "Ufw" (a Ubuntu) o "Firewalld" (a Fedora), però n'hi ha una de genèrica que serveix per totes les distribucions anomenada "Nftables". Per tant, és la que farem servir, així (les comandes següents són executades com administrador):

A Ubuntu: `systemctl --now disable ufw && apt install nftables`

A Fedora: `systemctl --now disable firewalld && dnf install nftables`

El fitxer de configuració on indicarem les regles és **/etc/nftables.conf** (a Ubuntu) o **/etc/sysconfig/nftables.conf** (a Fedora). Un cop l'haguem editat convenientment (veure següents diapositives), executarem la següent comanda per posar en marxa el tallafocs:

```
systemctl --now enable nftables
```

Configuració Nftables (I)

Nftables incorpora el concepte de "taula" per encabir tot el tràfic que serà gestionat per ell. Ens podem trobar, per exemple, amb taules de tipus **"ip"** (dins de les quals es gestiona el tràfic de tipus IPv4), taules de tipus **"ip6"** (dins de les quals es gestiona el tràfic de tipus IPv6), taules de tipus **"arp"** (dins de les quals es gestiona el tràfic homònim), etc.

Nosaltres la majoria de cops només necessitarem tenir definida una taula que serà de tipus **"inet"** (tipus que permet gestionar indistintament tràfic IPv4 i IPv6). Aquest taula pot tenir el nom que vulguem, però per raons històriques (herència de l'antiga eina *"iptables"*), és habitual anomenar-la **"filter"**.

Per tant, en resum, el que hem de fer per definir una taula de tipus *inet* anomenada *filter* hem d'escriure el següent dins de l'arxiu de configuració del servei Nftables:

```
table inet filter {
}
```

Configuració Nftables (I)

Dins de la taula *filter*, però, cal diferenciar entre el tràfic que entra al sistema, el que en surt i el que travessa (és a dir, el que hi arriba però ha de sortir-ne rebotat a un altre destí). Això s'aconsegueix amb el concepte de "cadena", una per cadascuan d'aquestes tres categories de tràfic. Serà dins de cada cadena en particular on podrem incloure les regles que necessitem especificar.

Per definir una cadena, a més del seu nom (que per tradició se solen anomenar, respectivament, **"input"**, **"output"** i **"forward"**, cal indicar també el seu tipus (totes tres en aquest cas són de tipus **"filter"** ; això ve donat pel tipus de taula on hi són presents), el seu "hook" (que és, respectivament, **"input"**, **"output"** i **"forward"** ja que indiquen efectivament a quin sentit del tràfic es refereix la cadena en qüestió) i la seva "prioritat" (que és un concepte avançat on no hi entrarem i, per tant, assignarem el seu valor estàndard, que és **0**). A més, cal definir una política per defecte (és a dir, decidir quina acció es realitzarà sobre cada paquet que travessi la cadena en qüestió si no ha concordat amb cap de les regles allà indicades; les accions més habituals són **"accept"** -es deixa passar- i **"drop"** -no es deixa passar-).

Per tant, en resum, el que hem de fer per definir les tres cadenes típiques dins de la taula de tipus *inet* anomenada *filter* és escriure el següent dins de l'arxiu de configuració del servei Nftables:

```
table inet filter {
 chain input { type filter hook input priority 0; policy accept;
   #Aquí s'escriuran les regles que afectin al tràfic d'entrada
 }
 chain output { type filter hook output priority 0; policy accept;
   #Aquí s'escriuran les regles que afectin al tràfic de sortida
 }
 chain forward { type filter hook forward priority 0;policy accept;
   #Aquí s'escriuran les regles que afectin al tràfic de rebot
 }
}
```

Elements i aplicació d'una regla

Les regles a indicar dins d'una cadena estan formades per dos elements:

- Un conjunt de característiques que es comprovaran en cada paquet que travessi la cadena (com per exemple, la seva IP d'origen, o el seu port de destí, etc). Si el paquet en qüestió conté totes i cadascuna de les caracerístiques indicades, llavors s'aplicarà...
- Una determinada acció sobre ell (per acceptar-lo o descartar-lo, etc)

En el cas que el paquet inspeccionat tingui unes característiques que no concordin amb totes i cadascuna de les característiques indicades a la regla, aquesta no s'aplicarà, ni per bé ni per mal. Això vol dir que es passarà a inspeccionar la regla següent de la cadena en qüestió (l'ordre de les regles és, per tant, molt important!; en aquest sentit, es recomana afegir primer les regles més específiques i d'allà anar "baixant" cap a regles més genèriques). Si després d'inspeccionar totes les regles presents, el paquet estudiat no concordés amb cap, llavors se li aplicarà la política per defecte definida a la cadena (la qual, recordem, pot ser *accept* o *drop*).

