Firewall Configuration Guide, 17.2.0

November 2017

Supporting AT&T Vyatta Network Operating System

Contents

About This Guide 9

Firewall Overview 10

Firewall functionality 10

Firewall and fragmented packets 10

Defining firewall instances 10

Firewall rules 10

Implicit drop 11

Exclusion rules 11

Stateful firewall and connection tracking 11

TCP strict tracking 11

Applying firewall instances to interfaces 12

Interaction between firewall, NAT, and routing 12

Traffic flow through firewall, NAT, and routing 12

Zone-based firewall 13

Control plane protection 14

Firewall denial of service protection 15

Session and packet logging 16

Application aware firewall 17

Configuration Examples 19

Packet-filtering 19
Filtering on source IP address 19
Filtering on source and destination IP addresses 20
Filtering on source IP address and destination protocol 21
Defining a network-to-network filter 22
Filtering on source MAC address 23
Excluding an address 24
Matching TCP flags 25
Matching ICMP type names 26
Matching groups 27
Stateful behavior 29
Configuring stateful behavior per rule set 29
Configuring global state policies 30
Changes in global-state-policy behavior 32
Using firewall with VRRP interfaces 33
Applying a rule set to a VRRP interface 33
Using VRRP with a zone-based firewall 35
Enabling firewall denial of service protection 35
Viewing firewall information 38
Showing firewall instance information 38
Showing firewall configuration 38

Global Firewall Commands ... 40
clear firewall ... 40
show security firewall ... 40
Firewall Commands ... 41
clear session policy ... 41
interfaces dataplane interface firewall local ruleset ... 41
interfaces firewall ... 42
show log firewall ... 43
security application firewall name <name> description <text> ... 43
security application firewall name <name> no-match-action <action> ... 44
security application firewall name <name> rule <rule-number> ... 45
security application firewall name <name> rule <rule-number> action <action> ... 45
security application firewall name <name> rule <rule-number> description <text> ... 46
security application firewall name <name> rule <rule-number> name <name> ... 47
security application firewall name <name> rule <rule-number> protocol <protocol> ... 48
security application firewall name <name> rule <rule-number> type <type> ... 49
security firewall all-ping ... 49
security firewall broadcast-ping ... 50
security firewall config-trap ... 51
security firewall global-state-policy ... 51
security firewall name default-action ... 52
security firewall name default-log ... 53

security firewall name description ... 54
security firewall name rule ... 54
security firewall name rule action ... 55
security firewall name rule description ... 56
security firewall name rule destination ... 56
security firewall name rule disable ... 58
security firewall name rule dscp ... 58
security firewall name rule ethertype ... 59
security firewall name rule fragment ... 60
security firewall name rule icmp ... 61
security firewall name rule icmpv6 ... 61
security firewall name rule ipv6-route type ... 62
security firewall name rule log ... 63
security firewall name rule mark ... 64
security firewall name rule pcp ... 65
security firewall name rule police ... 66
security firewall name rule protocol ... 67
security firewall name <name> rule <rule-number> session application firewall <app-firewall> ... 68
security firewall name <name> rule <rule-number> session application name <name> ... 68
security firewall name <name> rule <rule-number> session application protocol <protocol> ... 69
security firewall name <name> rule <rule-number> session type <type> ... 70

security firewall name rule source ... 71
security firewall name rule state ... 72
security firewall name rule tcp flags ... 73
security firewall session-log ... 74
security firewall syn-cookies ... 75
security firewall tcp-strict ... 76
monitor firewall ... 76
show session limit group ... 77
show session limit parameter ... 78
show session policy brief ... 79
set system session plocy name <name> max-halfopen action ... 80
system session limit global rate-limit ... 81
system session limit group name interface ... 81
system session limit group name rule destination ... 82
system session limit group name rule icmp ... 83
system session limit group name rule icmpv6 ... 84
system session limit group name rule parameter ... 86
system session limit group name rule protocol ... 86
system session limit group name rule source ... 87
system session limit group name rule tcp flags ... 88
system session limit parameter name max-halfopen ... 88
system session limit parameter name rate-limit ... 89

Zone-Based Firewall Commands ... 91
clear zone-policy ... 91
show zone-policy ... 91
security zone-policy zone ... 92
security zone-policy zone default-action ... 92
security zone-policy zone description ... 93
security zone-policy zone to ... 94
security zone-policy zone to firewall ... 94
security zone-policy zone interface ... 95
ICMP Types ... 97
ICMPv6 Types ... 100
Supported Interface Types ... 102
List of Acronyms ... 105

Copyright Statement

The training materials and other content provided herein for assistance in training on the Vyatta vRouter may have references to Brocade as the Vyatta vRouter was formerly a Brocade product prior to AT&T's acquisition of Vyatta. Brocade remains a separate company and is not affiliated to AT&T.

About This Guide

This guide describes firewall functionality on the AT&T Vyatta vRouter (referred to as a virtual router, vRouter, or router in the guide).

Firewall Overview

AT&T firewall functionality

Firewall functionality analyzes and filters IP packets between network interfaces. The most common application of functionality is to protect traffic between an internal network and the Internet. It allows you to filter packets based on their characteristics and perform actions on packets that match the rule. The AT&T Vyatta vRouter firewall functionality provides the following features:

- Packet filtering for traffic that traverses the router by using the **`in`** and **`out`** keywords on an interface
- Definable criteria for packet-matching rules, including source IP address, destination IP address, source port, destination port, IP protocol, and Internet Control Message Protocol (ICMP) type
- Ability to set the firewall globally for stateful or stateless operation

The vRouter firewall offers both IPv4 and IPv6 stateful packet inspection to intercept and inspect network activity and to allow or deny the attempt. The advanced firewall capabilities from the vRouter include stateful failover.

Firewall cannot be applied to outbound local traffic. It can only be applied to inbound interface traffic and forwarded outbound traffic.

Firewall and fragmented packets

An input firewall causes fragments to be reassembled. For both IPv4 and IPv6, if the packets arrive on an interface for which firewall is configured, the fragments are reassembled at input before passing to the firewall. If all the fragments of a packet are not received, then the packet is dropped. The reassembled packet passes through the remainder of the forwarding path and firewall does not recognize fragments at either input or output. Passing through an output firewall (or DNAT/SNAT) also results in fragment reassembly before processing by the firewall or NAT rules.

This behavior also applies to a packet arriving on an interface that is assigned to a firewall zone.

Defining firewall instances

Firewalls filter packets on interfaces. Use of the firewall feature has two steps:

1. Define a firewall instance and save it under a name. A firewall instance is also called a firewall rule set, where a rule set is just a series of firewall rules. You define the firewall instance and configure the rules in its rule set in the **firewall** configuration node.
2. Apply the instance to an interface or a zone by configuring the **interface** configuration node for the interface or zone. After the instance is applied to the interface or zone, the rules in the instance begin filtering packets on that location.

