ACTIVITAT FORMATIVA	MP: SMX-6 - Seguretat informática
	UF5: **Tallafocs i monitoratge de xarxes**

Netfilter (1)

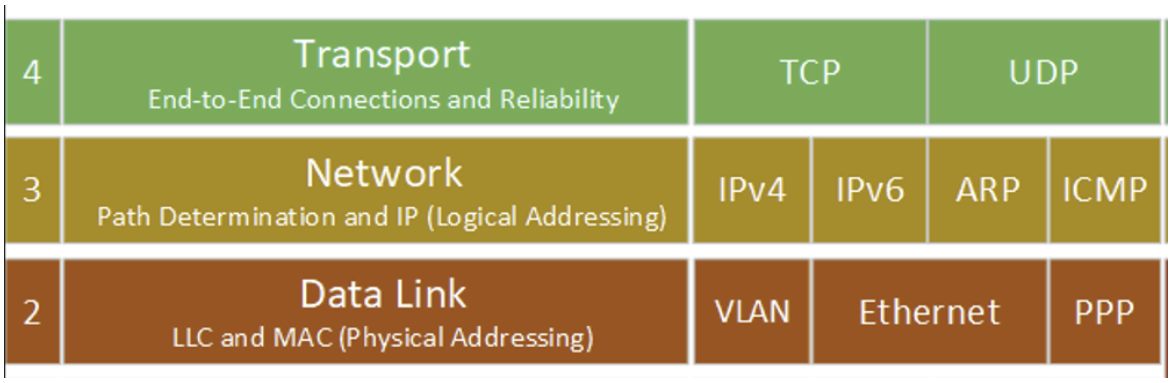
SMX-6-UF5-7 - Netfilter (1) (Privat)

Introducció

Netfilter forma part del nucli de Linux i permet manipular els diferents paquets de xarxa i el seu flux de processament.

Netfilter proporciona diferents "hooks" (punt d'anclatge) en el flux de processament que et permet registrar funcions per aplicar als paquets de xarxa en aquell punt concret de processament.

Netfilter permet filtrar en múltiples nivells de xarxa:



ntables

nftables forma part del projecte netfilter i és la utilitat que et permet interactuar amb nefilter mitjaçant l'eina `nft` .

Entorn de treball

Crea tres màquins virtual `filter`, `alpha` i `beta`.

Tables

Una taula a nftables és un espai de noms que conté una col·lecció de chains, sets, maps, flowtables, i stateful objects.

Family

Cada taula ha de tenir assignat una única family d'adreces que defineix els tipus de paquets que processa aquesta taula.

Quan crees una taula pots escollir entre aquestes famílies d'adreces:

- `ip` - La taula filtra paquets IPv4.
- `ip6` - La taula filtra paquets IPv6.
- `inet` - La taula filtra paquets IPv4 i IPv6, i pots utilitzar una única regla pels dos tipus de paquets en el que tenen en comú, per exemple el port.
- `arp` - La taula filtra paquets ARP a nivell 2, abans que el kernel faci qualsevol gestió a nivell 3.
- `bridge` - Un pont conecta dos segments Ethernet i els paquets es reenvien en funció de l'adreça Ethernet. Com que el reenviament es fa a la capa 2, aquest encaminament és transparent als protocols de capa superior com pot ser el protocol IP.
- `netdev` - La taula s'enllaça directament a una única interficie de xarxa enlloc de un protocol.
- `ingress` - Fa el mateix que netdev, però més fàcil de configurar (des del kernel 5.10).

N**o hi ha taules predeterminades per a nftables**.

Per tant, l'ordre `list tables` no pot tornar res si no has definit cap taula.

```
box@filter:~$ sudo nft list tables
box@filter:~$
```

UFW

A SMX-6-UF5-6 - UFW vas veure que per defecte UFW està deshabilitat.

Si habilites el firewall UFW pots verificar que s'han creat dos taules amb el nom `filter` :

```
box@filter:~$ sudo ufw allow 22
Rules updated
Rules updated (v6)
box@filter:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation
Firewall is active and enabled on system startup
box@filter:~$
box@filter:~$ sudo nft list tables
table ip filter
table ip6 filter
box@filter:~$
```

Encara que les dos taules tenen el mateix nom pertanyen a famílies diferents: `ip` i `ip6`.

En la taula `filter` està la regla que permet conexions entrants (`dport`) al port 22 :

```
box@filter:~$ sudo nft list table filter | grep "dport 22"
                meta l4proto tcp tcp dport 22 counter packets 0 bytes 0 accept
                meta l4proto udp udp dport 22 counter packets 0 bytes 0 accept
box@filter:~$
```

Deshabilita el firewall i elimina les taules `filter` :

```
box@filter:~$ sudo ufw disable
Firewall stopped and disabled on system startup
box@filter:~$
box@filter:~$ sudo nft delete table filter
box@filter:~$
box@filter:~$ sudo nft list tables
table ip6 filter
box@filter:~$
```

Com que per defecte la família és **`ip`** si no indiques altre cosa, només s'ha borrat la taula "`ip filter`",

Borra la taula "**`ip6 filter`**":

```
box@filter:~$ sudo nft delete table ip6 filter
box@filter:~$
box@filter:~$ sudo nft list tables
box@filter:~$
```

filter

Quan crees una taula li pots posar qualsevol nom, però per defecte és costum que el nom sigui `filter` com fa UFW.

Crea una taula nova (per defecte la família d'adreces és ip) :

```
box@filter:~$ sudo nft add table filter
box@filter:~$
```

Tal com hem vist abans, un cop s'ha afegit la taula l'ordre list tables retorna el nom de la taula:

```
box@filter:~$ sudo nft list tables
table ip filter
box@filter:~$
```

Pots obtenir més informació sobre la taula amb aquesta ordre:

```
box@filter:~$ sudo nft list table filter
table ip filter {
}
box@filter:~$
```

En fer-ho, es mostra informació sobre la taula, incloses les cadenes definides a la taula.

Com mostra l'exemple, la **taula de filtres** utilitza la **família IP** i actualment està buida.

Chains

Com has vist a SMX-6-UF5-6 - UFW, **un firewall és un conjunt de regles que es processen per decidir que és fa amb un paquet.**

Però les regles no s'afegeixen directament a les taules sinó a les cadenes.

Una cadena s'enllaça a un dels hooks que proporciona netfilter per tal de poder filtrar els paquets de xarxes en punts diferents del seu processament.