Característiques a inspeccionar d'un paquet

ether saddr <dirMAC> : MAC d'origen igual al valor indicat
ether daddr <dirMAC> : MAC de destí igual al valor indicat

ip saddr <dirIP> : IP d'origen (de host o de xarxa/mask) igual al valor indicat
ip saddr != <dirIP> : IP d'origen diferent del valor indicat
ip saddr <dirIP1>-<dirIP2> : IPs d'origen dins del rang indicat (ambdós inclosos)
ip saddr != <dirIP1>-<dirIP2> : IPs d'origen fora del rang indicat (ambdós inclosos)
ip saddr {<dirIP1>, <dirIP2,...} : IPs d'origen dins del conjunt indicat
ip saddr != {<dirIP1>, <dirIP2,...} : IPs d'origen fora del conjunt indicat

ip daddr <dirIP> : IP de destí (de host o de xarxa/mask) igual al valor indicat
ip daddr != <dirIP> : IP de destí diferent del valor indicat
ip daddr <dirIP1>-<dirIP2> : IPs de destí dins del rang indicat (ambdós inclosos)
ip daddr != <dirIP1>-<dirIP2> : IPs de destí fora del rang indicat (ambdós inclosos)
ip daddr {<dirIP1>, <dirIP2,...} : IPs de destí dins del conjunt indicat
ip daddr != {<dirIP1>, <dirIP2,...} : IPs de destí fora del conjunt indicat

udp sport <n> : Port d'origen UDP igual al valor indicat
udp sport != <n> : Port d'origen UDP diferent al valor indicat
udp sport <n1>-<n2> : Ports d'origen UDP dins del rang indicat (ambdós inclosos)
udp sport != <n1>-<n2> : Ports d'origen UDP fora del rang indicat (ambdós inclosos)
udp sport {<n1>, <n2>...} : Ports d'origen UDP dins del conjunt indicat
udp sport != {<n1>, <n2>...} : Ports d'origen UDP fora del conjunt indicat

udp dport <n> : Port de destí UDP igual al valor indicat

udp dport != <n> : Port de destí UDP diferent al valor indicat

udp dport <n1>-<n2> : Ports de destí UDP dins del rang indicat (ambdós inclosos)

udp dport != <n1>-<n2> : Ports de destí UDP fora del rang indicat (ambdós inclosos)

udp dport {<n1>, <n2>...} : Ports de destí UDP dins del conjunt indicat

udp dport != {<n1>, <n2>...} : Ports de destí UDP fora del conjunt indicat

tcp sport <n> : Port d'origen TCP igual al valor indicat

tcp sport != <n> : Port d'origen TCP diferent al valor indicat

tcp sport <n1>-<n2> : Ports d'origen TCP dins del rang indicat (ambdós inclosos)

tcp sport != <n1>-<n2> : Ports d'origen TCP fora del rang indicat (ambdós inclosos)

tcp sport {<n1>, <n2>...} : Ports d'origen TCP dins del conjunt indicat

tcp sport != {<n1>, <n2>...} : Ports d'origen TCP fora del conjunt indicat

tcp dport <n> : Port de destí TCP igual al valor indicat

tcp dport != <n> : Port de destí TCP diferent al valor indicat

tcp dport <n1>-<n2> : Ports de destí TCP dins del rang indicat (ambdós inclosos)

tcp dport != <n1>-<n2> : Ports de destí TCP fora del rang indicat (ambdós inclosos)