Firewall rules

Firewall rules specify the match conditions for traffic and the action to be taken if the match conditions are satisfied. Traffic can be matched on a number of characteristics, including source IP address, destination IP address, source port, destination port, IP protocol, and ICMP type.

Rules are executed in numeric sequence, according to the rule number, from lowest to highest. If the traffic matches the characteristics specified by a rule, the action of the rule is executed; if not, the system “falls through” to the next rule.

> **Note:** You can configure rules to match IPv4 ICMP, IPv6 ICMP, IPv6 routing header, or TCP without specifying the respective protocol, provided that a protocol specific match option is present. For example TCP flags, ICMP type.

The action can be one of the following:

- `Accept`: Traffic is allowed and forwarded.
- `Drop`: Traffic is silently discarded.

To avoid having to renumber firewall rules, a good practice is to number rules in increments of 10. This increment allows room for the insertion of new rules within the rule set.

Implicit Action

When one or more named firewall rules (including the hidden rule used for **default-action** or **default-log**) are applied to an interface and a packet does not match any of the rules in a given direction, then the implicit actions occur. The implicit actions are a property of firewall rules having been applied to an interface, not a property of the rules as such. Similar implicit behavior occurs for interfaces mentioned in zone policies.

When rules are present in one direction, there is an implicit action of drop for that direction. If any of the rules are stateful, there is an implicit drop action in the opposite direction even if no rules are present in the opposite direction. Despite this condition, stateful rules always allow for reverse direction stateful traffic to flow.

The **security firewall name <name> default-action <action>** and **security firewall name <name> default-log** commands use an explicit rule and as such will prevent implicit actions from occurring in the direction that they are applied to.

Exclusion rules

Note that you should take care in employing more than one “exclusion” rule, that is, a rule that uses the negation operator (exclamation mark [!]) to exclude a rule from treatment. Rules are evaluated sequentially, and a sequence of exclusion rules could result in unexpected behavior.

Stateful firewall and connection tracking

On the firewall, connection tracking allows for stateful packet inspection.

Stateless firewalls filter packets in isolation, is based on static source and destination information. In contrast, stateful firewalls track the state of network connections and traffic flows and allow or restrict traffic based on whether its connection state is known and authorized. For example, when an initiation flow is allowed in one direction, the responder flow is automatically and implicitly allowed in the return direction. While typically slower under heavy load than stateless firewalls, stateful firewalls are better at blocking unauthorized communication.

By default, the vRouter firewall is stateless. If you want the firewall to operate stateless in general, you can configure state rules within a specific rule set. Alternatively, you can configure the firewall globally to operate statefully. For more information, refer to security firewall global-state-policy <protocol> *(page 51)*.

For all protocols, the following are tracked for each session: interface, protocol, source address, and destination address. For ICMP, the ICMP identifier is also included. For TCP/UDP/UDP-Lite/DCCP/SCTP, the source and destination ports are also included.

TCP strict tracking

The TCP strict tracking of stateful firewall rules for traffic can be enabled by using security firewall tcp-strict *(page 76)*. This command also enables the user to toggle between loose or strict stateful behaviors for TCP.

Stateful tracking must be enabled through either a state rule or global rule.

TCP strict tracking disabled—TCP connections are validated by the following criteria:

Perform SEQ/ACK numbers check against boundaries. (Reference: Rooij G., “Real stateful TCP packet filtering in IP Filter,” 10th USENIX Security Symposium invited talk, Aug. 2001.)

The four boundaries are defined as follows:

- I) SEQ + LEN <= MAX {SND.ACK + MAX(SND.WIN, 1)}\
- II) SEQ >= MAX {SND.SEQ + SND.LEN - MAX(RCV.WIN, 1)}
- III) ACK <= MAX {RCV.SEQ + RCV.LEN}

- IV) ACK >= MAX {RCV.SEQ + RCV.LEN} - MAXACKWIN

TCP strict tracking enabled—The above validation is performed. In addition, the validation against the correct TCP sequencing of flags (or validation of TCP stateful transitions) is also performed.

The following stateful transitions are invalid when a packet is received with the following flag pattern:

Forward flow:

SYN-ACK FLAG to SS, ES, FW, CW, LA, TW, CL FIN FLAG to SS, SR, S2 ACK FLAG to SS, S2

Note: S2 is an identical SYN sent from either side of the connection.

Reverse flow:

SYN FLAG to SR, ES, FW, CW, LA, TW, CL

FIN FLAG to SS, SR

Keys to the codes above are as follows:

```
vyatta@vyatta:~$ show session-table
    TCP state codes: SS - SYN SENT, SR - SYN RECEIVED, ES - ESTABLISHED, FW - FIN WAIT,
                CW - CLOSE WAIT, CG - CLOSING, LA - LAST ACK, TW - TIME WAIT, CL - CLOSED
```

Applying firewall instances to interfaces

After defining firewall instances, you can apply them to interfaces, where the instances act as packet filters. Firewall instances filter packets in one of the following ways, depending on what direction you specify when you apply the firewall instance:

in: If you apply firewall instances with the in direction, the firewall filters packets entering the interface. These packets can be traversing the vRouter or be destined for the router.

out: If you apply instances with the out direction, the firewall filters packets leaving the interface. These packets can be traversing the vRouter or originating on the vRouter.

local: If you apply instances with the local, the firewall filters packets destined for the vRouter. The special interface "lo" can be used to affect packets received on any interface. Note that these instances are run after any "in" instances that may be on the interface.

You can apply many firewall instances to an interface on each direction. They are applied in the order that they are configured on the interface and direction.

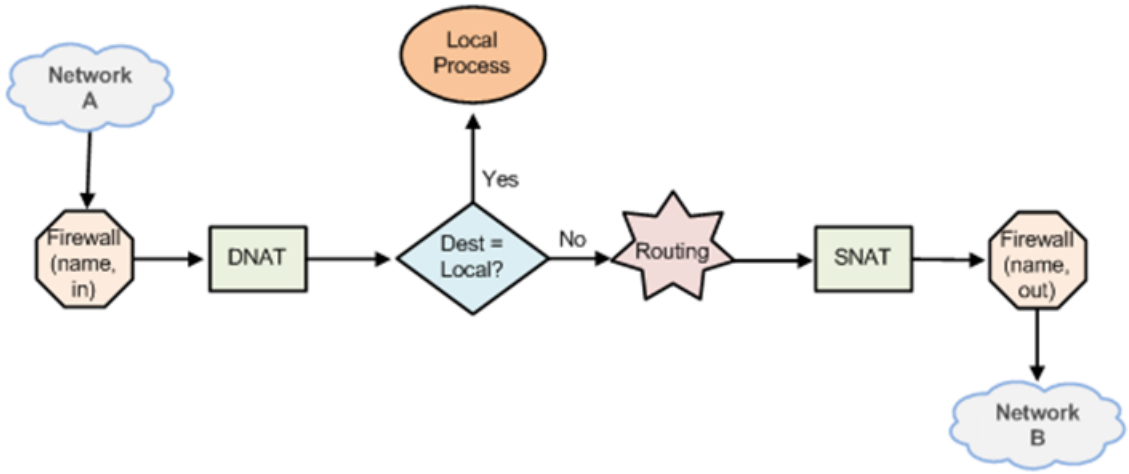
Interaction between firewall, NAT, and routing

The processing order of the various services that might be configured within the vRouter is one of the most important concepts to understand when working with firewall functionality. If the processing order of the services is not carefully configured, the results achieved might not be what you expect.