hooks

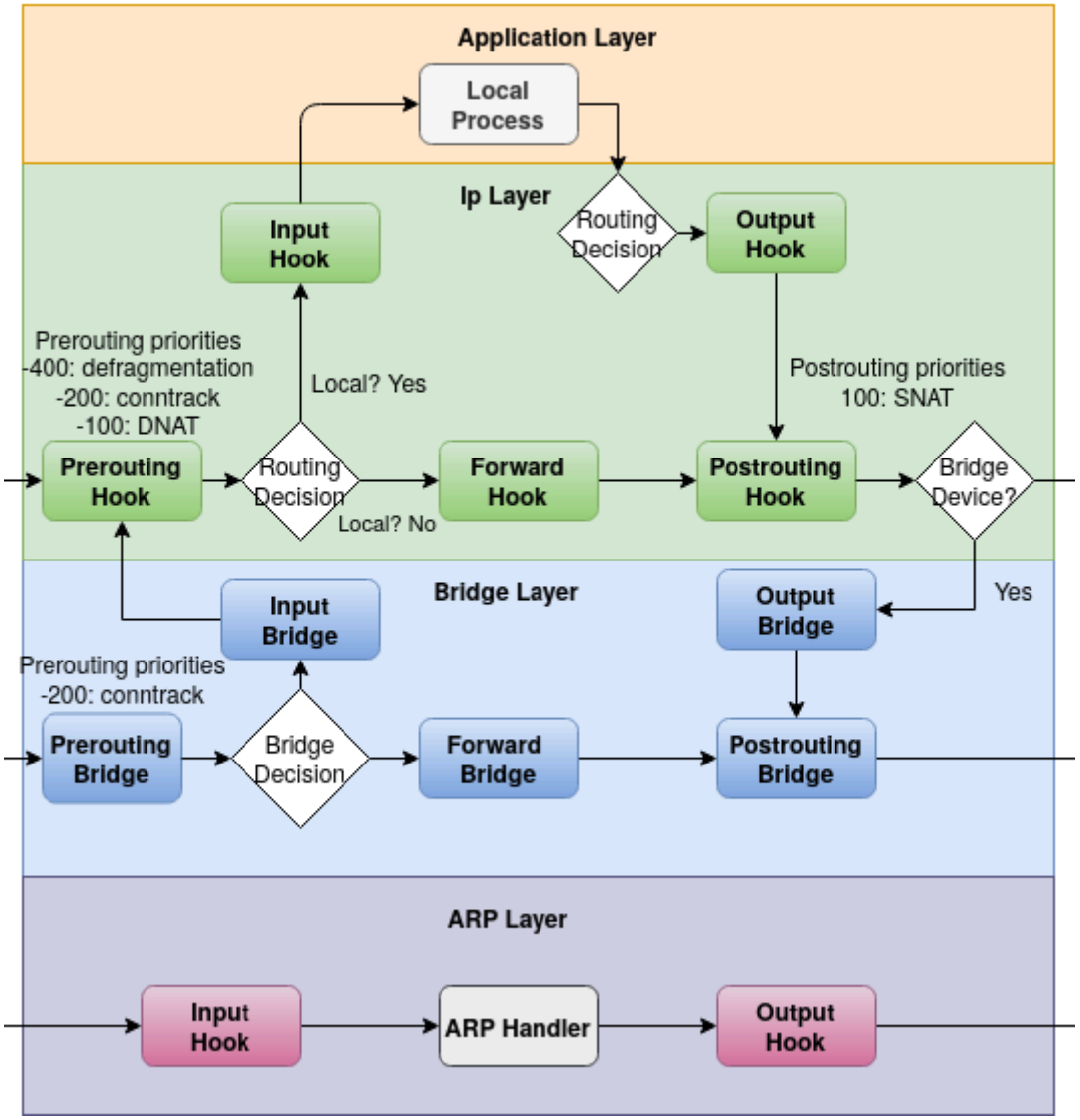
Els "hooks" que pots utilitzar al configurar una cadena base, i que et permeten manipular els paquets en un punt concret, són aquests:

- **ingress** – el paquet acaba se ser processat per la controladora de la NIC
- **prerouting** –el paquet no ha estat acceptat pel "flowtable" i està pendent de ser enrutat (a una altre host) o no.
- **input** – el paquet es processarà al mateix host i està pendent de ser processat pel procés corresponent.
- **forward** – el paquet no es processarà al host

- **output** – el paquet s'ha originat per un procés del host.
- **postrouting** – el paquet està a punt de sortir del host.

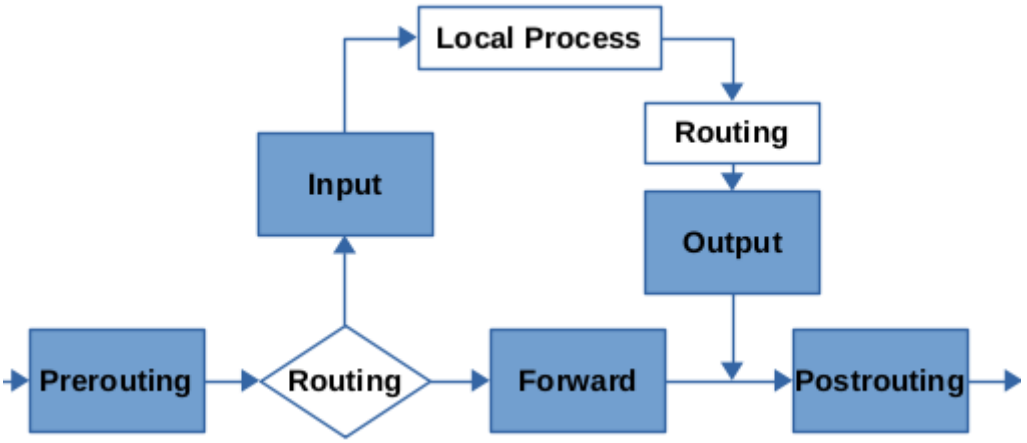
Però no tots els paquets passen per tots aquests hooks.

No és el mateix processar un paquet `ARP` que un paquet `IP`.



Un paquet **ARP** només passa pels "**hooks**" `input` i `output` (a més de l'`ingress`), no hi ha "routing".

En canvi, un paquet **IP** passa per tots els "hooks":



base chain

Una cadena base s'ha d'enllaçar a alguna d'aqueste fases (o "hooks").

La sintaxi és aquesta:

```
nft add chain [<family>] <table_name> <chain_name> {
    type <type> hook <hook> priority <value> \;
    [policy <policy> \;]
    [comment \"text comment\" \;]
}
```

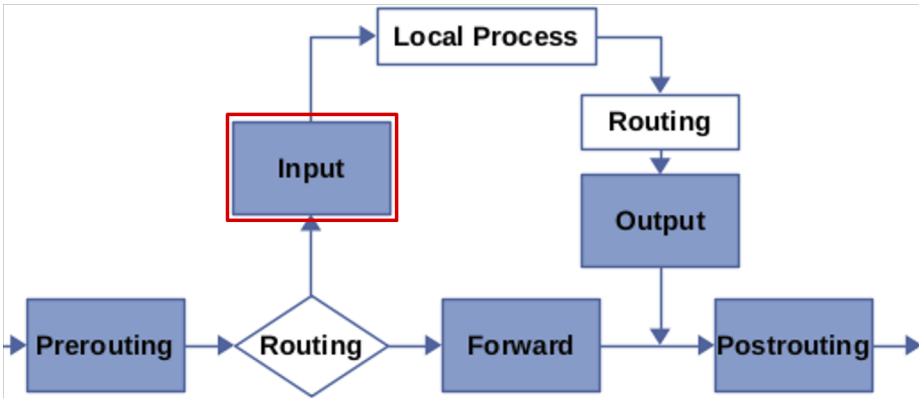
input hook

Afegeix una cadena `input` a la taula `filter` :

```
box@filter:~$ sudo nft add chain ip filter input \
> '{ type filter hook input priority 0; }'
box@filter:~$
box@filter:~$ sudo nft list table filter
table ip filter {
        chain input {
                type filter hook input priority filter; policy accept;
        }
}
box@filter:~$
```

Aquesta ordre crea un cadena "input" que s'enllaça al hook input.

Per tant veurà tots els paquets IPv4 que no es "routejen" a una altre adreça.



default policy

Totes les cadenes "base" tenen una política per defecte que es pot especificar al crear la cadena.