tcp dport {<n1>, <n2>...} : Ports de destí TCP dins del conjunt indicat

tcp dport != {<n1>, <n2>...} : Ports de destí TCP fora del conjunt indicat

tcp flags <nom> : Flag (**"syn","ack","fin","rst"**,...) indicada activada

tcp flags != <nom> : Flag indicada desactivada

tcp flags {<nom1>, <nom2>...} : Flags dins del conjunt indicat estan activades

tcp flags != {<nom1>, <nom2>...} : Flags fora del conjunt indicat estan activades

icmp type <tipus> : Tipus de paquet ICMP igual al valor indicat. Alguns dels valors possibles són: **"echo-request"**, **"echo-reply"**, **"destination-unreachable"**, **"time-exceeded"**,...

ct state <estat1>,<estat2>,... : L'estat de la connexió a la qual pertany el paquet és igual a l'indicat. Els valors possibles són: **"new"** (el paquet forma part d'un inici de nova connexió), **"established"** (el paquet pertany a una connexió ja establerta), **"related"** (el paquet es correspon a un inici de nova connexió però que està relacionada amb alguna altra connexió prèviament existent; això és habitual en transferències FTP o errors ICMP) i **"invalid"** (el paquet té una combinació "flags" TCP que no és admissible -com ara tots els "flags" a 1 ("Xmas attack"), tots a 0 ("Null attack"), SYN+RST (no admesa per cap transacció), SYN+FIN (tampoc admesa), etc)

iifname <nomTarja> : El paquet inspeccionat entra a la tarja de xarxa indicada

oifname <nomTarja> : El paquet inspeccionat surt de la tarja de xarxa indicada

limit rate <n/t> : Es detecta una ràtio per sota de nº paquets/temps, on temps es pot indicar amb les paraules "second", "minute", "hour", "day" o "week". També existeix **limit rate over <n/t>** , que detecta una ràtio per sobre de nº paquets/temps

limit rate <n b/t> : Es detecta una ràtio per sota de nº bytes/temps, on el valor "b" s'ha d'indicar amb les paraules "bytes", "kbytes", "mbytes" o "gbytes". També existeix **limit rate over <n b/t>** , que detecta una ràtio per sobre de nº bytes/temps

Accions a realitzar sobre un paquet

accept : El paquet s'envia amb èxit a l'aplicació destí, i s'atura el seu processament

drop : El paquet és rebutjat (sense notificar-ho al remitent), i s'atura el seu processament

reject : Igual que *drop*, però envia un missatge ICMP/ICMPv6 "port unreachable" al remitent informant del rebuig

log : S'envia una notificació relativa a la detecció del paquet inspeccionat al registre del sistema en forma de missatge de l'estil "IN=xxx OUT=xxx" (si el registre és de tipus Systemd, aquest registre es pot gestionar mitjançant la comanda *journalctl -k*) . El processament del paquet continuarà sense alteracions en espera de trobar una altra acció com *accept*, *drop* o *reject*.

counter: Indica que s'utilitzarà un comptador de paquets i bytes per la regla en qüestió. Per veure els valors en un moment donat dels comptadors definits a les regles d'una taula concreta, es pot executar la comanda `sudo nft -a list table <taula>`

Afegir regles

Hi ha dues formes d'afegir regles dins de les cadenes existents:

- Escrivint-les manualment dins de l'arxiu de configuració. Fent-ho així, les regles s'aplicaran a partir del següent reinici del servei Nftables (i en tots els reinicis posteriors) ja que formaran part de la seva configuració permanent
- Executant alguna comanda *nft add rule ...* com alguna de les següents que es posen d'exemple. Aquesta comanda aplica la regla en qüestió immediatament, però de forma efímera: si el servei Nftables (o el sistema sencer) es reinicia, les regles afegides d'aquesta forma es perdran.