Traffic flow through firewall, NAT, and routing

The following figure shows how traffic flows through the firewall, NAT, and routing services within the vRouter. Notice the order of firewall instances, destination Network Address Translation (DNAT), routing decisions, and source Network Address Translation (SNAT).

Figure 1: Traffic flow through firewall, NAT, and routing components



Scenario 1: firewall instances applied to inbound traffic

In this scenario, firewall instances are applied to inbound (in) traffic on an interface. Notice that firewall instances are evaluated before DNAT and routing decisions, and before SNAT.

Scenario 2: firewall instances applied to outbound traffic

In this scenario, firewall instances are applied to outbound (**`out`**) traffic on an interface. Notice that firewall is evaluated after DNAT and routing decisions, and after SNAT.

Zone-based firewall

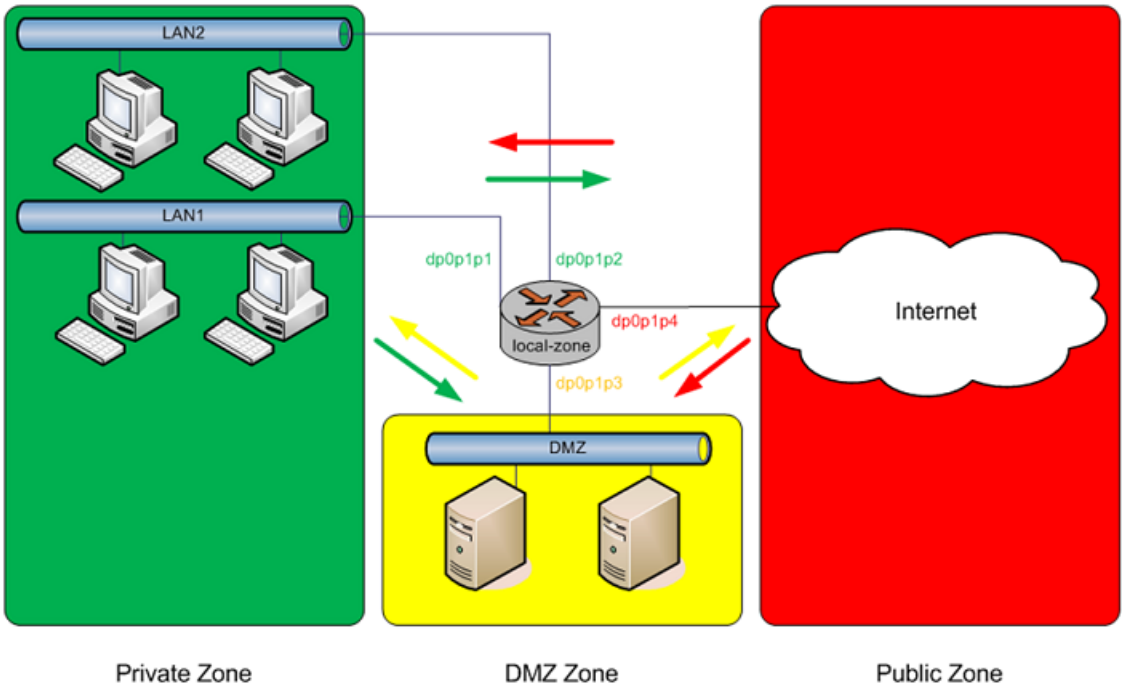
Ordinary firewall rule sets are applied on a per-interface basis to act as a packet filter for the interface. In a zone-based firewall, interfaces are grouped into security “zones,” where each interface in a zone has the same security level.

Packet-filtering policies are applied to traffic flowing between zones. Traffic flowing between interfaces that lie in the same zone is not filtered and flows freely because the interfaces share the same security level.

The following figure shows an example of a zone-based firewall implementation. This example has these characteristics:

- Three transit zones exist (that is, points where traffic transits the router): the private zone, the demilitarized zone (DMZ), and the public zone.
- The dp0p1p4 interface lies in the public zone; the dp0p1p1 and dp0p1p2 interfaces lie in the private zone; and the dp0p1p3 interface lies in the DMZ.
- The arrows from one zone to another zone represent traffic-filtering policies that are applied to traffic flowing between zones.
- Traffic flowing between LAN 1 and LAN 2 remains within a single security zone. Thus, traffic from LAN1 to LAN2, and conversely, flows unfiltered.

Figure 2: Zone-based firewall overview



By default, all traffic coming into the router and originating from the router is allowed.

Note the following additional points about zone-based firewalls:

- An interface can be associated with only one zone.
- An interface that belongs to a zone cannot have a per-interface firewall rule set applied to it, and conversely.
- Traffic between interfaces that do not belong to any zone flows unfiltered, and per-interface firewall rule sets can be applied to those interfaces.
- By default, all traffic to a zone is dropped unless explicitly allowed by a filtering policy for a source zone (**`from_zone`**) .
- Filtering policies are unidirectional; they are defined as a “zone pair” that identifies the zone from which traffic is sourced (**`from_zone`**) and the zone to which traffic is destined (**`to_zone`**). In the preceding figure, these unidirectional policies can be seen as follows:
  - From private to DMZ
  - From public to DMZ
  - From private to public
  - From DMZ to public
  - From public to private
  - From DMZ to private

Control plane policing

Control plane policing (CPP) provides protection against attacks on the AT&T Vyatta vRouter by allowing you to configure firewall policies that are assigned to desired interfaces and applying these policies to packets both entering and leaving the vRouter.

For the vRouter, CPP supports the addition of **local** keyword that can be applied to firewall policies for specific firewall interface types.

CPP is implemented when the **local** keyword is used in firewall policies that are assigned to any type of vRouter interface type supporting firewall functionality (an interface type that currently supports **in** and **out** directions) except for an administrator-defined loopback interface. The system loopback interface, **lo**, has the **local**

keyword assigned to it by default, and any attempt to assign a local firewall to a user-defined loopback interface causes an error. A local firewall policy with CPP runs on packets that are destined for the vRouter.

To configure CPP, define firewall policies or rule sets and assign them to the desired interfaces by using the **local** keyword. For the **lo** interface, assign firewall policies to control the flow of packets from the control plane. Assign firewall policies to other data plane interfaces to control the flow of packets to the control plane.

A few explicit differences exist between firewall policies that are assigned to the **local** keyword and all other firewall policies:

- Sessions are not created on a stateful rule match.
- Strict protocol tracking is silently ignored.
- Packets that do not match a firewall rule are allowed to pass into and out of the control plane.