Aquesta és la política que s'aplicarà a un paquet si no hi ha cap regla en la cadena que decideixi que fer amb un paquet.

Si no dius res, per defecte la cadena es crea amb `policy accept`.

```
box@filter:~$ sudo nft list table filter
table ip filter {
        chain input {
                type filter hook input priority filter; policy accept;
        }
}
box@filter:~$
```

Només pots escollir entre dues polítiques predeterminades:

1. **accept** – El paquet seguirà el seu camí.
2. **drop**. – El paquet serà eliminat.

Pots canviar la política per defecte d'una cadena:

```
box@filter:~$ sudo nft add chain filter input \
> ' { policy drop; } '
box@filter:~$ client_loop: send disconnect: Connection reset
PS C:\Users\david>
```

Si executes aquesta comanda, nftables no permetrà **cap paquet d'entrada i perds la connexió a la màquina virtual.**

Per sort, aquesta configuració només funciona mentres la màquina està funcionat.

Apaga la màquina, arranca de nou i conecta't.

Verifica que no hi ha cap taula:

```
box@filter:~$ sudo nft list tables
box@filter:~$
```

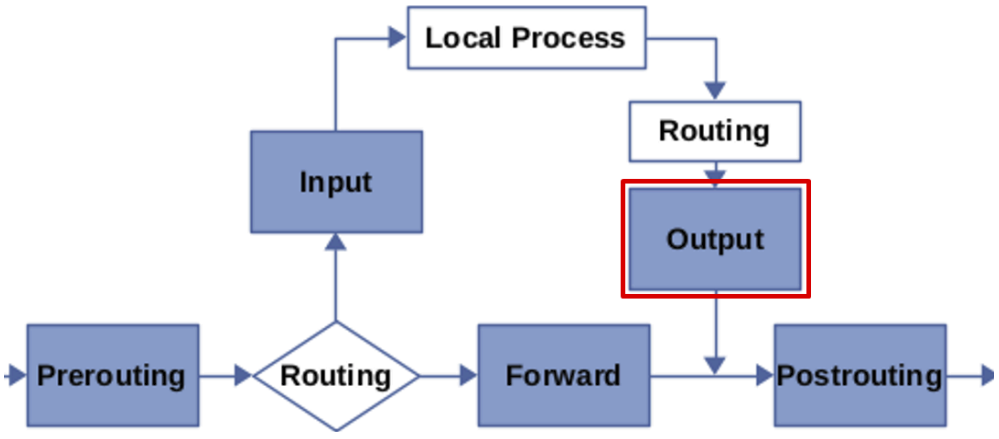
ouput hook

També pots registrar la cadena de sortida:

```
box@filter:~$ sudo nft 'add chain ip filter ouput \
{ type filter hook output priority 0; }'
box@filter:~$
box@filter:~$ sudo nft list table filter
table ip filter {
        chain input {
                type filter hook input priority filter; policy accept;
        }

        chain ouput {
                type filter hook output priority filter; policy accept;
        }
}
box@filter:~$
```

Output chain s'enganxa al hook output i veurà els paquets de sortida un cop s'han processat:



ingress hook

A diferència dels "hooks" anteriors, un ingress hook és específic d'un dispositiu de xarxa.

Has de crear una taula de la familia **`netdev`** :

```
box@filter:~$ sudo nft add table netdev filter
box@filter:~$
box@filter:~$ sudo nft list tables
table ip filter
table netdev filter
box@filter:~$
```

Com que la cadena és específica d'un dispositiu de xarxa, és obligatori especificar el dispositiu on s'enganxarà la cadena.

```
box@filter:~$ sudo nft 'add chain netdev filter enp0s8 \
{ type filter hook ingress device enp0s8 priority 0; }'
box@filter:~$
box@filter:~$ sudo nft list table netdev filter
table netdev filter {
        chain enp0s8 {
                type filter hook ingress device "enp0s8" priority
        }
}
box@filter:~$
```

type

Totes les cadenes que hem creat són de tipus `filter` :

```
box@filter:~$ sudo nft list chain filter input
table ip filter {
        chain input {
                type filter hook input priority filter; policy accept;
        }
}
box@filter:~$
```

Aquest tipus es fa servir per filtrar paquets de les familes `arp`, `bridge`, `ip`, `ip6` o `inet`.

L'altre tipus és route que es fa servir per redirigir els paquets en base a un camp de la capçalera IP o si es modifica la marca del paquet.

El tipus `route` només es pot fer servir amb les famílies `ip`, `ip6` o `inet`

priority

Si defineixes més d'una cadena pel mateix "hook" el valor `priority` indica **quina s'utilitzarà primer.**

Quan més petit sigui el número, la cadena s'executarà abans.

Afegeix una cadena **`input-ssh`** que anirà abans que **la cadena `input`** :

```
box@filter:~$ sudo nft add chain ip filter input-ssh \
> '{ type filter hook input priority -1; }'
box@filter:~$
box@filter:~$ sudo nft list table filter
table ip filter {
        chain input {
                type filter hook input priority filter; policy accept;
        }

        chain output {
                type filter hook output priority filter; policy accept;
        }

        chain input-ssh {
                type filter hook input priority filter - 1; policy accept;
        }
}
box@filter:~$
```

Borrar una cadena

Pots suprimir una cadena amb aquesta ordre:

```
nft delete chain [family] <table_name> <chain_name>
```

Crea una cadena `input-test` :

```
box@filter:~$ sudo nft add chain ip filter input-test \
> '{ type filter hook input priority 10; }'
box@filter:~$
box@filter:~$ sudo nft list chain filter input-test
table ip filter {
        chain input-test {
                type filter hook input priority filter + 10; policy accept;
        }
}
box@filter:~$
```

Borra la cadena que acabes de crear:

```
box@filter:~$ sudo nft delete chain filter input-test
box@filter:~$
box@filter:~$ sudo nft list chain filter input-test
Error: No such file or directory
list chain filter input-test
                  ^^^^^^^^^^
box@filter:~$
```

Rules

Les regles prenen mesures sobre els paquets de xarxa (p. ex. **acceptant-los o deixant-los caure**) en funció de si coincideixen amb els criteris especificats.

Cada regla consta de zero o més expressions seguides d'una o més declaracions:

```
<table> <chain> expressió-1 ... expressió-n declaració-1 ... declaració-n
```

Expressions

Cada expressió prova si un paquet coincideix amb un camp de càrrega útil específic o metadades de paquet/flux. Les expressions múltiples s'avaluen linealment d'esquerra a dreta: si la primera expressió coincideix, llavors s'avalua la següent expressió i així successivament.

Si arribem a l'expressió final, aleshores el paquet coincideix amb totes les expressions de la regla i s'executen les declaracions de la regla.

