Memento nftables

Auteur : Antoine Pernot

• Structure du fichier /etc/nftables.conf

```
#!/usr/sbin/nft -f
flush ruleset
# Exemple table IPv4 et IPv6
table inet nomTable {
 # Chaine de flux entrant
 chain input {
  type filter hook input priority filter;
  # Refuser tout trafic entrant par defaut
  policy drop;
  # Inserer les regles entrantes ici
  # Exemple : tcp dport { 80, 443 } accept
  # Autoriser trafic pour
  # les sessions deja etablies
  ct state {established,related} accept
 }
 # Chaine de flux en transit vers une autre
 # destination
 chain forward {
  type filter hook forward priority filter;
 }
 # Chaine de flux sortant
 chain output {
  type filter hook output priority filter;
 }
 # Chaine avant routage
 chain prerouting {
  type nat hook prerouting priority dstnat;
 }
 # Chaine apres routage
 chain postrouting {
  type nat hook postrouting priority srcnat;
 }
}
```

• Tables

Type	Description
ip	IPv4 (par défaut)
ip6	IPv6
inet	IPv4 et IPv6
arp	ARP (ex arptables)
bridge	Bridge (ex ebtables)
netdev	Interface unique

• Construction règles

```
<filtre> [filtre ...] <action>
```

• Filtres courants `ip`

Filtrer par protocole :

```
ip protocol tcp
ip protocol { tcp, icmp, udp }
```

Filtrer par adresse source :

```
ip saddr 10.0.0.1
ip saddr { 10.0.0.1, 10.0.0.2 }
ip saddr 10.0.0.1-10.0.0.250
ip saddr 10.0.0.0/24
ip saddr != 10.0.0.254
```

Filtrer par adresse destination :

```
ip daddr 10.0.0.1
```

• Filtres courants `ip6`

Filtrer par adresse source :

```
ip6 saddr fe80:cafe::beef
ip6 saddr { fe80::1, fe80::2}
ip6 saddr fe80::1-fe80::250
ip6 saddr fe80:cafe::/64
ip6 saddr fe80:cafe::acdc
```

Filtrer par adresse destination :

```
ip6 daddr fe80:cafe::beef
```

• Filtres courants `tcp` **et** `udp`

Filtrer par port source :

```
tcp sport 10250
udp sport { 53, 69 }
tcp sport { 8080-8089 }
udp saddr { dhcp, dns }
```

Filtrer par port destination :

```
tcp dport 22
```

• Ping `icmp` **et** `icmpv6`

```
icmp type echo-request accept
icmpv6 type echo-request accept
```

• Filtrer les connexion établies

```
ct state { established, related }
```

• Filtrer selon les interfaces

Filtrer par interface d'entrée :

```
iifname lo
iifname { eth0, eth1 }
iifname != wlan0
```

Filtrer par interface de sortie :

```
oifname eth0
```

• Actions

Accepter :

```
accept
```

Refuser :

```
drop
```

Limiter le trafic :

```
limit rate 1024 mbytes/second
```

• Actions pour `prerouting`

Redirection de port :

```
redirect to 8080
```

Exemple :

```
iifname eth0 tcp dport 80 redirect to 8080
```

NAT destination :

```
dnat to 10.0.0.1:8080
```

Exemple :

```
tcp dport 8080 dnat to 10.0.0.1:80
```

• Actions pour `postrouting`

NAT source :

```
masquerade
```

• Gérer le service nftables

Activer nftables au démarrage :

```
systemctl enable nftables
```

Recharger les règles :

```
systemctl restart nftables
```

Lister les règles actives :

```
nft list ruleset
```