# firewalld Zone-based firewall

**START HERE**

**`sudo firewall-cmd --state`**
Is the daemon running?

**`sudo firewall-cmd --get-active-zones`**
Zones in use and their interfaces

**`sudo firewall-cmd --list-all`**
Everything allowed in a zone

**`sudo firewall-cmd --add-service=<service>`**
Allow a service (runtime)

**`sudo firewall-cmd --permanent --add-port=<port>/tcp`**
Allow a port for good

**`sudo firewall-cmd --reload`**
Apply permanent config

**`sudo firewall-cmd --runtime-to-permanent`**
Save what is live now

**`sudo firewall-cmd --get-services`**
Known service names

**Packet arrives** on an interface or from a source → **matched to** **Zone** trust level → **checked against** **Services, ports, rich rules** allowed traffic → **otherwise** **Zone target** default: reject

## CONCEPTS

Without --zone, commands act on the default zone. Without --permanent, they change only the running firewall and are lost on reload or reboot.

Runtime and permanent are two separate configs. Test with a runtime change, then keep it with --runtime-to-permanent.

Interfaces match a zone first by interface, and a source address matches before the interface does.

## STATE AND RELOAD

**`firewall-cmd --state`**
Check the daemon is running

**`firewall-cmd --reload`**
Reload permanent config, drop runtime changes

*Keeps existing connections; runtime-only rules vanish.*

**`firewall-cmd --runtime-to-permanent`**
Save runtime config as permanent

**`systemctl enable --now firewalld`**
Start the daemon now and at boot

## ZONES

**`firewall-cmd --get-zones`**
List all zone names

**`firewall-cmd --get-active-zones`**
Zones with interfaces or sources

**`firewall-cmd --get-default-zone`**
Show the default zone

**`firewall-cmd --set-default-zone=<zone>`**
Change the default zone

**`firewall-cmd --list-all`**
Show a zone's full config

**`firewall-cmd --get-zone-of-interface=<iface>`**
Zone an interface is in

## ZONES (cont.)

**`firewall-cmd --zone=<zone> --list-all`**
Inspect a specific zone

**`firewall-cmd --permanent --zone=<zone> --set-target=DROP`**
Drop unmatched traffic

*Targets: default, ACCEPT, REJECT, DROP.*

## SERVICES AND PORTS

**`firewall-cmd --add-service=<service>`**
Allow a named service

**`firewall-cmd --remove-service=<service>`**
Stop allowing a service

**`firewall-cmd --list-services`**
Services allowed in a zone

**`firewall-cmd --get-services`**
All defined service names

**`firewall-cmd --add-port=<port>/tcp`**
Open a TCP port

**`firewall-cmd --add-port=<from>-<to>/udp`**
Open a UDP port range

**`firewall-cmd --remove-port=<port>/tcp`**
Close a port

**`firewall-cmd --list-ports`**
Ports open in a zone

**`firewall-cmd --add-service=<service> --timeout=10m`**
Allow temporarily, auto-revert

*Runtime only; cannot combine with --permanent. Units: s, m, h.*

## RICH RULES

**`firewall-cmd --add-rich-rule='<rule>'`**
Add a rule with conditions

**`firewall-cmd --list-rich-rules`**
Show rich rules in a zone

**`firewall-cmd --remove-rich-rule='<rule>'`**
Remove a rule, exact text

## RICH RULES (cont.)

**Rich rule shapes**

| Goal | Rule |
|---|---|
| Block a host | `rule family="ipv4" source address="<ip>" reject` |
| Port from subnet | `rule family="ipv4" source address="<cidr>" port port="<port>" protocol="tcp" accept` |
| Service with rate | `rule service name="<service>" accept limit value="1/m"` |
| Log then drop | `rule family="ipv4" source address="<cidr>" log prefix="fw " level="warning" drop` |

A rich rule is a single string: quote it in the shell and use double quotes inside single quotes.

## INTERFACES AND SOURCES

**`firewall-cmd --zone=<zone> --add-interface=<iface>`**
Put an interface in a zone

**`firewall-cmd --zone=<zone> --change-interface=<iface>`**
Move an interface to a zone

**`firewall-cmd --zone=<zone> --add-source=<cidr>`**
Assign a network to a zone

**`firewall-cmd --permanent --zone=<zone> --add-source=ipset:<name>`**
Bind an ipset to a zone

## NAT AND FORWARDING

**`firewall-cmd --zone=<zone> --add-masquerade`**
Enable NAT on outbound traffic

**`firewall-cmd --add-forward-port=<spec>`**
Forward a port, locally or to a host

*Spec: port=80:proto=tcp:toport=8080[:toaddr=<ip>]*
*A remote toaddr also needs masquerade in the zone.*

**`firewall-cmd --zone=<zone> --add-forward`**
Allow forwarding within a zone

## PANIC AND LOGGING

**`firewall-cmd --panic-on`**
Drop all traffic immediately

*Also kills your SSH session if remote.*

**`firewall-cmd --panic-off`**
Leave panic mode

**`firewall-cmd --set-log-denied=all`**
Log every denied packet

## DEFINITIONS

**`firewall-cmd --permanent --new-ipset=<name> --type=hash:ip`**
Create an IP set

**`firewall-cmd --permanent --ipset=<name> --add-entry=<ip>`**
Add an address to a set

**`firewall-cmd --permanent --service=<name> --add-port=<port>/tcp`**
Add a port to a custom service

**`firewall-cmd --permanent --new-policy=<name>`**
Create a policy between zones

## RECIPES

**Open a web server for good**

```
sudo firewall-cmd --permanent
--add-service=http
sudo firewall-cmd --permanent
--add-service=https
sudo firewall-cmd --reload
sudo firewall-cmd --list-
services
```

**Try a rule, keep it if it works**

```
sudo firewall-cmd --add-
port=8080/tcp --timeout=5m
# test from another machine
sudo firewall-cmd --runtime-to-
permanent
```

**SSH only from the office network**

```
sudo firewall-cmd --permanent
--remove-service=ssh
sudo firewall-cmd --permanent
--add-rich-rule='rule
family="ipv4" source
address="203.0.113.0/24"
service name="ssh" accept'
sudo firewall-cmd --reload
```

**Forward port 80 to a backend**

```
sudo firewall-cmd --permanent
--add-masquerade
sudo firewall-cmd --permanent
--add-forward-port=port=
80:proto=tcp:toaddr=10.0.0.
5:toport=8080
sudo firewall-cmd --reload
```