perplexity

explain deep about iptables table filter, anything technical

The `filter` table in `iptables` is the primary table for packet filtering, determining whether packets are allowed to reach their destination or blocked based on defined rules. It operates at the kernel level via the `netfilter` framework and is central to Linux firewall configurations. Below is a technical breakdown of its components and functionality:

Filter Table Structure

Chains and Netfilter Hooks

The `filter` table contains three built-in chains, each tied to specific `netfilter` hooks that trigger rule evaluation at different stages of packet processing:

- **INPUT**: Processes packets destined for the local system (triggered by `NF_IP_LOCAL_IN`)[1] [2].
- **OUTPUT**: Handles packets originating from the local system (triggered by `NF_IP_LOCAL_OUT`)[1] [2].
- **FORWARD**: Manages packets routed through the system (triggered by `NF_IP_FORWARD`)[1] [2].

Chains are traversed in a fixed order depending on the packet's path (inbound, outbound, or routed).

Default Policies

Each chain has a default policy (`ACCEPT`, `DROP`, or `REJECT`) applied if no rules match the packet:

```
# Example: Set default policy for INPUT chain to DROP
sudo iptables -P INPUT DROP
```

Default policies are visible via `iptables -L`[3] [4].

Stateful Filtering

The `filter` table is stateful, leveraging connection tracking (`conntrack`) to classify packets into states:

- **NEW**: Initial packet of a connection.
- **ESTABLISHED**: Packets part of an existing connection.

- **RELATED**: Packets related to but not part of an existing connection (e.g., FTP data channels).

Example rule allowing established SSH connections:

```
sudo iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT
```

This ensures only responses to outgoing SSH connections are permitted[1] [5].

Rule Criteria and Targets

Match Criteria

Rules in the `filter` table evaluate packets based on:

- **Source/Destination IPs**: `-s 192.168.1.0/24` or `-d 10.0.0.1`.
- **Protocols**: `-p tcp`, `-p udp`, or `-p icmp`.
- **Ports**: `--dport 80` (destination port) or `--sport 22` (source port).
- **Interfaces**: `-i eth0` (input interface) or `-o eth1` (output interface).

Targets

- **ACCEPT**: Allows the packet to proceed.
- **DROP**: Silently discards the packet.
- **REJECT**: Discards the packet and sends an ICMP error (e.g., `--reject-with icmp-port-unreachable`).
- **LOG**: Logs the packet (via `--log-prefix "Firewall: "`) without affecting traversal[5] [2].

Integration with Other Tables

While the `filter` table handles packet filtering, it interacts with other tables:

1. **Mangle Table**: Alters packet headers (e.g., TTL, marks) before filtering[1] [2].
2. **NAT Table**: Modifies source/destination addresses (e.g., masquerading) before packets reach the `filter` table[1] [2].

Example Rule Workflow

For an incoming HTTP request:

1. **PREROUTING** (nat table): No address translation.
2. **INPUT** (filter table):
   - Rule 1: `-p tcp --dport 80 -j ACCEPT` → Allows HTTP traffic.

- Default Policy: `DROP` → Blocks all other traffic.

If no rules match, the packet is dropped[2] [6].

Command Syntax

- List rules with counters and line numbers:

```
sudo iptables -vnL --line-numbers
```

- Delete a rule by line number:

```
sudo iptables -D INPUT 3
```

- Flush all rules:

```
sudo iptables -F
```

The `filter` table's flexibility and integration with Linux networking internals make it indispensable for configuring firewalls, enforcing security policies, and managing traffic flow. Mastery of its stateful capabilities and interaction with other tables is critical for advanced network configurations.

⁂

1. https://www.digitalocean.com/community/tutorials/a-deep-dive-into-iptables-and-netfilter-architecture
2. https://www.novell.com/documentation/opensuse110/opensuse110_reference/data/sec_fire_iptables.html?view=print
3. https://docs.vultr.com/how-to-set-up-firewall-policies-using-iptables
4. https://lowendbox.com/blog/iptables-tutorial-delete-insert-rules/
5. https://www.mankier.com/8/iptables
6. http://www.novell.com/documentation/opensuse103/opensuse103_reference/data/sec_fire_iptables.html