Declaracions

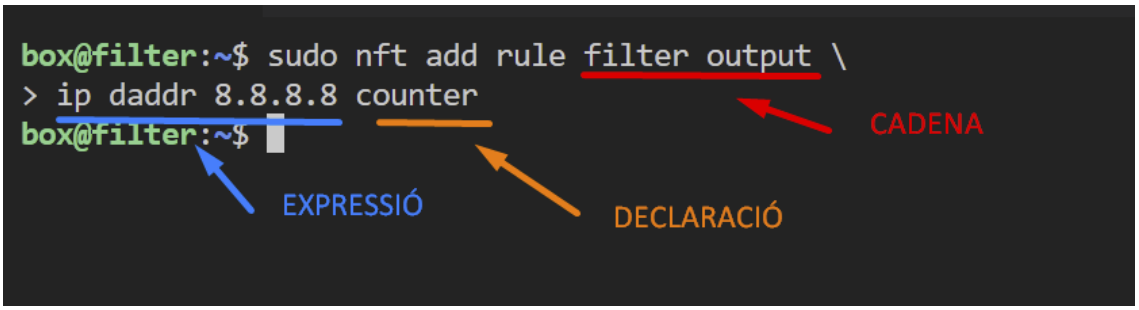
Cada declaració fa una acció, com ara establir la marca netfilter, comptar el paquet, registrar el paquet o emetre un veredicte com acceptar o deixar el paquet o saltar a una altra cadena.

Igual que amb les expressions, diverses declaracions s'avaluen linealment d'esquerra a dreta: una sola regla pot fer diverses accions utilitzant diverses sentències.

Tingues en compte que una declaració de veredicte per la seva naturalesa posa fi a la regla.

Afegir una regla

Per afegir una regla has d'especificar la taula i la cadena que vols utilitzar:



- `filter` és la taula i `output` és la cadena
- `ip daddr 8.8.8.8` és una expressió que ha de fet match: paquets IP que tenen com adreça destí `8.8.8.8` (`daddr` = *destination address*)
- `counter` és una declaració: va contant el número de bytes que han activat la regla.

Mira el contingut de la cadena `filter output` :

```
box@filter:~$ sudo nft list chain filter output
table ip filter {
        chain output {
                type filter hook output priority filter; policy a
                ip daddr 8.8.8.8 counter packets 0 bytes 0
        }
}
box@filter:~$
```

EL CONTADOR ESTÀ A 0

Quan enumeres les regles d'una cadena pots utilitzar modificadors per traduir adreces IP a noms DNS, protocols TCP, etc. : Output text modifiers

```
box@filter:~$ sudo nft -N list chain filter output
table ip filter {
        chain output {
                type filter hook output priority filter; policy ac
                ip daddr dns.google counter packets 0 bytes 0
        }
}
box@filter:~$
```

counter

Afegir un contador és una técnica molt útil per verificar si una regla funciona:

Fes un `ping` a `8.8.8.8` i verifica que la regla s'ha activat :

```
box@filter:~$ ping -c 1 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=116 time=12.8 ms

--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 12.762/12.762/12.762/0.000 ms
box@filter:~$
box@filter:~$ sudo nft list chain filter output
table ip filter {
        chain output {
                type filter hook output priority filter; policy accept;
                ip daddr 8.8.8.8 counter packets 1 bytes 84
        }
}
box@filter:~$
```

Un comptador compta tant el nombre total de paquets com el total de bytes que ha vist des de l'últim restabliment.

Fes un `ping` a `dns.google` i verifica que la regla s'ha activat :

```
box@filter:~$ ping -c 1 dns.google
PING dns.google (8.8.8.8) 56(84) bytes of data.
64 bytes from dns.google (8.8.8.8): icmp_seq=1 ttl=116 time=12.4 ms

--- dns.google ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 12.416/12.416/12.416/0.000 ms
box@filter:~$
box@filter:~$ sudo nft list chain filter output
table ip filter {
        chain output {
                type filter hook output priority filter; policy accept;
                ip daddr 8.8.8.8 counter packets 2 bytes 168
        }
}
box@filter:~$
```

Fes un `ping` a `1.1.1.1` i verifia que la regla no s'ha activat :

```
box@filter:~$ ping -c 1 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=55 time=14.0 ms

--- 1.1.1.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 13.964/13.964/13.964/0.000 ms
box@filter:~$
box@filter:~$ sudo nft list chain filter output
table ip filter {
        chain output {
                type filter hook output priority filter; policy accept;
                ip daddr 8.8.8.8 counter packets 2 bytes 168
        }
}
box@filter:~$
```

replace

Si no volem que la regla segueixi contant paquets podem reemplaçar la regla amb una sense contador.

Primer hem de saber quin és el "handle" de la regla amb l'opció `-a` :

```
box@filter:~$ sudo nft -a list chain filter output
table ip filter {
        chain output { # handle 7
                type filter hook output priority filter; policy accept;
                ip daddr 8.8.8.8 counter packets 2 bytes 168 # handle 8
        }
}
box@filter:~$
```

Ara podem reemplaçar la regla indicant el “handle” corresponent :

```
box@filter:~$ sudo nft -a list chain filter output
table ip filter {
        chain output { # handle 7
                type filter hook output priority filter; policy accept;
                ip daddr 8.8.8.8 counter packets 2 bytes 168 # handle 8
        }
}
box@filter:~$ sudo nft replace rule filter output handle 8 \
> ip daddr 8.8.8.8
box@filter:~$
box@filter:~$ sudo nft -a list chain filter output
table ip filter {
        chain output { # handle 7
                type filter hook output priority filter; policy accept;
                ip daddr 8.8.8.8 # handle 8
        }
}
box@filter:~$
```

Afegir més regles

Torna a modificar la regla afegint un contador i torna a afegir la mateixa regla una altre vegada:

Una cadena es composa de regles que es processen una rere l’altre.

Si fas ping a `dns.google` els dos contadors s’incrementen:

```
box@filter:~$ ping -c 1 8.8.8.8 > /dev/null
box@filter:~$ sudo nft -a list chain filter output
table ip filter {
        chain output { # handle 7
                type filter hook output priority filter; policy accept;
                ip daddr 8.8.8.8 counter packets 1 bytes 84 # handle 8
                ip daddr 8.8.8.8 counter packets 1 bytes 84 # handle 10
        }
}
box@filter:~$
```

Però les regles no estan només per contar paquets, sinó per acceptar o rebutjar paquets.

Modifica l'última regla perquè elimini els paquets:

```
box@filter:~$ sudo nft replace rule filter output handle 10 \
> ip daddr 8.8.8.8 counter drop
box@filter:~$
box@filter:~$ ping -c 1 8.8.8.8 > /dev/null
box@filter:~$
box@filter:~$ sudo nft -a list chain filter output
table ip filter {
        chain output { # handle 7
                type filter hook output priority filter; policy accept;
                ip daddr 8.8.8.8 counter packets 2 bytes 168 # handle 8
                ip daddr 8.8.8.8 counter packets 1 bytes 84 drop # handle 10
        }
}
box@filter:~$
```

Com que les regles es processen de dalt a baix, l'última regla regla elimina el paquet i no activa el contador.

Afegir una regla a una posició determinada

Per defecte, al afegir una regla aquesta es col.loca al final de la cadena.

Si vols afegir una regla en una posició determinada, has d'utilitzar el handle com a referència:

```
box@filter:~$ sudo nft -a list chain filter output
table ip filter {
        chain output { # handle 7
                type filter hook output priority filter; policy accept;
                ip daddr 8.8.8.8 counter packets 2 bytes 168 # handle 8
                ip daddr 8.8.8.8 counter packets 1 bytes 84 drop # handle 10
        }
}
box@filter:~$ sudo nft add rule filter output position 8 \
> ip daddr 1.1.1.1 counter
box@filter:~$
box@filter:~$ sudo nft -a list chain filter output
table ip filter {
        chain output { # handle 7
                type filter hook output priority filter; policy accept;
                ip daddr 8.8.8.8 counter packets 2 bytes 168 # handle 8
                ip daddr 1.1.1.1 counter packets 0 bytes 0 # handle 11
                ip daddr 8.8.8.8 counter packets 1 bytes 84 drop # handle 10
        }
}
box@filter:~$
```

La regla s'afegeix a continuació de la posició indicada.

