The command sudo nft list ruleset displays your system's **entire active firewall ruleset** as configured by nftables. This output shows that firewalld is managing the rules using the **inet family** (which covers both IPv4 and IPv6), under a table called firewalld.

**🔍 Let's break this down step by step:**

**🧱 Top-Level Structure**

table inet firewalld {

flags owner,persist

...

}

* This is the firewalld table using the inet family (common for both IPv4 and IPv6).
* The flags owner,persist:
  + owner: firewalld manages this table.
  + persist: it will be saved across reboots.

**🔁 Chains and Hooks**

Your firewall is organized into **chains**, and each chain may be attached to a **hook** (like prerouting, postrouting, input, output). Here's what each part means:

**🔄 Mangle Chains (marking/altering packets)**

**chain mangle\_PREROUTING**

type filter hook prerouting priority mangle + 10; policy accept;

jump mangle\_PREROUTING\_ZONES

* Intercepts **incoming packets before routing** decisions.
* Calls (jump) mangle\_PREROUTING\_ZONES.

**mangle\_PREROUTING\_ZONES**

iifname "ens160" goto mangle\_PRE\_public

* For packets on interface ens160, it jumps to zone-specific rules like mangle\_PRE\_public.

**🌐 NAT Chains (for IP translation)**

**nat\_PREROUTING**

type nat hook prerouting priority dstnat + 10; policy accept;

* Catches incoming packets **before destination translation** (DNAT).

**nat\_POSTROUTING**

type nat hook postrouting priority srcnat + 10; policy accept;

* For **source NAT (SNAT)** — modifies source IP after routing decision.

**nat\_OUTPUT**

type nat hook output priority dstnat + 10; policy accept;

* For locally generated packets that may need NAT.

**🔒 Filter Chains (packet filtering – allow/deny)**

**filter\_PREROUTING**

type filter hook prerouting priority filter + 10; policy accept;

icmpv6 type { nd-router-advert, nd-neighbor-solicit } accept

meta nfproto ipv6 fib saddr . mark . iif oif missing drop

* Filters packets **before routing**.
* Allows important IPv6 Neighbor Discovery ICMP types.
* Drops some IPv6 packets based on routing FIB and mark.

**filter\_INPUT**

type filter hook input priority filter + 10; policy accept;

ct state { established, related } accept

ct status dnat accept

iifname "lo" accept

ct state invalid drop

* For **incoming packets**.
* Allows:
  + Existing connections.
  + DNAT'ed packets.
  + Loopback interface.
* Drops **invalid connections**.
* Then jumps to filter\_INPUT\_ZONES (part of firewalld logic – appears cut off in your output).

**🧠 What Is Firewalld Doing Here?**

* Firewalld **dynamically generates** all these chains and rules based on:
  + Zones (like public, internal)
  + Services (like ssh, http)
* It uses multiple hooks for mangle, nat, and filter phases to allow maximum control over packet handling.
* Each \*\_ZONES chain contains the actual **zone-specific rules**.

**✅ Summary**

| **Component** | **Purpose** |
| --- | --- |
| inet firewalld | Table created and managed by firewalld |
| mangle\_\* chains | Packet marking, mostly for QoS or routing decisions |
| nat\_\* chains | Performs DNAT/SNAT (IP address translation) |
| filter\_\* chains | Accept/drop traffic based on rules |
| \*\_ZONES | Dynamically generated zone-based rules by firewalld |

**🛠 To view zone-specific rules managed by firewalld:**

sudo firewall-cmd --list-all --zone=public

Would you like help with **interpreting the full filter chain**, or want to **write custom nftables rules** manually without firewalld?