nftables Linux packet filtering

CONCEPTS

Nothing exists by default: you create your own tables and chains. A table only filters if it has a base chain attached to a hook.

Table family: ip ip6 inet arp bridge netdev

Chain base (hook + priority) or regular

Rule matches then statements

Quote braces and semicolons in the shell ('{ ... }') or escape them with a backslash. In files they need no quoting.

INSPECT

`nft list ruleset`
Print all tables

`nft list table <family> <table>`
One table

`nft -a list ruleset`
Show rule handles

TABLES

`nft add table inet <table>`
Create a table

CHAINS

`nft add chain inet <table> input '{ type filter hook input priority 0; policy drop; }'`
Base chain on a hook

RULES

`nft add rule inet <table> <chain> <match> <statement>`
Append a rule

`nft insert rule inet <table> <chain> <match> <statement>`
Insert at top of chain

`nft delete rule inet <table> <chain> handle <n>`
Delete a rule by handle

`nft add rule inet <table> input tcp dport 22 accept`
Allow SSH

RULES (cont.)

`nft add rule inet <table> input tcp dport { 80, 443 } accept`
Allow several ports

`nft add rule inet <table> input ip saddr 10.0.0.0/8 drop`
Block a network

`nft add rule inet <table> input iifname lo accept`
Allow loopback

`nft add rule inet <table> input ct state established,related accept`
Allow replies

MATCHES

`ip saddr / ip daddr <addr>`
IPv4 source / destination

`tcp dport / udp dport <port>`
Destination port

`iifname "eth0" / oifname "eth0"`
Interface by name

`ct state new`
Connection tracking state

`<field> { a, b, c }`
Match any of several values

STATEMENTS

`accept / drop`
Final verdict

`counter`
Count packets and bytes

`log prefix "<text> "`
Log to the kernel log

NAT

`nft add table ip nat`
Table for NAT

`nft add rule ip nat postrouting oifname "eth0" masquerade`
Share an uplink

SAVE AND RESTORE

`nft list ruleset > <file>`
Save the ruleset

`nft -f <file>`
Load a rule file