Si enlloc de `add` utilitzes `insert` la regla s'afegeix abans de la posició indicada.

```
box@filter:~$ sudo nft insert rule filter output position 8 \
> ip daddr 1.1.1.1 counter
box@filter:~$
box@filter:~$ sudo nft -a list chain filter output
table ip filter {
        chain output { # handle 7
                type filter hook output priority filter; policy accept;
                ip daddr 1.1.1.1 counter packets 0 bytes 0 # handle 12
                ip daddr 8.8.8.8 counter packets 2 bytes 168 # handle 8
                ip daddr 1.1.1.1 counter packets 0 bytes 0 # handle 11
                ip daddr 8.8.8.8 counter packets 1 bytes 84 drop # handle 10
        }
}
box@filter:~$
```

insert

Com has vist abans, a diferència de `add`, `insert` afegeix una regla abans de la posició indicada.

Si no indiques un “handle” la regla s'afegeix al principi de la cadena.

```
box@filter:~$ sudo nft insert rule filter output \
> ip daddr 127.0.0.1 counter
box@filter:~$
box@filter:~$ sudo nft -a list chain filter output
table ip filter {
        chain output { # handle 7
                type filter hook output priority filter; policy accept;
                ip daddr 127.0.0.1 counter packets 2 bytes 214 # handle 13
                ip daddr 1.1.1.1 counter packets 0 bytes 0 # handle 12
                ip daddr 8.8.8.8 counter packets 2 bytes 168 # handle 8
                ip daddr 1.1.1.1 counter packets 0 bytes 0 # handle 11
                ip daddr 8.8.8.8 counter packets 1 bytes 84 drop # handle 10
        }
}
box@filter:~$
```

Eliminar regles

Pots eliminar una regla mitjançant el "handle"

```
box@filter:~$ sudo nft delete rule filter output handle 10
box@filter:~$ sudo nft delete rule filter output handle 12
box@filter:~$
box@filter:~$ sudo nft -a list chain filter output
table ip filter {
        chain output { # handle 7
                type filter hook output priority filter; policy accept;
                ip daddr 127.0.0.1 counter packets 12 bytes 1284 # handle 13
                ip daddr 8.8.8.8 counter packets 2 bytes 168 # handle 8
                ip daddr 1.1.1.1 counter packets 0 bytes 0 # handle 11
        }
}
box@filter:~$
```

També pots eliminar totes les regles d'una cadena amb una única ordre :

```
box@filter:~$ sudo nft flush chain filter output
box@filter:~$
box@filter:~$ sudo nft -a list chain filter output
table ip filter {
        chain output { # handle 7
                type filter hook output priority filter; policy
        }
}
box@filter:~$
```

Script

Encara que és útil interactuar amb **`nftables`** de manera interactiva, sobretot si ho fas amb Python tal com s'explica a ASIX-11-UF3-2 - Nftables, és una mica tediós si ho has de fer-ho a mà.

Nftables **permet executar scripts** de manera atòmica: s'aplica tot l'script o no s'aplica.

D'aquesta manera tens garantit que mai un script s'aplicarà a mitges.

1. Crea un fitxer `test.nft` :

```
test.nft
1   #!/usr/sbin/nft -f
2
3   add rule filter output ip daddr 1.1.1.1 counter
4
```

2. Fes que el fitxer sigui executable, executa **`test.nft`** i verifica que la regla s'ha afegit :

```
box@filter:~$ sudo ./test.nft
box@filter:~$ sudo nft list chain filter output
table ip filter {
        chain output {
                type filter hook output priority filter; policy accept;
                ip daddr 1.1.1.1 counter packets 0 bytes 0
        }
}
box@filter:~$
```

3. Però, què hi passa si torno a executar l'script ?

```
box@filter:~$ sudo ./test.nft
box@filter:~$ sudo nft list chain filter output
table ip filter {
        chain output {
                type filter hook output priority filter; policy accept;
                ip daddr 1.1.1.1 counter packets 0 bytes 0
                ip daddr 1.1.1.1 counter packets 0 bytes 0
        }
}
box@filter:~$
```

Tinc dos regles repetides!

🔎He de de modificar l'script perquè primer elimini les regles de la cadena …

4. Però, millor fer un script ben fet que ho faci tot 🙂!

```
test.nft
1   #!/usr/sbin/nft -f
2
3   flush ruleset          <-- BORRO TOT
4
5   add table filter
6
7   add chain filter input { type filter hook input priority 0; }
8   add chain filter input-ssh { type filter hook input priority -1; }
9   add chain filter output { type filter hook output priority 0; }
10
11  add rule filter output ip daddr 8.8.8.8 counter
12
```

Amb l'ordre **`flush ruleset`** eliminem tot:

```
box@filter:~$ sudo nft flush ruleset
box@filter:~$
box@filter:~$ sudo nft list tables
box@filter:~$
```

5. Verifica que pots executar tres vegades l'script i que la configuració és correcte:

```
box@filter:~$ sudo ./test.nft
box@filter:~$ sudo ./test.nft
box@filter:~$ sudo ./test.nft
box@filter:~$
box@filter:~$ sudo nft list table filter
table ip filter {
        chain input {
                type filter hook input priority filter; policy accept;
        }

        chain input-ssh {
                type filter hook input priority filter - 1; policy accept;
        }

        chain output {
                type filter hook output priority filter; policy accept;
                ip daddr 8.8.8.8 counter packets 0 bytes 0
        }
}
box@filter:~$
```

Format estructurat

També pots escriure les **taules, les cadenes i les regles** en el mateix format que et mostra **`nft`** quan executem l'ordre **`list`**.

1. Modifica el fitxer `test.nft` :

test.nft

```
#!/usr/sbin/nft -f

flush ruleset

table ip filter {
  chain input-ssh {
    type filter hook input priority filter - 1; policy accept;
  }
  chain input {
    type filter hook input priority filter; policy accept;
  }
  chain output {
    type filter hook output priority filter; policy accept;
    ip daddr 1.1.1.1 counter
    counter
  }
}

```

2. Borra la taula `filter`, executa l'script `test.nft` i verifica amb `nft list` que el resultat és el mateix:

```
box@filter:~$ sudo nft delete table filter
box@filter:~$ sudo ./test.nft
box@filter:~$ sudo nft list table filter
table ip filter {
        chain input-ssh {
                type filter hook input priority filter - 1; policy accept;
        }

        chain input {
                type filter hook input priority filter; policy accept;
        }

        chain output {
                type filter hook output priority filter; policy accept;
                ip daddr 1.1.1.1 counter packets 0 bytes 0
                counter packets 113 bytes 7945
        }
}
box@filter:~$
```

Comentaris i variables

1. A continuació modifica l'script afegint comentaris i variables :

test.nft

```
#!/usr/sbin/nft -f

flush ruleset

# Windows Host
define host = 192.168.56.1

# Google Public DNS
define dns_google = { 8.8.8.8, 8.8.4.4}

table ip filter {
  chain input-ssh {
    type filter hook input priority filter - 1; policy accept;
    ip saddr != $host tcp dport 22 drop;
  }
  chain input {
    type filter hook input priority filter; policy accept;
  }
  chain output {
    type filter hook output priority filter; policy accept;
    ip daddr $dns_google counter
    counter
  }
}

```

2. Executa l'script i verifica que s'ha modificat:

```
box@filter:~$ sudo nft list table filter
table ip filter {
        chain input-ssh {
                type filter hook input priority filter - 1; policy accept;
                ip saddr != 192.168.56.1 tcp dport 22 drop
        }

        chain input {
                type filter hook input priority filter; policy accept;
        }

        chain output {
                type filter hook output priority filter; policy accept;
                ip daddr { 8.8.4.4, 8.8.8.8 } counter packets 0 bytes 0
                counter packets 1450 bytes 102889
        }
}
box@filter:~$
```

Pots veure que els comentaris han desaparegut i enlloc de les variables hi ha els valors definits a les variables.

ssh

La regla ssh funciona, sinó ara mateix hauries perdut la connexió.

