- Can be configured to accept or discard a packet before it enters or exits a port or interface

- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

Reason to use a Routing Policy

- Control which routes a protocol imports into the routing table

- Control which routes a protocol exports from the routing table

- Use a routing protocol to announce active routes learned from another routing protocol, which is sometimes called route redistribution

- Manipulate route characteristics such as the preference value

- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

Default Routing Policies

Default Policy for RIP

- Default import policy – accept all RIP routes learned from configured neighbors and import into the `inet.0` routing table
- Default export policy – reject everything, to export RIP routes an export policy must be configured

Default Policy for OSPF

- Default import policy – accept all OSPF routes and import into the `inet.0` routing table
- Default export policy – reject everything, OSPF does not export internally learned routes (directly connected routes on interfaces running OSPF)

Default Policy for IS-IS

- Default import policy – accept all IS-IS routes and import into the `inet.0` and `inet6.0` routing table
- Default export policy – reject everything

- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

Default Policy for BGP

- Default import policy – accept all received BGP IPv4 and IPv6 routes learned from configured neighbors and import into the `inet.0` and `inet6.0` routing table
- Default export policy – readvertise all active BGP routes

Building Blocks of Routing Policies

Policy Components

Terms:
- named structures in which match conditions and actions are defined
- one or more terms can be defined

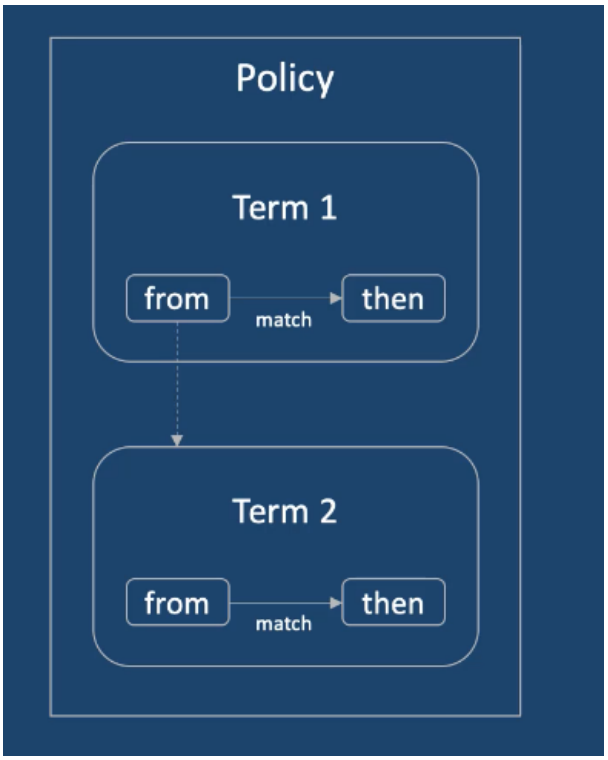
Match Conditions:
- criteria against which a route or packet is compared
- one or more criteria can be configured
- **if all criteria match, one or more actions are applied**

Actions:
- what happens if all criteria match
- one or more actions can be configured

Terms are basic building blocks of all Junos policies
Essentially, they are if..then statements
If all the match conditions specified in the <u>from</u> statement are true, all the actions in the <u>then</u> statement are executed.

- Can be configured to accept or discard a packet before it enters or exits a port or interface

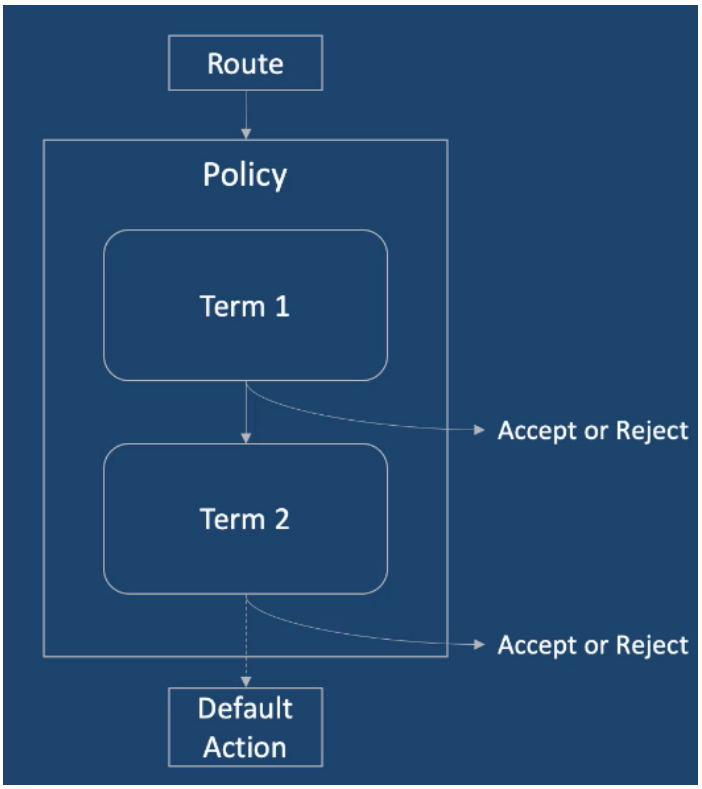
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks



- Can be configured to accept or discard a packet before it enters or exits a port or interface

- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

Routing Policy Evaluation




```
policy-statement my-policy {
   term accept-direct-routes {
      from {
         protocol direct;
         interface ge-0/0/0;
      }
      then accept;
   }
   term reject-rip-routes {
      from protocol rip;
      then reject;
   }
}
```

- Can be configured to accept or discard a packet before it enters or exits a port or interface

- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

Routing Policy Match Conditions

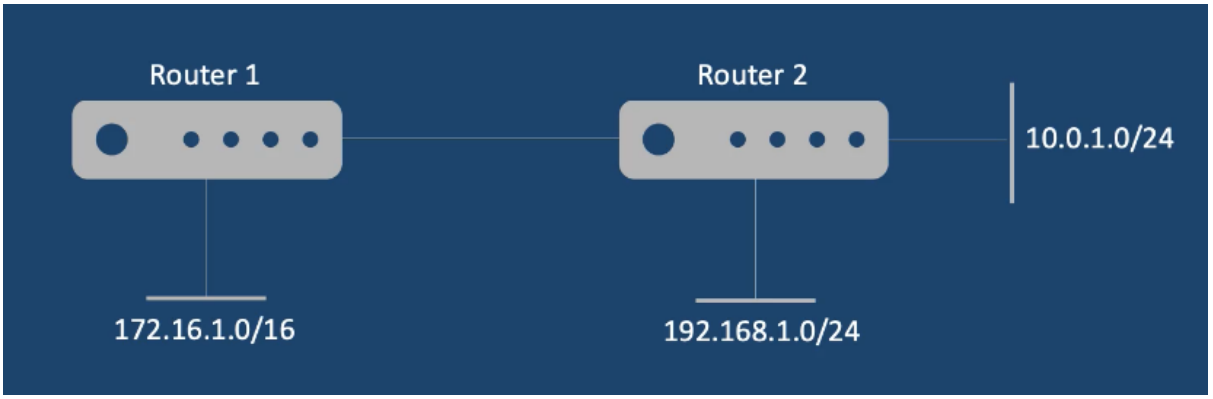
- If omitted the from/to statement, all routes are considered to be a match
- The then statement may include one or more actions

Three types of actions:

- **Flow control actions** - these decide whether to accept or reject the route and whether to evaluate the next term or routing policy
- Actions that **manipulate route characteristic**
- **Trace action** - this logs route matches

- The `then` statement is optional, if omitted one of the following occurs:
  - the next term in the routing policy, if one present, is evaluated
  - if there are no more terms in the routing policy, the next routing policy, if one present is evaluated
  - if there are no more terms or routing policies, the accept or reject action specified by the default policy is taken

Routing Policy Example



- Can be configured to accept or discard a packet before it enters or exits a port or interface

- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

Configure to accept 192.168.1.0/24 for Router 1

```
user@Router1# show policy-options
policy-statement rip-import {
    term 1 {
        from {
            protocol rip;
            route-filter 192.168.0.0/16 orlonger;
        }
        then accept;
    term 2 {
        then reject;
    }
}
```

Terminal Example

edit policy-options
edit ?

```
Possible completions:
> application-maps     Define application maps
> as-path              BGP autonomous system path regular expression
> as-path-group        Group a set of AS paths
> community            BGP community information
> condition            Define a route advertisement condition
> damping              BGP route flap damping properties
> defaults             Policy default behaviour
> mac-list             Define a named set of mac addresses
> policy-statement     Routing policy
> prefix-list          Define a named set of address prefixes
> route-distinguisher  Route-distinguisher information
> route-filter-list    Define a named set of route-filter address prefixes
> rtf-prefix-list      Define a named set of family route target prefixes
> satellite-policies   Satellite Policy configuration
> source-address-filter-list  Define a named set of source address filter address prefixes
> vsi-policy           Define a named set of VSI policies
```

edit policy-statment RIP-POLICY
edit term Term1
set from protocol rip
set from interface ge-0/0/0
set then accept

- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

up
up
show

```
[edit policy-options]
root@SRX# show
policy-statement RIP-POLICY {
    term Term1 {
        from {
            protocol rip;
            interface ge-0/0/0.0;
        }
        then accept;
    }
}
```

Prefix List

Example

- Named list of IP addresses used to match routes
- Configured under the `[edit policy-options]` hierarchy
- Can be referenced in multiple terms within a single policy or in different policies
- Can be used with routing policies and firewall filters

- Can be configured to accept or discard a packet before it enters or exits a port or interface

- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

```
prefix-list rfc1918 {
    10.0.0.0/8;
    172.16.0.0/12;
    192.168.0.0/16;
}
policy-statement my-policy {
    term reject-rfc1918 {
        from {
            prefix-list rfc1918;
        }
        then reject;
    }
}
```

edit policy-options
edit prefix-list RFC1918
set 10.0.0.0/8
set 172.16.0.0/12
set 192.168.0.0/16
up
show

```
root@SRX# show
prefix-list RFC1918 {
    10.0.0.0/8;
    172.16.0.0/12;
    192.168.0.0/16;
}
```

- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

Prefix List Filter

- With `prefix-list-filter`, you can specify a match type of `exact`, `longer`, or `orlonger` on the listed prefixes
- You can specify an optional action to be taken if the filter matches
- The action is executed immediately after the match occurs, and the `then` statement is not evaluated

Example

```
prefix-list rfc1918 {
    10.0.0.0/8;
    172.16.0.0/12;
    192.168.0.0/16;
}
policy-statement my-policy {
    term reject-rfc1918 {
        from {
            prefix-list-filter rfc1918 orlonger reject;
        }
    }
}
```

So here we've specified or longer and we've specified the action along with the match conditions

```
root@SRX# set policy-statement my-policy from prefix-list-filter RFC1918 longer
```

- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

Route Filters

- List of prefixes configured within a single routing policy or policy term
- Unlike prefix lists, these are not reusable but rather are specific to the policy or term in which they're configured
- Like with `prefix-list-filter` statement, an optional action can be specified to be taken if the `route-filter` statement matches

Match Types

- exact
- orlonger
- longer
- upto
- prefix-length-range

Exact

```
from route-filter 172.16.0.0/16 exact;
```

- Only routes that match the given prefix exactly are considered to be a match
- In the above example, only `172.16.0.0/16` can match

- Can be configured to accept or discard a packet before it enters or exits a port or interface

- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

```
policy-statement my-policy {
    term term-A {
        from {
            route-filter 172.16.0.0/16 exact accept;
        }
    }
}
```

Orlonger

```
from route-filter 172.16.0.0/16 orlonger;
```

- Only routes within the specified prefix, with prefix length equal to or greater than the given prefix length are considered to be a match

- Here, `172.16.0.0/16` is an exact match

- Also routes within the subset `172.16.0.0/16` with a prefix length between `/17` and `/32` are considered to be a match

- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

```
from route-filter 172.16.0.0/16 orlonger;
```

- Matches: `172.16.0.0/17, 172.16.32.0/20, 172.16.38.0/24, 172.16.12.128/26, 172.16.1.192/29`
- Not Matches: `172.16.0.0/15, 172.17.0.0/16, 192.168.0.0/24, 10.0.0.0/8`

Longer

```
from route-filter 172.16.0.0/16 longer;
```

- Only routes within the specified prefix, with prefix length greater than the given prefix length are considered to be a match
- Here, `172.16.0.0/16` is **not** a match
- Routes within the subset `172.16.0.0/16` with a prefix length between `/17` and `/32` are considered to be a match

- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

```
from route-filter 172.16.0.0/16 longer;
```

- Matches: `172.16.0.0/17, 172.16.32.0/20, 172.16.38.0/24, 172.16.12.128/26, 172.16.1.192/29`
- Not Matches: `172.16.0.0/16, 172.16.0.0/15, 172.17.0.0/16, 192.168.0.0/24, 10.0.0.0/8`