For the first two explicit differences, regardless of whether a matched rule implies stateful or strict protocol tracking, these attributes of the rule are silently ignored. This behavior is required because packets entering or leaving the control plane also pass through an input or output interface and the possibility of performing duplicate state tracking can result in false-positive state transitions, which lead to packet drop. To enforce stateful behavior, strict protocol tracking, or both, add appropriate rules to the input or output interfaces as desired.

The third difference enables packets that are unmatched by a policy or rule set to pass. This behavior is the direct opposite of all other firewall behavior. Other firewalls have an implicit drop rule for all packets that do not match an existing rule in the rule set. This behavior is implemented as a convenience for the administrator to allow various control plane packets, such as DHCP, IPv6 ND, BGP, and so forth, to pass without requiring the administrator to create specific rules for these packets. Administrators can have full control over this behavior and can add an explicit drop rule to the firewall group, if desired.

CPP is described in RFC 6192, and a suggested configuration for filtering rules is included in that document. Administrators are encouraged to review RFC 6192 for a list of suggested ACLs and configuration filtering rules for control plane policing.

The AT&T Vyatta vRouter also includes a template of suggested filtering rules that you can incorporate into your CPP configuration. This rule set excludes various routing protocol packets from filtering and provides a default policing rule to rate-limit all other packets entering the control plane. The template CPP configuration also assigns the rule set to the **lo** system loopback interface.

The template rule set is located on the vRouter in: /opt/vyatta/etc/cpp.conf. After reviewing the template configuration, you can add this rule set to your existing configuration by using the **merge** command in configuration mode:

```
vyatta@R1# merge /opt/vyatta/etc/cpp.conf
vyatta@R1# commit
vyatta@R1# save
```

Administrators may also choose to modify the template rules to meet their particular needs.

Firewall denial of service protection

A stateful firewall or NAT creates a session for each traffic flow matching that firewall or NAT provided it is not blocked. This applies to both connection-oriented protocols (for example, TCP) and nonconnection-oriented protocols (for example, UDP and ICMP echo).

The Firewall Denial-of-Service Protection feature provides commands that perform the following tasks:

- Monitor the number of sessions, rate of session creation, and time last session was created
- Limit the maximum number of half-open sessions
- Rate-limit new sessions

Maximum half-open sessions

The definition of a half-open session depends upon the protocol. For TCP, a session is deemed to be half open while it is going through the SYN, SYN-ACK, and ACK three-way handshake. For nonconnection-oriented protocols, a session is deemed half open when traffic has been seen only in the forward direction.

A half-open session has a default timeout period of 30 seconds. If no further traffic is seen on this session for that time period, the session is "expired". An expired session then exists for a further 5 to 10 seconds before it is deleted and memory released. Once expired, a session is not available to traffic.

When the maximum half-open limit is reached, a matching packet is prevented from creating a session.

Session rate limiting

Session rate limiting limits the maximum rate at which a session can be created. A "rate" value and a "burst" value may be configured. These values combine to determine the interval over which the rate limiting is evaluated. For example, if the rate limit is 20 sessions per second, and the burst is 100 sessions, the interval is 5 seconds (100/20). A maximum of 100 new sessions is allowed during that 5-second interval. In the `show` command output, the interval is shown in milliseconds.

When the rate-limit rate is reached, a matching packet is prevented from creating a session.

Rate limiting itself limits the maximum number of half-open sessions. For example, if the rate limit is 20 seconds and the default timeout of 30 seconds applies, the maximum number of half-open sessions is 600 sessions (20 x 30, that is, the number of sessions that can be created before the oldest expires).

If the rate limiting and maximum half-open features are combined, with a rate limit of 20 sessions per second and a maximum half-open value of 300, then it takes 15 seconds (300/20) for the maximum half-open limit to be reached.

DoS protection configuration considerations

DoS protection requires that you configure a system session limit parameter and a session limit group. The parameter contains the configuration and state for maximum half-open and rate-limiting. The group contains the match criteria rule set, and a list of interfaces to which that rule set is applied. The rule set contains a list of rules, each of which must reference a parameter.

Multiple interfaces can be configured on the same session limit group. A session limit group's rule set can reference multiple session limit parameters. Multiple session limit groups can reference the same session limit parameter.

A session limit parameter can be configured with one, both, or neither of the following features:

- Policing of maximum half-open sessions
- Rate-limiting new sessions

Note: If not configured with either feature, the session limit parameter just gathers session rate and statistics information.

A session limiter configured on an interface applies to both inbound and outbound sessions created on that interface. There is no direction (in or out) when configuring a session limit interface. The session limiter is applied to sessions that are created for both inbound and outbound, if other firewall or NAT rules exist to create those sessions. Therefore, if a session limiter is configured for the dp0p1s1 interface, and there is only an input firewall on dp0p1s1, the session limiter applies only to inbound sessions because outbound sessions exist.

A session limiter can limit only sessions that are created after the session limiter is created. For example, if there are 100 half-open sessions and a session limiter is created with max-halfopen configured as 50, those 100 half-open sessions remain. Also, the session limiter counts do not count those 100 half-open sessions.

Session and packet logging

You can configure the vRouter for the following types of logging:

- Session logging. Configure stateful rules to log session state transitions.
- Per packet logging. Log every packet that matches a network packet filter rule, such as a firewall rule or NAT rule.

Note: Per-packet logging generates large amounts of output and can negatively affect the performance of the entire system. Use per packet logging only for debugging purposes.

When logging is enabled, all log messages can be accessed by using the **`show dataplane log`** command.

Session Logging

A stateful firewall rule is created by adding the **state enabled** keywords to a firewall rule. By design, all NAT rules are stateful rules.

When a flow matches either a stateful firewall rule or a NAT rule, a session is created. The session tracks the state transitions of its IP protocol.

For UDP, ICMP, and all non-TCP flows, a session transitions to four states over the lifetime of the flow. For each transition, you can configure the product to log a message. TCP has a larger number of state transitions, each of which can be logged.

Use the **security firewall session-log** command to configure firewall session logging. When logging is configured, a log message is generated for each state transition.

Per packet logging for debugging

You can set up filtering rules so that each packet matched by the rule is logged.

AT&T recommends limiting per packet logging to debugging. Per packet logging occurs in the forwarding paths and can greatly reduce the throughput of the system and dramatically increase the disk space used for the log files. For all operational purposes, use stateful session logging instead of per packet logging.

To implement per packet logging for debugging purposes, you can include the **log** keyword when specifying a rule. When the logging option is specified, a log message containing the parameters of the packet is generated and logged.

Application aware firewall - DPI support

Deep packet inspection (DPI) is a mechanism whereby the content of packets beyond the basic IP and transport (TCP/UDP) headers is inspected.

The application aware firewall feature allows you to apply DPI processing to firewall rules in a stateful firewall session. The recommended use case is to restrict or limit the expected protocol on a port. For example, if you have opened a port for SMTP traffic, you can use the application aware firewall to ensure that the port is used only for SMTP. You can also open a port and then restrict the set of applications that can run on the port.