Pots veure que la regla s'aplica a:

1. `ip saddr != 192.168.56.1` – Paquets IP que no tenen com adreça origen la 192.168.56.1 (`saddr` = *source address*)
2. `tcp` – Que són paquets TCP
3. `dport 22` – Que tenen com a port de destinació el 22 (`dport` = *destination port*)

I que quan la regla fa "match":

1. `drop` – Els paquets són eliminats.

També pots comprovar que la màquina `alpha` no pot connectar-se al port ssh de `filter`:

```
box@filter:~$ ip -brief addr | grep enp0s8
enp0s8           UP             192.168.56.15/24 fe80::a00:27ff:fe39:1
box@filter:~$
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
box@alpha:~$ ssh 192.168.56.15
ssh: connect to host 192.168.56.15 port 22: Connection timed out
box@alpha:~$
```

Fixa't que l'error és molt diferent que si intento una conexió ssh d'`alpha` a `beta`:

```
box@beta:~$ ip -brief addr | grep enp0s8
enp0s8           UP             192.168.56.17/24 fe80::a00:27ff:fe1e:141
box@beta:~$
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
box@alpha:~$ ssh 192.168.56.17
The authenticity of host '192.168.56.17 (192.168.56.17)' can't be establ
ED25519 key fingerprint is SHA256:4PoJG3bJjnfM9lLx50jSq0q/kcwEKLD9y6d3ST
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.17' (ED25519) to the list of know
box@192.168.56.17: Permission denied (publickey).
box@alpha:~$
```

ping

Fes un ping a `8.8.8.8` i un altre a `8.8.4.4` i verifica que la regla `ip daddr { … }` funciona:

```
box@filter:~$ ping -c 1 8.8.8.8 > /dev/null
box@filter:~$ sudo nft list chain filter output
table ip filter {
        chain output {
                type filter hook output priority filter; policy accept;
                ip daddr { 8.8.4.4, 8.8.8.8 } counter packets 1 bytes 84
                counter packets 290 bytes 19979
        }
}
box@filter:~$ ping -c 1 8.8.4.4 > /dev/null
box@filter:~$ sudo nft list chain filter output
table ip filter {
        chain output {
                type filter hook output priority filter; policy accept;
                ip daddr { 8.8.4.4, 8.8.8.8 } counter packets 2 bytes 168
                counter packets 805 bytes 55636
        }
}
box@filter:~$
```

systemd

nftables és un servei del sistema que es controla a través de **`systemd`**.

El servei executa l'script `/etc/nftables.conf`.

Si vols pots modificar aquest script directament o inclure **test.nft** en aquest script perquè també s'executi.

1. Mou el fitxer test<.nft a /etc/
2. Modifica el fitxer `/etc/nftables.conf` :

```
  GNU nano 6.2                /etc/nftables.conf
#!/usr/sbin/nft -f

include "/etc/test.nft"
```

3. Borra tota la configuració i arrenca el firewall amb **`systemctl`** :

```
box@filter:~$ sudo nft flush ruleset
box@filter:~$
box@filter:~$ sudo systemctl start nftables
box@filter:~$
box@filter:~$ sudo nft list table filter
table ip filter {
        chain input-ssh {
                type filter hook input priority filter - 1; policy accept;
                ip saddr != 192.168.56.1 tcp dport 22 drop
        }

        chain input {
                type filter hook input priority filter; policy accept;
        }

        chain output {
                type filter hook output priority filter; policy accept;
                ip daddr { 8.8.4.4, 8.8.8.8 } counter packets 0 bytes 0
                counter packets 567 bytes 38757
        }
}
box@filter:~$
```

4. Si vols que el firewall arrenqui al inciar el sistema has d'habilitar el servei `nftables` :

```
box@filter:~$ sudo systemctl enable nftables
Created symlink /etc/systemd/system/sysinit.target.wants/nftable
box@filter:~$
```

Activitat

ssh

1. Configura un firewall a cada màquina (**`filter`, `alpha` i `beta`**) que **només** permeti connexions **ssh** des de la màquina host (Windows) i que s'executi amb **`systemctl`**. Has de modificar el fitxer `/etc/nftables.conf`.

   Consell: Primer verifica que el teu script funciona abans de posar-lo permanent.

```
  GNU nano 6.2                /etc/nftables.conf *
#!/usr/sbin/nft -f

flush ruleset

table ip filter {
  chain input {
    type filter hook input priority 0;
    ip saddr != 192.168.56.1 tcp dport 22 drop;
  }
}
```

2. **Activa el firewall:**

```
box@alpha:~$ sudo systemctl start nftables
box@alpha:~$
```

3. **Comprovació**

   exemple des de filter cap a alpha

```
box@filter:~$ ssh 192.168.56.18
^C
box@filter:~$
```

   i des de alpha cap a filter

```
box@alpha:~$ ssh 192.168.56.15
^C
box@alpha:~$
```

http

1. Arrencar un servidor nginx amb `apt` a les tres màquines (`apt install -y nginx`).
2. Configura nftables perquè només accepti connexions entrants ssh des de la màquina host:

```
  GNU nano 6.2                /etc/nftables.conf
#!/usr/sbin/nft -f

flush ruleset

table ip filter {
  chain input {
    type filter hook input priority 0; policy drop;
    iifname "lo" accept;
    ip saddr 192.168.56.1 tcp dport 22 accept;
  }
}
```

Fixa't que:

- Hem canviat la política per defecte de la cadena si no hi ha cap regla que accepti el paquet: `policy drop;`
- També que hem afegit la regla `iifname "lo" accept;`

  `iifname` = *input interface name*

  Això és necessari perquè alguns serveis bàsics del sistema operatiu funcionen amb la interfície "**loopback**", com per exemple, **la resolució de noms**.

- Que hem reescrit la regla ssh: `ip saddr 192.168.56.1 tcp dport 22 accept;`

  Ara enlloc d'eliminar els paquets que no venen de la IP 192.168.56.1 només acceptem els que venen de la IP 192.168.56.1.

