Applying a Filter to an Interface

To apply firewall filters to an interface, include the filter statement:

```
filter {
    group filter-group-number;
    input filter-name;
    input-list [ filter-names ];
    output filter-name;
    output-list [ filter-names ];
}
```

To apply a single filter, include the input statement:

```
filter {
    input filter-name;
}
```

To apply a list of filters to evaluate packets received on an interface, include the input-list statement.

```
filter {
    input-list [ filter-names ];
}
```

Up to 16 filter names can be included in an input list.

To apply a list of filters to evaluate packets transmitted on an interface, include the output-list statement.

```
filter {
    output-list [ filter-names ];
}
```

When you apply filters using the input-list statement or the output-list statement, a new filter is created with the name *<interface-name>.<unit-direction>*. This filter is exclusively interface-specific.

You can include these statements at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family *family*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family *family*]

In the family statement, the protocol family can be ccc, inet, inet6, mpls, or vpls.

In the group statement, specify the interface group number to associate with the filter.

In the input statement, list the name of one firewall filter to be evaluated when packets are received on the interface.

In the input-list statement, list the names of filters to evaluate when packets are received on the interface. You can include up to 16 filter names.

In the output statement, list the name of one firewall filter to be evaluated when packets are transmitted on the interface.

NOTE: Output filters do not work for broadcast and multicast traffic, including VPLS traffic, as shown in "Example: Applying a Filter to an Interface" on page 3.

NOTE: On an MX-series router, you cannot apply as an output filter, a firewall filter configured at the [edit firewall filter family ccc] hierarchy level. Firewall filters configured for the family ccc statement can be applied only as input filters.

In the output-list statement, list the names of filters to evaluate when packets are transmitted on the interface. You can include up to 16 filter names.

You can use the same filter one or more times. On M-series platforms (except the M320 and M120 routers), if you apply a firewall filter or policer to multiple interfaces, the filter or policer acts on the sum of traffic entering or exiting those interfaces.

On T-series, M120, and M320 platforms, interfaces are distributed among multiple packet forwarding components. Therefore, on these platforms, if you apply a firewall filter or policer to multiple interfaces, the filter or policer acts on the traffic stream entering or exiting each interface, regardless of the sum of traffic on the multiple interfaces.

If you apply the filter to the interface lo0, it is applied to packets received or transmitted by the Routing Engine. You cannot apply MPLS filters to the management interface (fxp0) or the loopback interface (lo0).

For more information about firewall filters, see the *JUNOS Policy Framework Configuration Guide*. For more information about MPLS filters, see the *JUNOS MPLS Applications Configuration Guide*.

See also the following sections:

- Defining Interface Groups in Firewall Filters on page 2
- Filter-Based Forwarding on the Output Interface on page 3
- Example: Applying a Filter to an Interface on page 3

Defining Interface Groups in Firewall Filters

When applying a firewall filter, you can define an interface to be part of an *interface group*. Packets received on that interface are tagged as being part of the group. You can then match these packets using the interface-group match statement, as described in the *JUNOS Policy Framework Configuration Guide*.

To define the interface to be part of an interface group, include the group statement:

group *filter-group-number*;

You can include this statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family *family* filter]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family *family* filter]

Filter-Based Forwarding on the Output Interface

If port-mirrored packets are to be distributed to multiple monitoring or collection interfaces, based on patterns in packet headers, it is helpful to configure a filter-based forwarding (FBF) filter on the port-mirroring egress interface.

When an FBF filter is installed as an output filter, a packet that is forwarded to the filter has already undergone at least one route lookup. After the packet is classified at the egress interface by the FBF filter, it is redirected to another routing table for additional route lookup. To avoid packet looping inside the Packet Forwarding Engine, the route lookup in the latter routing table (designated by an FBF routing instance) must result in a different next hop from any next hop specified in a table that has already been applied to the packet.

If an input interface is configured for FBF, the source lookup is disabled for those packets headings to a different routing instance, since the routing table is not set up to handle the source lookup.

For more information about FBF configuration, see the *JUNOS Routing Protocols Configuration Guide*. For more information about port mirroring, see the *JUNOS Services Interfaces Configuration Guide*.

Example: Applying a Filter to an Interface

Input Filter for VPLS Traffic

For M-series and T-series routing platforms only, apply an input filter to VPLS traffic. Output filters do not work for broadcast and multicast traffic, including VPLS traffic.

```
[edit interfaces]
fe-2/2/3 {
   vlan-tagging;
   encapsulation vlan-vpls;
   unit 601 {
      encapsulation vlan-vpls;
      vlan-id 601;
      family vpls {
         filter {
            input filter1;  # Works for multicast destination MAC address
            output filter1;  # Does not work for multicast destination MAC address
         }
      }
   }
}
[edit firewall]
family vpls {
   filter filter1 {
      term 1 {
```

```
            from {
                destination-mac-address {
                    01:00:0c:cc:cc:cd/48;
                }
            }
            then {
                discard;
            }
        }
        term 2 {
            then {
                accept;
            }
        }
    }
}
```

Filter-Based Forwarding at the Output Interface

The following example illustrates the configuration of filter-based forwarding at the output interface. In this example, the packet flow follows this path:

1. A packet arrives at interface fe-1/2/0.0 with source and destination addresses 10.50.200.1 and 10.50.100.1 respectively.
2. The route lookup in routing table inet.0 points to the egress interface so-0/0/3.0.
3. The output filter installed at so-0/0/3.0 redirects the packet to routing table fbf.inet.0.
4. The packet matches the entry 10.50.100.0/25 in the fbf.inet.0 table, and finally leaves the router from interface so-2/0/0.0.

```
[edit interfaces]
so-0/0/3 {
    unit 0 {
        family inet {
            filter {
                output fbf;
            }
            address 10.50.10.2/25;
        }
    }
}
fe-1/2/0 {
    unit 0 {
        family inet {
            address 10.50.50.2/25;
        }
    }
}
so-2/0/0 {
    unit 0 {
        family inet {
            address 10.50.20.2/25;
        }
    }
}
```

```
[edit firewall]
filter fbf {
   term 0 {
      from {
         source-address {
            10.50.200.0/25;
         }
      }
      then routing-instance fbf;
   }
   term d {
      then count d;
   }
}
[edit routing-instances]
fbf {
   instance-type forwarding;
   routing-options {
      static {
         route 10.50.100.0/25 next-hop so-2/0/0.0;
      }
   }
}
[edit routing-options]
interface-routes {
   rib-group inet fbf-group;
}
static {
   route 10.50.100.0/25 next-hop 10.50.10.1;
}
rib-groups {
   fbf-group {
      import-rib [inet.0 fbf.inet.0];
   }
}
```