When a stateful firewall session is created, the system determines the application that is running over the session, and bases decisions on the fate of packets on that application. The processing applies to TCP or UCP packets only. All packets in the session are passed until a rule in the ordered list of application firewall rules matches, or until too much traffic has passed without achieving a match.

The following example shows one method of configuring the vRouter to limit traffic on the 'smtp' port to SMTP only.

The first set of commands defines the application firewall 'ensure-SMTP' and includes a description, numbered rules, and a no-match action. (The no-match action is included for completeness; it is not required in this context, because the default value is 'dropped'.)

The second group of commands defines a standard stateful firewall and specifies the behavior for firewall rule 10. The final command ties this configuration back to the application firewall that is configured in the first set of commands by enabling the firewall state and behavior and specifying that the 'ensure-SMTP' application firewall will run within the firewall session.

```
set security application firewall name ensure-SMTP description 'Only allow an
SMTP session'
set security application firewall name ensure-SMTP no-match-action 'drop'
set security application firewall name ensure-SMTP rule 10 action 'accept'
set security application firewall name ensure-SMTP rule 10 name 'smtp‘


set security firewall name DPI-example rule 10 action 'accept'
set security firewall name DPI-example rule 10 description 'Allow SMTP'
set security firewall name DPI-example rule 10 destination port 'smtp'
set security firewall name DPI-example rule 10 protocol 'tcp'
set security firewall name DPI-example rule 10 session application firewall
'ensure-SMTP'
```

The following configuration achieves the same result as the previous example. In this simplified case, an application firewall is not defined as a container for additional configuration. Instead, this configuration applies firewall rule 10 directly to any session with application name 'smtp.' You can apply this type of configuration to sessions based on application name, application type, or application protocol.

```
set security firewall name DPI-example rule 10 action 'accept'
set security firewall name DPI-example rule 10 description 'Allow SMTP'
set security firewall name DPI-example rule 10 destination port 'smtp'
set security firewall name DPI-example rule 10 protocol 'tcp'
set security firewall name DPI-example rule 10 session application name ‘smtp’
```

Comparison of protocol and application name matching

Protocol matching and application matching can have similar results, but there are differences between the two. Consider the following commands:

```
set security application firewall name <name> rule <rule-number> name <app-name>

set security application firewall name <name> rule <rule-number> protocol <app-protocol>
```

The first command matches a specific application by name, while the second command matches the application protocol.

For example, if the application 'facebook' is running over HTTP, the protocol layers include IP, TCP, HTTP, and facebook. The first command would specify facebook, while the second command would specify HTTP.

An issue can occur if a future release of the DPI engine gains the ability to identify a new application over HTTP, The current DPI engine cannot identify 'newapp' traffic, so it classifies it with the protocol 'http' and the application name 'http.' The updated DPI engine would continue to identify 'newapp' traffic with protocol 'http,' but the application name would change from 'http' to 'newApp.' The necessary application level rule would have to specify 'newapp' as the application name.

Best practices for protocol and application matching:

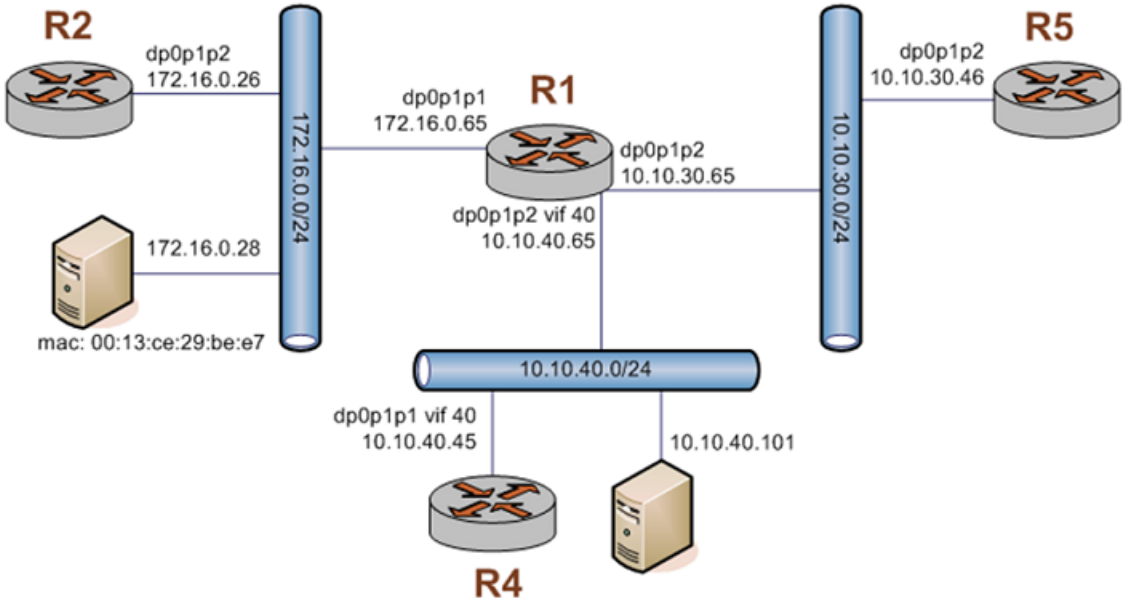
- Use a protocol rule if you want to match any applications that use that protocol.
- Use an application rule if you want to match only a specific named application.
- Use more specific rules (such as application name rules) earlier in the ruleset, and more general rules (such as protocol rules) later in the ruleset as a catch-all

Configuration Examples

Packet-filtering

This section describes a sample configuration for firewall. When you have finished, the firewall is configured on the R1 router, as shown in the following figure.

Figure 3: Firewall: sample configuration



This section includes the following examples:

- Filtering on source IP address *(page 19)*
- Filtering on source and destination IP addresses *(page 20)*
- Filtering on source IP address and destination protocol *(page 21)*
- Defining a network-to-network filter *(page 22)*
- Filtering on source MAC address *(page 23)*
- Excluding an address *(page 24)*
- Matching TCP flags *(page 25)*
- Matching ICMP type names *(page 26)*
- Matching groups *(page 27)*
- Configuring stateful behavior per rule set *(page 29)*

Filtering on source IP address

The following figure shows how to define a firewall instance that contains one rule, which filters packets only on source IP address. This rule denies packets coming from the R2 router. It then applies the firewall instance to packets inbound on the dp0p1p1 interface.

To create an instance that filters on source IP address, perform the following steps in configuration mode.