Upto

```
from route-filter 172.16.0.0/16 upto /24;
```

- The `upto` match type is similar to `orlonger` match type, except that it provides an upper limit to the acceptable prefix length
- Only routes within the specified prefix, with prefix length greater than or equal to the given prefix length, but less than or equal to the `upto` prefix length, are considered to be a match

- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

```
from route-filter 172.16.0.0/16 upto /24;
```

- Matches: `172.16.0.0/17, 172.16.128.0/17, 172.16.32.0/20, 172.16.38.0/24`
- Not Matches: `172.16.12.128/26, 172.16.1.192/29, 172.16.0.0/15, 172.17.0.0/16`

Prefix-length-range

```
from route-filter 172.16.0.0/16 prefix-length-range /20-/24;
```

- The `prefix-length-range` match type is similar to the `upto` match type, except that it provides both a lower and an upper limit to the acceptable prefix length
- Only routes within the specified prefix, with prefix length greater than or equal to the first given prefix length, but less than or equal to the second prefix length, are considered to be a match

- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

```
from route-filter 172.16.0.0/16 prefix-length-range /20-/24;
```

- Matches: `172.16.32.0/20, 172.16.38.0/24`
- Not Matches: `172.16.0.0/16, 172.16.0.0/17, 172.16.12.128/26, 172.16.1.192/29, 172.17.0.0/16`

Applying Routing Policies

- Depending on the routing protocol, you can apply import and export policies at multiple levels of the hierarchy
- RIP import policies can be applied at either the global, group or the neighbor level – this will affect routes from either all peers or a specific neighbor
- RIP export policies may only be applied at the group level, allowing you to alter routing knowledge for a specific set of peers only

- BGP import and export policies can be applied at the global, group or neighbor level
- OSPF allows only protocol-level import and export policies

- Can be configured to accept or discard a packet before it enters or exits a port or interface

- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

```
protocols {
    rip {
        import imp-policy;
        group my-group {
            import imp-policy;
            export exp-policy;
            neighbor ge-0/0/2.0 {
                import imp-policy;
            }
        }
    }
}
```

```
bgp {
    import imp-policy;
    export exp-policy;
    group my-group {
        import imp-policy;
        export exp-policy;
        neighbor 1.1.1.1 {
                import imp-policy;
                export exp-policy;
        }
    }
}
```

- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

```
ospf {
    import imp-policy;
    export exp-policy;
    area 0.0.0.0 {
        interface ge-0/0/0.0;
    }
}
```

Firewall Filters

- Provide rules that define whether to accept or discard packets that are transiting an interface
- If a packet is accepted, actions such as class-of-service and traffic policing can be performed
- Also referred to as Access Control Lists on other vendors' equipment

- Stateless in nature, so each packet is examined individually
- Packet contents are evaluated statically, and it does not keep track of the state of network connections

- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

Firewall Filter Components

Actions

- Terminating actions
- Nonterminating action
- Flow control action

Terminating Action

- Stops evaluation of a firewall filter for a specific packet
- Specified action is performed, no additional terms are examined
- Examples include `accept`, `discard`, and `reject`

- `accept` – causes the system to accept the packet
- `discard` – causes the system to silently discard the packet, without sending an ICMP message to the source address
- `reject` – causes the system to discard the packet and send an ICMP message back to the source address

- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

Nonterminating Action

- Used to perform functions such as incrementing a counter, logging information about the packet header, sampling the data, or sending information to a remote host
- Examples include `count`, `log`, `policer`, or `syslog`

- `count` – count the packet
- `log` – log the packet header information
- `policer` – use a policer to rate limit traffic
- `syslog` – log the packet to the system log file

- Using a nonterminating action without an explicit terminating action results in a default terminating action of `accept`
- To prevent the firewall filter action from terminating, use the `next term` action after the nonterminating action

- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

Flow Control Action

- Allows the device to perform configured actions on the packet and then evaluate the next term in the filter, rather than terminating the filter
- This is the `next term` action

Configuring Firewall Filters

Applying Firewall Filters

- Fireware filters can be applied to all interfaces to filter traffic entering or exiting them
- Can also be applied to lo0 interface to filter traffic destined for the system
- An IPv6 filter cannot be applied to an IPv4 interface - the protocol family of the firewall filter and interface must match

```
interfaces ge-0/0/1 {
    unit 0 {
        family inet {
            filter {
                input filter-in;
                output filter-out;
            }
        }
    }
}
```