3. Torna a carregar les regles :

```
box@alpha:~$ sudo systemctl restart nftables
box@alpha:~$
```

4. Verifica que et pots connectar-te al servidor nginx des de la mateixa màquina, però no al servidor nginx de les altres màquines.

```
box@alpha:~$ curl -s localhost | head -2
<!DOCTYPE html>
<html>
box@alpha:~$ curl 192.168.56.15
^C
box@alpha:~$ curl 192.168.56.17
^C
box@alpha:~$
```

5. Configura nftables perquè també accepti connexions entrants tcp al port 80:

```
  GNU nano 6.2              /etc/nftables.conf *
#!/usr/sbin/nft -f

flush ruleset

table ip filter {
  chain input {
    type filter hook input priority 0; policy drop;
    iifname "lo" accept;
    ip saddr 192.168.56.1 tcp dport 22 accept;
    tcp dport 80 accept;
  }
}
```

6. Verifica que les regles estan actives:

```
box@filter:~$ sudo nft list table filter
table ip filter {
        chain input {
                type filter hook input priority filter; policy d
                iifname "lo" accept
                ip saddr 192.168.56.1 tcp dport 22 accept
                tcp dport 80 accept
        }
}
box@filter:~$
```

```
box@filter:~$ sudo nft -a list chain filter output
table ip filter {
        chain output { # handle 7
                type filter hook output priority filter; policy accept;
                ip daddr 8.8.8.8 counter packets 0 bytes 0 # handle 8
                ip daddr 8.8.8.8 counter packets 0 bytes 0 # handle 10
        }
}
box@filter:~$
```

7. Verifica que encara que acceptes els paquets destinats al port 80, no et pots connectar als servidor nginx al port 80 de les altres màquines: (Depend com tinguis les regles probablemente et connectis)

```
box@beta:~$ curl -s 192.168.56.15 | head -1
^C
box@beta:~$ curl -s 192.168.56.16 | head -1
^C
box@beta:~$ curl -s 192.168.56.17 | head -1
<!DOCTYPE html>
box@beta:~$
```

Tracing

Quan tenim un problema amb la configuració de nftables a més d'utilitzar un contador ("counter") per veure que una regla s'ha activat podem fer un "**tracing**".

1. Modifica el fitxer `/etc/nftables.conf`, i afegeix al final l'expressió: `meta nftrace set 1`

```
  GNU nano 6.2                    /etc/nftables.conf *
#!/usr/sbin/nft -f

flush ruleset

table ip filter {
  chain input {
    type filter hook input priority 0; policy drop;
    iifname "lo" accept;
    ip saddr 192.168.56.1 tcp dport 22 accept;
    tcp dport 80 accept;
    meta nftrace set 1;
  }
}
```

2. Torna a carregar nftables i comença a monitoritzar l'entrada de paquets: nft monitor trace

```
box@filter:~$ sudo systemctl restart nftables
box@filter:~$
box@filter:~$ sudo nft monitor trace
trace id 6b9d7668 ip filter input packet: iif "enp0s8" ether saddr 0a:00:27:
00:00:14 ether daddr ff:ff:ff:ff:ff:ff ip saddr 192.168.56.1 ip daddr 192.16
8.56.255 ip dscp cs0 ip ecn not-ect ip ttl 128 ip id 17225 ip length 78 udp
sport 137 udp dport 137 udp length 58 @th,64,96 0xc0a30110000100000000000
trace id 6b9d7668 ip filter input rule meta nftrace set 1 (verdict continue)
trace id 6b9d7668 ip filter input verdict continue
trace id 6b9d7668 ip filter input policy drop
```

`(host) 192.168.56.1:137` → `192.168.56.255:137 (broadcast)`

Pots veure que cada segon el host (`saddr 192.168.56.1`) envia a l'adreça broadcast (`daddr 192.168.56.255`) un paquet **UDP** desde el port 137 (`sport 137`) al port 137 (`dport 137`).

Els Windows fan servir el port 137 per traduir una adreça IP a un nom "Windows" de tots els hosts que té accés, per exemple una impresora, per presentar noms "Windows" als usuaris.

3. Afegim una regla per eliminar aquest paquets:

```
  GNU nano 6.2                    /etc/nftables.conf *
#!/usr/sbin/nft -f

flush ruleset

table ip filter {
  chain input {
    type filter hook input priority 0; policy drop;
    udp dport 137 drop;
    iifname "lo" accept;
    ip saddr 192.168.56.1 tcp dport 22 accept;
    tcp dport 80 accept;
    meta nftrace set 1
  }
}
```

4. Tornem-hi !

```
box@filter:~$ sudo systemctl restart nftables
box@filter:~$ sudo nft monitor trace
```

5. A veure que passa …

```
box@filter:~$ sudo systemctl restart nftables
box@filter:~$ sudo nft monitor trace

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

box@alpha:~$ curl -s 192.168.56.15 | head -1
```

El que era d'esperar, els paquets arriben a `filter` i són acceptats ja que podem veure que la regla `meta nftrace` no s'activa.

6. Per estar segurs, mou el tracing abans de la regla "http" i afegeix un contador a la regla "http":

```
GNU nano 6.2                                    /etc/nftables.conf *
    type filter hook input priority 0; policy drop;
    udp dport 137 drop;
    iifname "lo" accept;
    ip saddr 192.168.56.1 tcp dport 22 accept;
    meta nftrace set 1;
    tcp dport 80 counter accept;
  }
}
```

7. A veure que està passant:

```
box@filter:~$ sudo systemctl restart nftables
box@filter:~$ sudo nft monitor trace
trace id c4fccc1a ip filter input packet: iif "enp0s8" ether saddr 08:00:2
7:aa:27:81 ether daddr 08:00:27:39:19:ca ip saddr 192.168.56.16 ip daddr 1
92.168.56.15 ip dscp cs0 ip ecn not-ect ip ttl 64 ip id 33098 ip length 60
 tcp sport 58278 tcp dport 80 tcp flags == syn tcp window 64240
trace id c4fccc1a ip filter input rule meta nftrace set 1 (verdict continu
e)
trace id c4fccc1a ip filter input rule tcp dport 80 counter packets 0 byte
s 0 accept (verdict accept)
```

PROBLEMS OUTPUT TERMINAL ··· box

```
box@alpha:~$ curl -s 192.168.56.15 | head -1
```

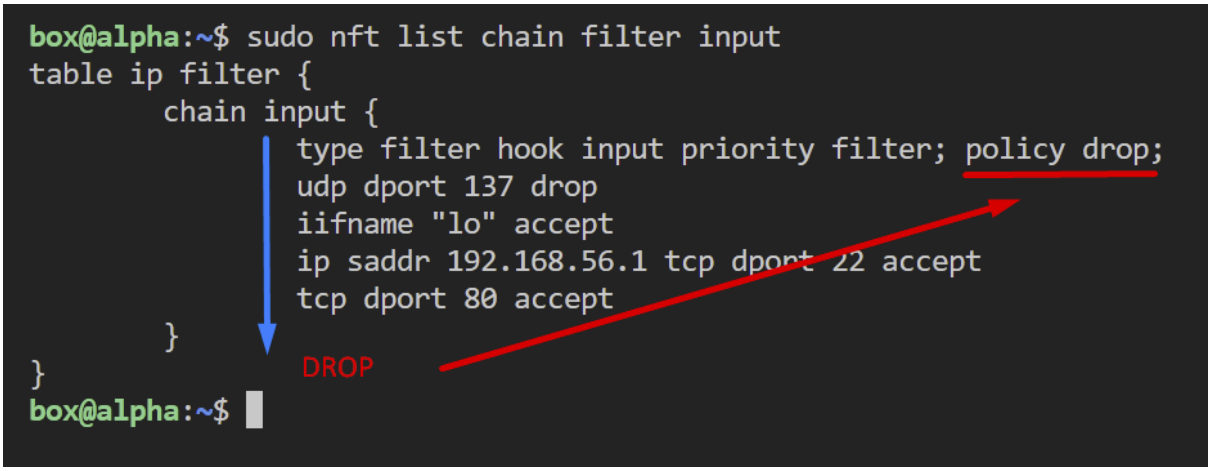
(alpha) 192.168.56.16:58278 ⟶ 192.168.56.255:80 (filter)

Doncs què `alpha` està enviant paquets des del port `58278` (`tcp sport 58278`) al port `80` de filter (`tcp dport 80`).