Table 1: Filtering on source IP

Step	Command
Define the action of this rule.	`vyatta@R1# set security firewall name FWTEST-1 rule 1 action accept`
Define a rule that filters traffic on the 176.16.0.26 source IP address.	`vyatta@R1# set security firewall name FWTEST-1 rule 1 source address 172.16.0.26`
Apply FWTEST-1 to inbound packets on dp0p1p1.	`vyatta@R1# set interfaces dataplane dp0p1p1 firewall in FWTEST-1`
Commit the configuration.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show security firewall name FWTEST-1`
	`name FWTEST-1 {`
	`  rule 1 {`
	`  action accept`
	`  source {`
	`  address 172.16.0.26`
	`  }`
	`  }`
	`}`
	`vyatta@R1# show interfaces dataplane dp0p1p1`
	`dataplane dp0p1p1 {`
	`  address 172.16.1.1/24`
	`  firewall {`
	`  in FWTEST-1`
	`  }`
	`}`

Filtering on source and destination IP addresses

The following example shows how to define another firewall instance. This instance contains one rule, which filters packets on both source and destination IP addresses. The rule accepts packets leaving R5 through dp0p1p2 using 10.10.30.46 and destined for 10.10.40.101. It then applies the firewall instance to packets outbound from the 1 virtual interface (vif 1) on the dp0p1p2 interface.

To create an instance that filters on source and destination IP addresses, perform the following steps in configuration mode.

Table 2: Filtering on source and destination IP

Step	Command
Create the configuration node for the FWTEST-2 firewall instance and its rule 1. This rule accepts traffic matching the specified criteria.	`vyatta@R1# set security firewall name FWTEST-2 rule 1 action accept`
Define a rule that filters traffic on the 10.10.30.46 source IP address.	`vyatta@R1# set security firewall name FWTEST-2 rule 1 source address 10.10.30.46`

Step	Command
Define a rule that filters traffic on the 10.10.40.101 destination IP address.	`vyatta@R1# set security firewall name FWTEST-2 rule 1 destination address 10.10.40.101`
Apply FWTEST-2 to outbound packets on dp0p1p2 vif 40.	`vyatta@R1# set interfaces dataplane dp0p1p2 vif 40 firewall out FWTEST-2`
Commit the configuration.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show security firewall name FWTEST-2`
	`name FWTEST-2 {`
	`  rule 1 {`
	`  action accept`
	`  destination {`
	`  address 10.10.40.101`
	`  }`
	`  source {`
	`  address 10.10.30.46`
	`  }`
	`  }`
	`}`
	`vyatta@R1# show interfaces dataplane dp0p1p2`
	`dataplane dp0p1p2 {`
	`  vif 40 {`
	`  firewall {`
	`  out FWTEST-2`
	`  }`
	`  }`
	`}`

Filtering on source IP address and destination protocol

The following example shows how to define a firewall rule that filters on source IP address and destination protocol. This rule allows TCP packets originating from address 10.10.30.46 (that is, R5), and destined for the Telnet port of R1. The instance is applied to local packets (that is, packets destined for this router, R1) through the dp0p1p2 interface.

To create an instance that filters on source IP address and destination protocol, perform the following steps in configuration mode.

Table 3: Filtering on source IP and destination protocol

Step	Command
Create the configuration node for the FWTEST-3 firewall instance and its rule 1. This rule accepts traffic matching the specified criteria.	`vyatta@R1# set security firewall name FWTEST-3 rule 1 action accept`
Define a rule that filters traffic on the 10.10.30.46 source IP address.	`vyatta@R1# set security firewall name FWTEST-3 rule 1 source address 10.10.30.46`
Define a rule that filters TCP traffic.	`vyatta@R1# set security firewall name FWTEST-3 rule 1 protocol tcp`

Step	Command
Define a rule that filters traffic destined for the Telnet service.	`vyatta@R1# set security firewall name FWTEST-3 rule 1 destination port telnet`
Apply FWTEST-3 to packets bound for this router arriving on dp0p1p2.	`vyatta@R1# set interfaces dataplane dp0p1p2 firewall in FWTEST-3`
Commit the configuration.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show security firewall name FWTEST-3`
	`name FWTEST-3 {`
	`  rule 1 {`
	`  action accept`
	`  destination {`
	`  port telnet`
	`  }`
	`  protocol tcp`
	`  source {`
	`  address 10.10.30.46`
	`  }`
	`  }`
	`}`
	`vyatta@R1# show interfaces dataplane dp0p1p2`
	`dataplane dp0p1p2 {`
	`  firewall {`
	`  in FWTEST-3`
	`  }`
	`}`

Defining a network-to-network filter

The following example shows how to define a network-to-network packet filter, allowing packets originating from 10.10.40.0/24 and destined for 172.16.0.0/24. It then applies the firewall instance to packets inbound through the 40 virtual interface (vif 40) and the dp0p1p2 interface.

To create a network-to-network filter, perform the following steps in configuration mode.

Table 4: Defining a network-to-network filter

Step	Command
Create the configuration node for the FWTEST-4 firewall instance and its rule 1. This rule accepts traffic matching the specified criteria.	`vyatta@R1# set security firewall name FWTEST-4 rule 1 action accept`
Define a rule that filters traffic coming from the 10.10.40.0/24 network.	`vyatta@R1# set security firewall name FWTEST-4 rule 1 source address 10.10.40.0/24`
Define a rule that filters traffic destined for the 172.16.0.0/24 network.	`vyatta@R1# set security firewall name FWTEST-4 rule 1 destination address 172.16.0.0/24`

Step	Command
Apply FWTEST-4 to packets bound for this router arriving through vif 40 on dp0p1p2.	`vyatta@R1# set interfaces dataplane dp0p1p2 vif 40 firewall in FWTEST-4`
Commit the configuration.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show security firewall name FWTEST-4`
	`name FWTEST-4 {`
	`  rule 1 {`
	`  action accept`
	`  destination {`
	`  address 172.16.0.0/24`
	`  }`
	`  source {`
	`  address 10.10.40.0/24`
	`  }`
	`  }`
	`}`
	`vyatta@R1# show interfaces dataplane dp0p1p2`
	`dataplane dp0p1p2 {`
	`  vif 40 {`
	`  firewall {`
	`  in FWTEST-4`
	`  }`
	`  }`
	`}`

Filtering on source MAC address

The following example shows how to define a firewall instance that contains one rule, which filters packets only on source medium access control (MAC) address. This rule allows packets coming from a specific computer, identified by its MAC address rather than its IP address. The instance is applied to packets inbound on the dp0p1p1 interface.

To create an instance that filters on source MAC address, perform the following steps in configuration mode.