Configure firewall filter to discard ICMP traffic

edit

- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

edit firewall
edit filter BLOCK-ICMP-TELNET
edit term BLOCK-ICMP
set from protocol icmp
set then discard
set then log
set then count
edit term ALLOW-ALL //must explicitly define other traffic to be accepted
set then accept
up
commit
show

```
[edit firewall filter BLOCK-ICMP-TELNET]
root@SRX# show
term BLOCK-ICMP {
    from {
        protocol icmp;
    }
    then {
        count BLOCK-ICMP-COUNTER;
        log;
        discard;
    }
}
term ALLOW-ALL {
    then accept;
}
```

top
edit interfaces lo0 unit 0 family inet //apply to loopback interface
set filter input BLOCK-ICMP-TELNET
commit
> show firewall log

Configure firewall filter to deny telnet traffic

edit firewall filter BLOCK-ICMP-TELNET
edit term BLOCK-TELNET
set from port telnet
set from protocol tcp

- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

set then reject
set then log
set then syslog
show

```
[edit firewall filter BLOCK-ICMP-TELNET term BLOCK-TELNET]
root@SRX# show
from {
    protocol tcp;
    port telnet;
}
then {
    log;
    syslog;
    reject;
}
```

up
show

```
[edit firewall filter BLOCK-ICMP-TELNET]
root@SRX# show
term BLOCK-ICMP {
    from {
        protocol icmp;
    }
    then {
        count BLOCK-ICMP-COUNTER;
        log;
        reject;
    }
}
term ALLOW-ALL {
    then accept;
}
term BLOCK-TELNET {
    from {
        protocol tcp;
        port telnet;
    }
    then {
        log;
        syslog;
        reject;
    }
}
```

insert term BLOCK-TELNET before term ALLOW-ALL
show

- Can be configured to accept or discard a packet before it enters or exits a port or interface

- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

```
[edit firewall filter BLOCK-ICMP-TELNET]
root@SRX# show
term BLOCK-ICMP {
    from {
        protocol icmp;
    }
    then {
        count BLOCK-ICMP-COUNTER;
        log;
        reject;
    }
}
term BLOCK-TELNET {
    from {
        protocol tcp;
        port telnet;
    }
    then {
        log;
        syslog;
        reject;
    }
}
term ALLOW-ALL {
    then accept;
}
```

Traffic Policing

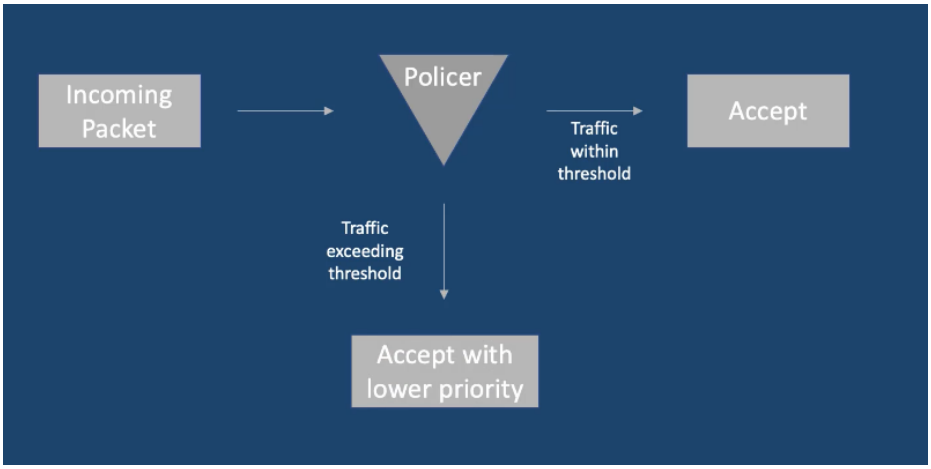
- Enables you to control the max rate of traffic sent or received on an interface
- Also known as rate limiting, it is designed to thwart DoS attack
- Can be applied to inbound or outbound traffic
  - inbound: allows you to conserve resources by dropping traffic
  - outbound: allows you to control the bandwidth that is being used

**

Traffic Policing employs a token-bucket algorithm, which enforces a limit on the average bandwidth while allowing bursts up to a specified maximum value

- Can be configured to accept or discard a packet before it enters or exits a port or interface

- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks



- Rate limits:
  - bandwidth – number of bits per second permitted on average
  - burst size – total number of bytes the system allows during a burst

- Can be configured to accept or discard a packet before it enters or exits a port or interface

- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

```
firewall {
    policer DROP-EXCESS {
        if-exceeding {
            bandwidth-limit 2m;
            burst-size-limit 5k;
        }
        then discard;
    }
}
```

```
firewall {
    term FROM-TRUST {
        from {
            source-address {
                10.0.0.0/24 ;
            }
        then {
            policer DROP-EXCESS;
            accept;
        }
    }
}
```

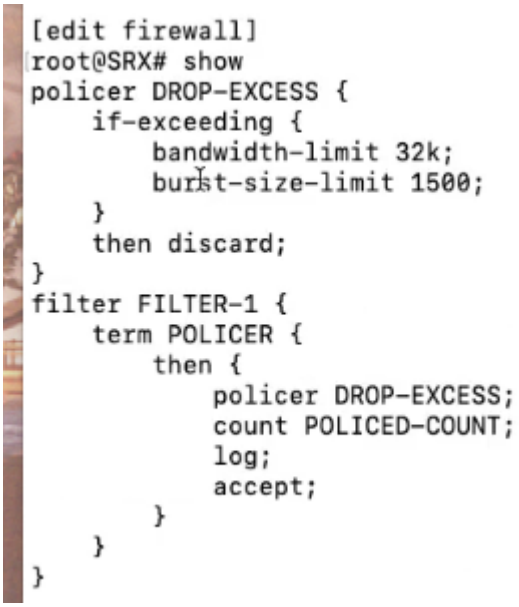
edit firewall
edit policer DROP-EXCESS
set if-exceeding bandwidth-limit 32000 //bits per second (32000 - 5000000…)
set if-exceeding burst-size-limit 150 //bytes (1500- …)

- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

set then discard
up
show

```
[edit firewall]
root@SRX# show
policer DROP-EXCESS {
    if-exceeding {
        bandwidth-limit 32k;
        burst-size-limit 1500;
    }
    then discard;
}
```

edit filter Filter-1
edit term POLICER
set then policer DROP-EXCESS
set then accept
set then log
set then count POLICED-COUNT
up
show



```
[edit firewall]
root@SRX# show
policer DROP-EXCESS {
    if-exceeding {
        bandwidth-limit 32k;
        burst-size-limit 1500;
    }
    then discard;
}
filter FILTER-1 {
    term POLICER {
        then {
            policer DROP-EXCESS;
            count POLICED-COUNT;
            log;
            accept;
        }
    }
}
```

top

```
root@SRX# set interfaces ge-0/0/1 unit 0 family inet filter input FILTER-1
```

commit

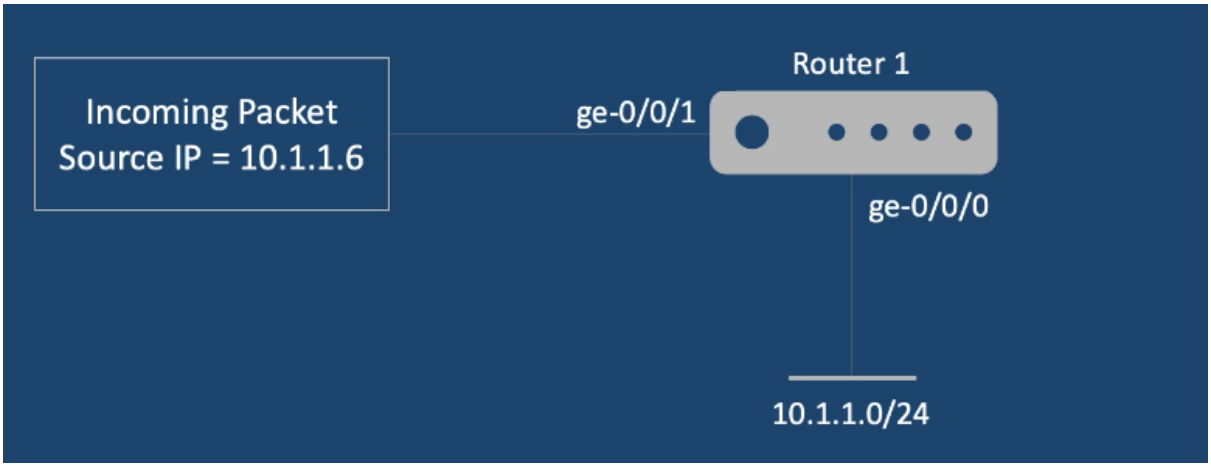
- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