`nft add rule inet filter input iifname enp0s3 drop` : Afegeix una regla (dins de la cadena "input", per tant referent al tràfic d'entrada) que descarta tot el tràfic entrant per la tarja 'enp0s3'

`nft add rule inet filter output ip daddr 1.1.1.1 drop` : Afegeix una regla (dins de la cadena "output", per tant referent al tràfic de sortida) que descarta tot el tràfic dirigit a la IP indicada

Llistar regles

Per saber en un moment determinat quines són les regles actives (ja que n'hi podrien haver que no estiguessin escrites al fitxer "nftables.conf" degut a haver utilitzat la comanda *nft add rule...*, es pot executar la comanda següent:

```
nft list ruleset
```

També la podem executar així...:

```
nft -a list ruleset
```

...per fer que es mostrin, a més, els "handlers" associats a cada cadena i regla definida. Un "handler" és un identificador numèric que el podem fer servir per referenciar la cadena/regla en determinats moments, com per exemple, a l'hora de voler esborrar-les, com veurem a la següent diapositiva.

Eliminar regles

Per esborrar una determinada regla actualment carregada, primer hem d'esbrinar el seu "handler" tal com acabem de veure, i un cop conegut, executar una comanda ^nft delete rule...* tal com la següent, on s'ha d'especificar tota "la ruta": tipus de taula->nom de taula->nom de cadena i, un cop allà, indicar el número de "handler" de la regla que es vol eliminar:

```
nft delete rule inet filter input handler <n>
```

Si es vol eliminar tota l'estructura de regles i cadenes i taules actualment carregades (per fer un "reset" de la configuració des de 0), es pot executar la comanda següent:

```
nft flush ruleset
```

Afegir regles permanentment

Una forma de treballar habitual és, després d'escriure l'estructura bàsica de taules i cadenes buides dins el fitxer de configuració, provar "in situ" les regles mitjançant les comandes *nft add rule* i, si cal *nft delete rule* fins tenir totes les regles desitjades aplicades en RAM. Un cop arribat aquest moment, en lloc d'haver de reescriure les regles ja funcionals en el fitxer de configuració per a què romanguin de forma permanent, el que se sol fer és simplement executar la següent comanda, la qual aprofita la configuració actualment activa del tallafocs per sobreescriure amb ella el contingut del seu fitxer de configuració:

A Ubuntu: `nft list ruleset > /etc/nftables.conf`

A Fedora: `nft list ruleset > /etc/sysconfig/nftables.conf`

Exemples de regles

- Bloquejar (només) el tràfic que provingui d'una IP concreta (també es podria bloquejar a la xarxa sencera especificant 192.168.1.0/24):

```
sudo nft add rule inet filter input ip saddr 192.168.1.234 drop
```

- Bloquejar (només) tot el tràfic dirigit a dues IPs concretes, establint comptadors:

```
sudo nft add rule inet filter output ip daddr "{8.8.8.8, 8.8.4.4}" counter drop
```

- Bloquejar (només) el tràfic ICMP entrant que sigui de tipus "echo-request":

```
sudo nft add rule inet filter input icmp type echo-request drop
```

- Bloquejar (només) el tràfic que entri per la tarja de xarxa enp0s3 (provinent de qualsevol lloc) i que a més vagi dirigit a certs ports concrets del nostre sistema (en aquest cas, els ports 22 i 80 TCP):

```
sudo nft add rule inet filter input iifname enp0s3 tcp dport {22, 80} drop
```

- Bloquejar (només) el tràfic que entri per la tarja de xarxa enp0s3 (provinent de qualsevol lloc) i que a més vagi dirigit a un port determinat del nostre sistema (en aquest cas, el port 22 TCP) i que, a més, guardi un registre sobre aquest fet:

```
sudo nft add rule inet filter input iifname enp0s3 tcp dport 22 log prefix "Accés al port 22 detectat" drop
```

- Bloquejar (només) el tràfic que entri per la tarja de xarxa enp0s3 que a més provingui d'una IP concreta i que a més vagi dirigit a un port determinat del nostre sistema (en aquest cas, el port 22 TCP):

```
sudo nft add rule inet filter input iifname enp0s3 ip saddr 192.168.1.234 tcp dport 22 drop
```

- Acceptar tots els paquets que provinguin de l'exterior però només pertanyents a connexions prèviament obertes des del nostre sistema (aquesta regla és útil -si la política per defecte és drop- per evitar inicis de connexió generats des de l'exterior ja que només accepta els paquets d'entrada corresponents a les respostes de les connexions obertes des del sistema local):

  ```
  sudo nft add rule inet filter input ct state established, related accept
  ```