Table 5: Filtering on source MAC address

Step	Command
Create the configuration node for the FWTEST-5 firewall instance and its rule 1. This rule accepts traffic matching the specified criteria.	`vyatta@R1# set security firewall name FWTEST-5 rule 1 action accept`
Define a rule that filters traffic with the 00:13:ce:29:be:e7 source MAC address.	`vyatta@R1# set security firewall name FWTEST-5 rule 1 source mac-address 00:13:ce:29:be:e7`
Apply FWTEST-5 to inbound packets on dp0p1p1.	`vyatta@R1# set interfaces dataplane dp0p1p1 firewall in FWTEST-5`
Commit the configuration.	`vyatta@R1# commit`

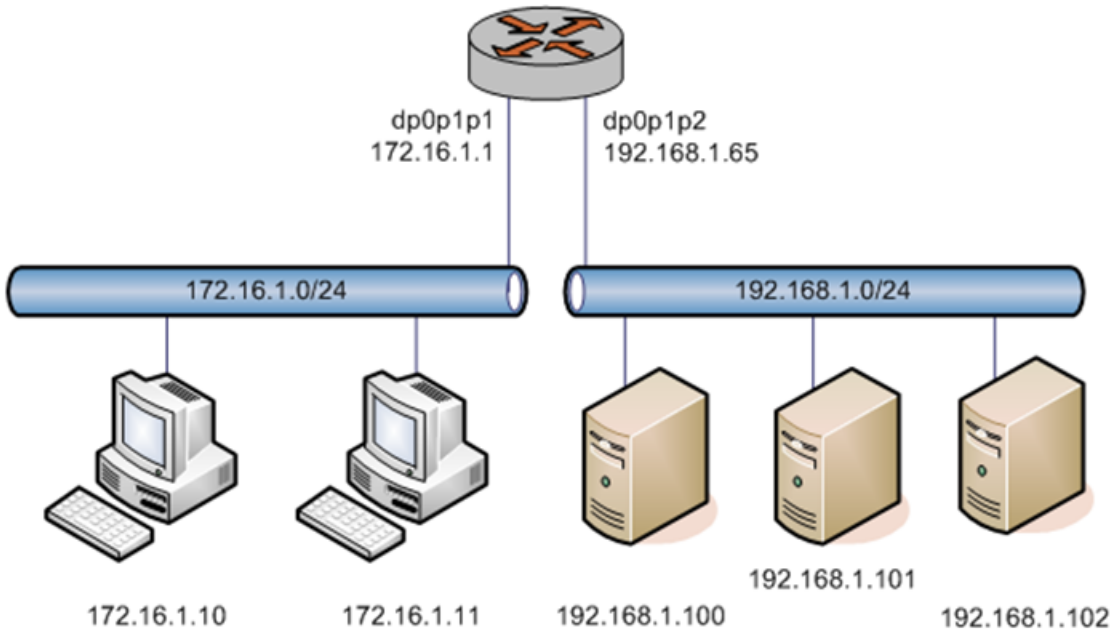
Step	Command

Show the configuration.	vyatta@R1# show security firewall name
	FWTEST-5
	name FWTEST-5 [
	rule 1 {
	action accept
	source {
	mac-address 0:13:ce:29:be:e7
	}
	}
	}
	vyatta@R1# show interfaces dataplane dp0p1p1
	dataplane dp0p1p1 {
	address 172.16.1.1/24
	firewall {
	in FWTEST-5
	}
	}

Excluding an address

The firewall rule shown in the following example allows all traffic from the 172.16.1.0/24 network except traffic to the 192.168.1.100 server.

Figure 4: Excluding an address



To create an instance that excludes an address, perform the following steps in configuration mode.

Table 6: Excluding an address

Step	Command
Create the configuration node for the FWTEST-5 firewall instance and its rule 10. Give a description for the rule.	`vyatta@R1# set security firewall name NEGATED-EXAMPLE rule 10 description "Allow all traffic from LAN except to server 192.168.1.100"`
Allow all traffic that matches the rule to be accepted.	`vyatta@R1# set security firewall name NEGATED-EXAMPLE rule 10 action accept`
Allow any traffic from the 172.16.1.0/24 network that matches the rule to be accepted.	`vyatta@R1# set security firewall name NEGATED-EXAMPLE rule 10 source address 172.16.1.0/24`
Allow traffic destined anywhere except the 192.168.1.100 destination address that matches the rule to be accepted.	`vyatta@R1# set security firewall name NEGATED-EXAMPLE rule 10 destination address !192.168.1.100`
Apply the NEGATED-EXAMPLE instance to inbound packets on dp0p1p1.	`vyatta@R1# set interfaces dataplane dp0p1p1 firewall in NEGATED-EXAMPLE`
Commit the configuration.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show security firewall`
	
	`name NEGATED-EXAMPLE {`
	`  rule 10 {`
	`  action accept`
	`  description "Allow all traffic from LAN except to server 192.168.1.100"`
	`  destination {`
	`  address !192.168.1.100`
	`  }`
	`  source {`
	`  address 172.16.1.0/24`
	`  }`
	`  }`
	`}`
	
	`vyatta@R1# show interfaces dataplane dp0p1p1`
	`dataplane dp0p1p1 {`
	`  address 172.16.1.1/24`
	`  firewall {`
	`  in NEGATED-EXAMPLE`
	`  }`
	`}`

Matching TCP flags

The vRouter supports filtering on the TCP flags within TCP packets. For example, to create a rule to accept packets with the SYN flag set and the ACK, FIN, and RST flags unset, perform the following steps in configuration mode.

Table 7: Accepting packets with specific TCP flags set

Step	Command
Set the protocol to match to TCP.	`vyatta@R1# set security firewall name TCP-FLAGS rule 30 protocol tcp`
Set the TCP flags to match.	`vyatta@R1# set security firewall name TCP-FLAGS rule 30 tcp flags SYN,!ACK,!FIN,!RST`
Set the action to accept.	`vyatta@R1# set security firewall name TCP-FLAGS rule 30 action accept`
Commit the configuration.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show security firewall name TCP-FLAGS`
	
	`name TCP-FLAGS {`
	`  rule 30 {`
	`  action accept`
	`  protocol tcp`
	`  tcp {`
	`  flags SYN,!ACK,!FIN,!RST`
	`  }`
	`  }`
	`}`
	`vyatta@R1#`

Matching ICMP type names

Packets can be filtered for ICMP type names. For example, to create a rule that allows only ICMP echo request packets, perform the following steps in configuration mode.

Note: You can configure rules to match IPv4 ICMP, IPv6 ICMP, IPv6 routing header, or TCP without specifying the respective protocol, provided that a protocol specific match option is present. For example, ICMP type and TCP flags.

Table 8: Accepting ICMP packets with specific type names

Step	Command
Set the protocol to match to ICMP.	`vyatta@R1# set security firewall name ICMP-NAME rule 40 protocol icmp`
Set the ICMP packet type to match.	`vyatta@R1# set security firewall name ICMP-NAME rule 40 icmp name echo-request`
Set the action to accept.	`vyatta@R1# set security firewall name ICMP-NAME rule 40 action accept`
Commit the configuration.	`vyatta@R1# commit`

Step	Command

Show the configuration.	`vyatta@R1# show security firewall name ICMP-NAME`
	
	`name ICMP-NAME {`
	`  rule 40 {`
	`  action accept`
	`  protocol icmp`
	`  icmp {`
	`  name echo-request`
	`  }`
	`  }`
	`}`
	`vyatta@R1#`

Matching groups

Groups of addresses, ports, and networks can be defined for similar filtering. For example, to create a rule that rejects traffic to a group of addresses and ports and from a group of networks, perform the following steps in configuration mode.

