iptables Linux packet filter

CONCEPTS

Rules are checked top to bottom; the first match with a final target (ACCEPT, DROP, REJECT) decides. No match falls to the chain policy.

PREROUTING raw, mangle, nat (DNAT)
↓ routing
INPUT / FORWARD local host or routed
↓ local out
OUTPUT locally made packets
↓ routing
POSTROUTING mangle, nat (SNAT)

Packet path (simplified): table order differs per chain

Changes apply instantly but vanish at reboot unless you save them (e.g. iptables-persistent on Debian). Dumped rules are not loaded on boot by themselves.

A wrong DROP policy over SSH locks you out. Add the ACCEPT rule for your port first, and keep a second session open.

Modern distributions favour nftables; iptables-nft is a compatibility front end over it. See the nftables sheet.

TABLES AND CHAINS

iptables -t *<table>* **...**
Pick a table (default filter)

Tables and built-in chains

Table	Chains	Used for
filter	INPUT FORWARD OUTPUT	Allow or block (default)
nat	PREROUTING INPUT OUTPUT POSTROUTING	Address translation
mangle	all five	Alter packets, marks
raw	PREROUTING OUTPUT	Skip connection tracking
security	INPUT FORWARD OUTPUT	SELinux-style marks

sudo iptables -N *<chain>*
Create a user chain

TABLES AND CHAINS (cont.)

sudo iptables -X *<chain>*
Delete an empty user chain

sudo iptables -P INPUT DROP
Set default policy

sudo iptables -A INPUT -j *<chain>*
Jump into a user chain

LIST AND INSPECT

sudo iptables -L
List rules of the filter table

sudo iptables -L INPUT -n -v --line-numbers
Chain with numbers and counters

sudo iptables -S *<chain>*
Print a chain as commands

sudo iptables -C INPUT -p tcp --dport *<port>* **-j ACCEPT**
Test if a rule exists

ADD AND REMOVE RULES

sudo iptables -A *<chain> <rule>*
Append a rule

sudo iptables -I *<chain>* **[***<num>***]** *<rule>*
Insert (default: top)

sudo iptables -D *<chain> <num>*
Delete by position

sudo iptables -D *<chain> <rule>*
Delete by matching rule

sudo iptables -R *<chain> <num> <rule>*
Replace a rule

sudo iptables -F [*<chain>***]**
Flush rules

MATCHES

-p tcp / udp / icmp
Match protocol

-s *<ip>***[/***<mask>***]**
Match source address

-d *<ip>***[/***<mask>***]**
Match destination address

-i *<iface>*
Match incoming interface

MATCHES (cont.)

-o *<iface>*
Match outgoing interface

-p tcp --dport *<port>*
Match destination port

-m multiport --dports *<p1>*,*<p2>*
Several ports in one rule

-m conntrack --ctstate ESTABLISHED,RELATED
Match connection state

TARGETS

-j ACCEPT
Let the packet through

-j DROP
Discard silently

-j REJECT --reject-with tcp-reset
Refuse with a reply

-j LOG --log-prefix "*<text>***: "**
Log to the kernel log

NAT AND FORWARDING

sudo iptables -t nat -A POSTROUTING -o *<iface>* **-j MASQUERADE**
Share one dynamic address

sudo iptables -t nat -A POSTROUTING -s *<net>* **-j SNAT --to-source** *<ip>*
Rewrite source address

sudo iptables -t nat -A PREROUTING -p tcp --dport *<port>* **-j DNAT --to-destination** *<ip>*:*<port>*
Forward a port inward

sudo sysctl -w net.ipv4.ip_forward=1
Enable IPv4 routing

sudo iptables -A FORWARD -i *<in>* **-o** *<out>* **-j ACCEPT**
Allow routed traffic

SAVE AND RESTORE

sudo iptables-save
Dump all tables

sudo ip6tables ...
Same commands for IPv6