Unicast Reverse Path Forwarding

IP Spoofing

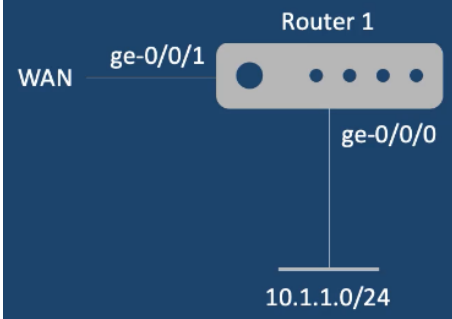
- IP spoofing is a method of attempting to gain access by inserting a false source address in the packet header
- This makes the packet appear as if it's coming from a trusted source

Prevent IP Spoofing on ge-0/0/1 by filter



- Can be configured to accept or discard a packet before it enters or exits a port or interface

- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks




```
firewall {
    filter SPOOF-PREVENTION {
        term DROP-SPOOFED-LAN {
            from {
                source-address 10.1.1.0/24;
            }
            then {
                discard;
            }
        }
        term ALLOW-OTHERS {
            then {
                accept;
            }
        }
    }
}
```

Unicast Reverse Path Forwarding

- Unicast reverse-path-forwarding (RPF) check is a tool to reduce forwarding of IP packets that may be spoofing an address

- It performs a route table lookup on an IP packet's source address and checks the incoming interface

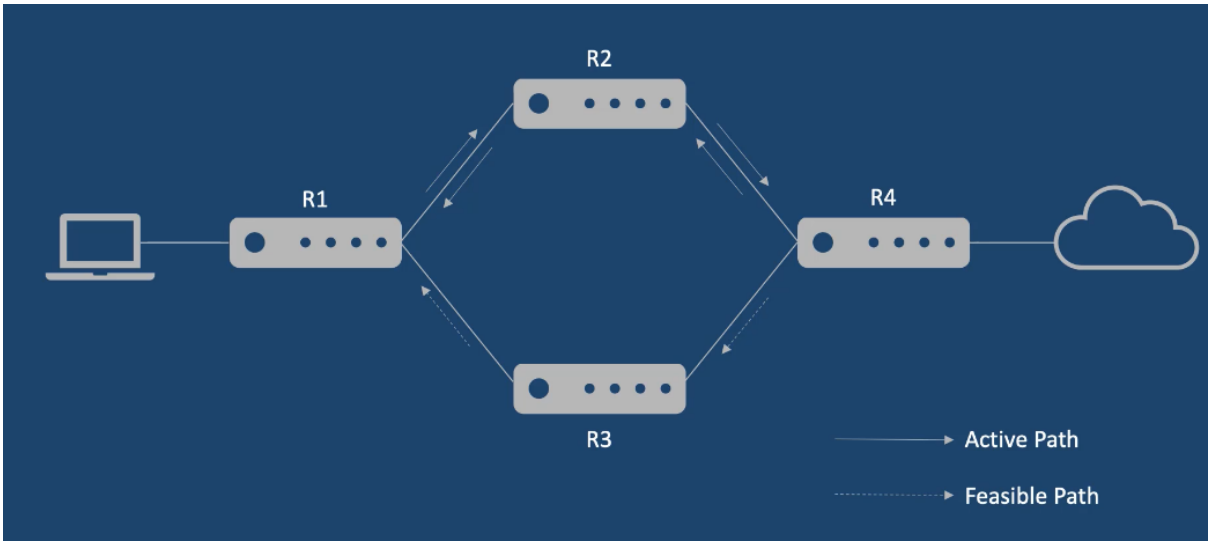
- If the packet is from a valid path, the router forwards the packet to the destination address. Otherwise the router discards the packet

- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

Modes

- Loose mode – the incoming packet's source address must be in the route table
- Strict mode – the incoming packet must be received on the interface that would be used to forward traffic to the source IP address
- Strict mode is the default

Active vs Feasible Path



- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

- By default, when Junos performs its RPF check, it considers only the active routes to a given destination
- In networks where multiple routes exist (different forward and reverse paths), the default behavior of considering only active routes can cause legitimate traffic to be dropped
- To address this, Junos can be configured to consider all feasible routes to a destination when it performs RPF

- In this mode, the system considers all routes it receives to a given destination, even if they are not the active route to the destination
- This option should be activated where the possibility of asymmetric routing exists

Fail Filter

- Allows you to perform additional processing on packets that have failed the unicast RPF check
- Can perform operations such as accepting, rejecting, logging, sampling or policing of packets

Useases

- Can be configured to accept or discard a packet before it enters or exits a port or interface

- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

- Allow packets that would normally fail an RPF check, such as BOOTP packets and DHCP packets – these packets have a source address of 0.0.0.0 and a destination address of 255.255.255.255

- Allow failed packets to be further processed such as logging or counting

```
firewall {
    filter DHCP-BOOTP {
        term ALLOW-DHCP-BOOTP {
            from {
                source-address 0.0.0.0/32;
                }
                destination-address {
                    255.255.255.255/32;
                }
            }
            then accept;
        }
    }
}
```

- Can be configured to accept or discard a packet before it enters or exits a port or interface

- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

Example

Setup Fail Filter

```
root@SRX# edit firewall

[edit firewall]
root@SRX# edit filter DHCP-BOOTP

[edit firewall filter DHCP-BOOTP]
root@SRX# edit term ALLOW-DHCP-BOOTP

[edit firewall filter DHCP-BOOTP term ALLOW-DHCP-BOOTP]
root@SRX# set from source-address 0.0.0.0/32

[edit firewall filter DHCP-BOOTP term ALLOW-DHCP-BOOTP]
root@SRX# set from destination-address 255.255.255.255/32

[edit firewall filter DHCP-BOOTP term ALLOW-DHCP-BOOTP]
root@SRX# show
from {
    source-address {
        0.0.0.0/32;
    }
    destination-address {
        255.255.255.255/32;
    }
}

[edit firewall filter DHCP-BOOTP term ALLOW-DHCP-BOOTP]
root@SRX# set then accept

[edit firewall filter DHCP-BOOTP term ALLOW-DHCP-BOOTP]
root@SRX# set then count ?
Possible completions:
  <count>              Count the packet in the named counter
[edit firewall filter DHCP-BOOTP term ALLOW-DHCP-BOOTP]
root@SRX# set then count DHCP-BOOTP-COUNTER

[edit firewall filter DHCP-BOOTP term ALLOW-DHCP-BOOTP]
root@SRX# up
```

- Can be configured to accept or discard a packet before it enters or exits a port or interface

- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

```
[edit firewall filter DHCP-BOOTP]
root@SRX# edit term DEFAULT

[edit firewall filter DHCP-BOOTP term DEFAULT]
root@SRX# set then reject;
                  ^
syntax error.

[edit firewall filter DHCP-BOOTP term DEFAULT]
root@SRX# set then reject

[edit firewall filter DHCP-BOOTP term DEFAULT]
root@SRX# set then log

[edit firewall filter DHCP-BOOTP term DEFAULT]
root@SRX# up

[edit firewall filter DHCP-BOOTP]
root@SRX# show
term ALLOW-DHCP-BOOTP {
    from {
        source-address {
            0.0.0.0/32;
        }
        destination-address {
            255.255.255.255/32;
        }
    }
    then {
        count DHCP-BOOTP-COUNTER;
        accept;
    }
}
term DEFAULT {
    then {
        log;
        reject;
    }
}

[edit firewall filter DHCP-BOOTP]
root@SRX#
```

- Can be configured to accept or discard a packet before it enters or exits a port or interface
- Can be used to do the following:
  - restrict traffic destined for the Routing Engine based on its source, protocol and application
  - limit the rate of packets destined for the Routing Engine to protect against flood or denial-of-service (DoS) attacks

Apply fail filter with RPF-check

```
[edit]
root@SRX# set interfaces ge-0/0/1 unit 0 family inet rpf-check ?
Possible completions:
  <[Enter]>            Execute this command
+ apply-groups         Groups from which to inherit configuration data
+ apply-groups-except  Don't inherit configuration data from these groups
  fail-filter          Name of filter applied to packets failing RPF check
> mode                 Mode for reverse path forwarding
  |                    Pipe through a command
[edit]
root@SRX# set interfaces ge-0/0/1 unit 0 family inet rpf-check fail-filter ?
Possible completions:
  <fail-filter>        Name of filter applied to packets failing RPF check
  DHCP-BOOTP           [firewall filter]
[edit]
root@SRX# set interfaces ge-0/0/1 unit 0 family inet rpf-check fail-filter DHCP-BOOTP
```

to enable feasible path

```
root@SRX# set routing-options forwarding-table unicast-reverse-path ?
Possible completions:
  active-paths         Consider active paths when performing RP verification
  feasible-paths       Consider all feasible paths for RP verification
[edit]
root@SRX# set routing-options forwarding-table unicast-reverse-path feasible-paths
```