Per tant, `filter` ha de tornar la resposta al port `58278` d'`alpha`.

(alpha) 192.168.56.16:58278 ⟵ 192.168.56.255:80 (filter)

I el firewall d'`alpha` no permet entrades al port `58278` 🧐



Quan **`curl`** fa una connexió **`HTTP`** ha de buscar un port que no s'estigui fent servir (**`sport`**), obtenir el recurs i llavors el pot fer servit per connectar-se al port 80 (**`dport`**) del servidor web).

conntrack

Fins ara estem aplicant regles sense estat: només tenim en compte el paquet que estem filtrant.

Però hi ha molts paquets que estan relacionats perquè pertanyen a una mateixa connexió.

T'enrecordes que al principi al explicar el **hooks** hem parlat de la "**flowtable**"?

Netfilter té un sistema de seguiment de connexions que permet rastrejar quins paquets pertanyen a la mateixa connexió.

En l'activitat anterior la connexió era:

`(alpha) 192.168.56.16:58278` ⟶ `192.168.56.255:80 (filter)`

`(alpha) 192.168.56.16:58278` ⟵ `192.168.56.255:80 (filter)`

Per tant el paquet de tornada és una resposta al paquet d'anada al port 80 de filter que sí que estaba permés.

Podem afegir aquesta regla a la cadena output:

```
ct state established,related accept
```

Aquesta regla diu que:

- `ct` – És una expressió de tipus "connection tracking"
- `state` – Fem un tracking de l'estat de la connexió
- `established, related` — Són els estats de conntrack que volem fer "match"

1. Modifica i reinicia **tots els** firewalls amb la nova configuració:

```
  GNU nano 6.2                    /etc/nftables.conf
#!/usr/sbin/nft -f

flush ruleset

table ip filter {
  chain input {
    type filter hook input priority 0; policy drop;
    udp dport 137 drop;
    iifname "lo" accept;
    ct state established, related accept;
    ip saddr 192.168.56.1 tcp dport 22 accept;
    tcp dport 80 accept;
    meta nftrace set 1;
  }
}
```

2. Verifica que totes les connexions “http” funcionen:

```
box@beta:~$ curl -s 192.168.56.15 | head -1
<!DOCTYPE html>
box@beta:~$ curl -s 192.168.56.16 | head -1
<!DOCTYPE html>
box@beta:~$ curl -s 192.168.56.17 | head -1
<!DOCTYPE html>
box@beta:~$
```

docker

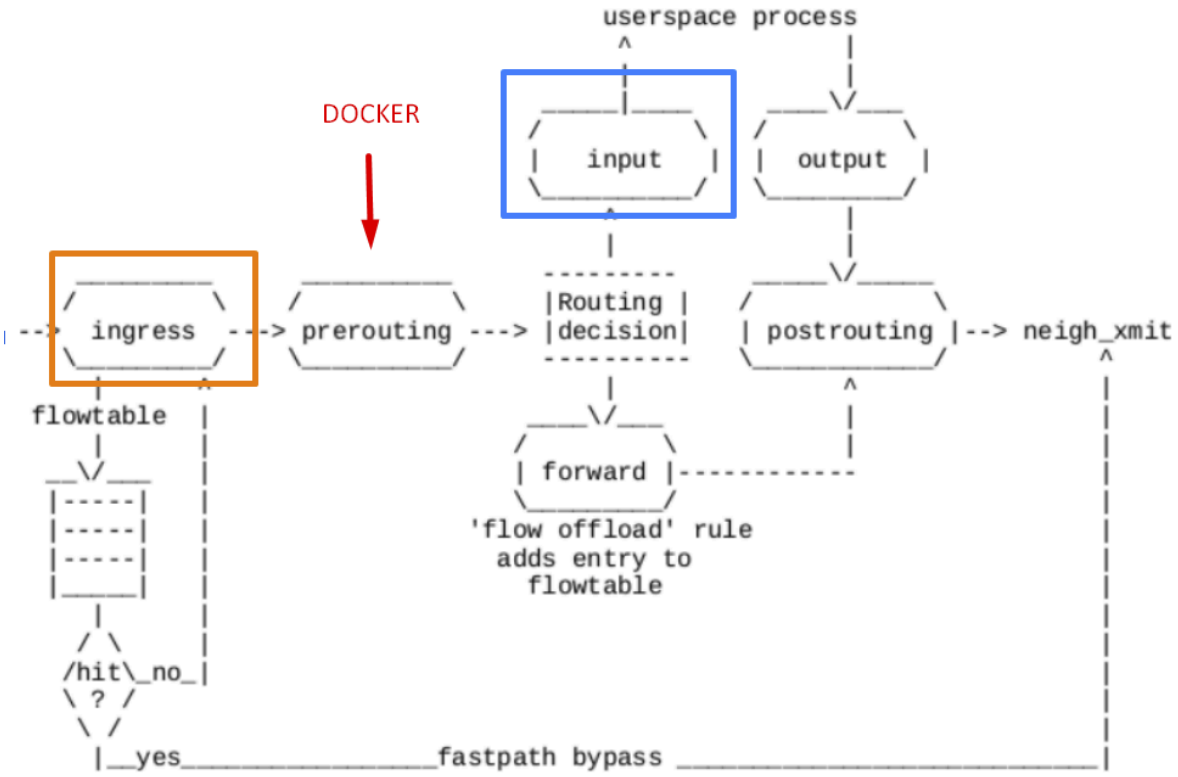
1. Arrenca un contenidor nginx al port 8000:

```
box@filter:~$ docker run --rm -d --name nginx \
> -p 8000:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
```

2. Verifica tal com vas fer a SMX-6-UF5-6 - UFW que el “**hook**” input **no pot filtrar** els paquets dirigits al port 8000 del contenidor:

```
box@beta:~$ curl -s 192.168.56.15:8000 | head -1
<!DOCTYPE html>
box@beta:~$
```

El motiu és que Docker s’enganxa al “hook” **`prerouting`** que va abans del “**hook**” `input`:



Si volem bloquejar la connexió ho podem fer des del "hook" `ingress` :

```
  GNU nano 6.2                    /etc/nftables.conf
#!/usr/sbin/nft -f

flush ruleset

table ip filter {
  chain input {
    type filter hook input priority 0; policy drop;
    udp dport 137 drop;
    iifname "lo" accept;
    ct state established, related accept;
    ip saddr 192.168.56.1 tcp dport 22 accept;
    tcp dport 80 accept;
    meta nftrace set 1;
  }
}

table netdev filter {
  chain enp0s8 {
    type filter hook ingress device enp0s8 priority 0;
    tcp dport 8000 drop;
  }
}
```

Verifica que `beta` ja no es pot connectar al port 8000 de `filter`:

```
box@beta:~$ curl 192.168.56.15:8000
curl: (28) Failed to connect to 192.168.56.15 port 8000 after 129389 ms:
box@beta:~$
```