Table 9: Rejecting traffic based on groups of addresses, networks, and ports

Step	Command
Add an address to an address group.	`vyatta@R1# set resources group address-group SERVERS address 1.1.1.7`
Add a network to a address group.	`vyatta@R1# set resources group address-group SERVERS address 10.0.10.0/24`
Add port 22 and ports 1000 through 2000 to the PORTS port group.	`vyatta@R1# set resources group port-group PORTS port 22` `vyatta@R1# set resources group port-group PORTS port 1000-2000`
Add a port name to the PORTS port group.	`vyatta@R1# set resources group port-group PORTS port http`
Commit the configuration.	`vyatta@R1# commit`

Step	Command

Show the configuration.	`vyatta@R1# show resources`
	
	`group {`
	`  address-group SERVERS {`
	`  address 10.0.10.0/24`
	`  address 1.1.1.7`
	`  }`
	`  port-group PORTS {`
	`  port 22`
	`  port 1000-2000`
	`  port http`
	`  }`
	`}`
	`vyatta@R1#`

Specify a reject action within a firewall instance.	`vyatta@R1# set security firewall name REJECT-GROUPS rule 10 action drop`
Specify the protocol.	`vyatta@R1# set security firewall name REJECT-GROUPS rule 10 protocol tcp`
Specify an address group to match as a destination.	`vyatta@R1# set security firewall name REJECT-GROUPS rule 10 destination address SERVERS`
Specify a port group to match as a destination.	`vyatta@R1# set security firewall name REJECT-GROUPS rule 10 destination port PORTS`
Specify an address group to match as a source.	`vyatta@R1# set security firewall name REJECT-GROUPS rule 10 source address SERVERS`
Commit the configuration.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show security firewall name`
	` REJECT-GROUPS`
	
	`name REJECT-GROUPS {`
	`  rule 10{`
	`  action drop`
	`  destination {`
	`  address SERVERS`
	`  port PORTS`
	`  }`
	`  protocol tcp`
	`  source {`
	`  address SERVERS`
	`  }`
	`  }`
	`}`
	`vyatta@R1#`

Stateful behavior

Stateless firewalls filter packets in isolation, based on static source and destination information. In contrast, stateful firewalls track the state of network connections and traffic flows and allow or restrict traffic based on whether its connection state is known and authorized. For example, when an initiation flow is allowed in one direction, the responder flow is automatically and implicitly allowed in the return direction.

The firewall always attempts to perform stateful matching, even if there are no sessions or stateful rules. The existence of a stateful rule on an interface means that the implicit behaviors for that interface are filtered. A stateful rule in one direction causes the other direction (in the absence of any rules) to block packets if they do not match a session.

For stateful behavior,

- The system determines if the packet can be matched to an existing session, such as would have been created by a stateful rule.
- For ICMP errors, a check is done to determine whether the embedded packet (which triggered the error) matches an existing session. If no session matches, a rule-based match is attempted.
- If a session created by a stateful firewall rule (accept rule) matches, the packet is allowed to pass.
- If a session created by NAT matches, and the packet is flowing in the backwards direction, it is allowed to pass. The only way to block backward direction NAT packets is to block the forward direction packet with a firewall rule.
- If a session created by an ALG matches (match on a child session such as an FTP data flow), the packet is allowed to pass. The only way to block such ALG child flows is to block the parent flow.
- When a stateful firewall rule is processed and the action is accept, a session is created based on the IP addresses, protocol and ports (for supported protocols that use ports).

To improve efficency of the firewall handling, further packets matching the session will be accepted, without running checks given in the firewall rule.

Apart from the initial packet, the checks associated with the following per-rule configuration are not performed:

- dscp <DSCP-value>
- pcp <PCP-value>
- tcp flags <TCP-flags-to-match>

Configuring stateful behavior per rule set

Even if you want the firewall to operate statelessly in general, you can still configure state rules within a specific rule set.

The following example shows how to configure a rule in the TEST1 firewall rule set. Rule 1 accepts stateful traffic flows and allows related flows for the ALGs that are enabled.

To configure per-rule set state rules, perform the following steps in configuration mode.

Table 10: Creating a per-rule set state rule

Step	Command
Create the configuration node for the TEST1 rule set and give a description for the rule set.	`vyatta@R1# set security firewall name TEST1 description "Filter traffic statefully"`

Create a state rule that allows only established and related traffic.	`vyatta@R1# set security firewall name TEST1 rule 1 action accept`
	
	`vyatta@R1# set security firewall name TEST1 rule 1 state enable`

Step	Command
Commit the configuration.	`vyatta@R1# commit`

Show the firewall configuration.	`vyatta@R1# show security firewall name TEST1`
	`description "Filter traffic statefully"`
	`rule 1 {`
	`  action accept`
	`  state enable`
	`}`

Configuring global state policies

You can change behavior to be globally stateful by setting a global state policy with security firewall global-state-policy <protocol> *(page 51)*. When state policies are defined, state rules for return traffic of that type need not be explicitly mentioned within the rule sets.

The following apply to global stateful rules:

- A global stateful rule affects only the firewall rules that explicitly (or by inference) refer to that protocol. This inference can occur if the **protocol** keyword has been omitted for TCP, ICMP or ICMPv6 rules.
- ICMP sessions are created only for echo-request packets. Attempting to create a session for an echo-response results in a packet drop.
- It is usually not necessary to specify default-action (or default-log). Reserve default-action for use with a stateless firewall if you want to block only a few packets and pass all others using default-action accept.

Consider the following configuration. In this configuration, each of the rules 10, 20, 30, 40, 100, 200 act as if they also had state enable present. Rule 400 is not affected, and does not enable a state.

The following protocol-specific notes apply to this example:

ICMP

An IPv4 ICMP echo-request packet matches rule 10, creates a state, and allows ICMP echo-response packets to be received. The same applies to IPv6 ICMP echo-request packets and rule 20.

ICMP sessions are created only for echo-request packets. Any attempt to create a session for echo-response packet fails. An echo-response in the presence of the example ruleset will match rule 30 (or 40 for IPv6), and be dropped. Other ICMP packets are allowed through. In this example, it is not necessary to use the **security firewall global-state-policy icmp** rule because state enable can be used for rule 10 or 20. ICMP errors corresponding to an existing session are always passed (and NAT translated) unless explicitly blocked by a firewall rule.

TCP

For TCP, rule 200 allows outbound traffic to port 80 (http), and allows its response packets. Rule 400 allows out all other packets (including other TCP packets), but packets matching these rules do not create a state. Outbound TCP traffic to a port such as port 88 is allowed, but its response packets are blocked.

UDP

The example ruleset allows all UDP traffic, including requests and responses.

Example configuration

```
security {
        firewall {
                global-state-policy {
                        icmp
                        tcp
                        udp
                }
                name GblState {
                        rule 10 {
                                action accept
```

```
                                icmp {
                                        name echo-request
                                }
                        }
                        rule 20 {
                                action accept
                                icmpv6 {
                                        name echo-request
                                }
                        }
                        rule 30 {
                                action accept
                                protocol icmp
                        }
                        rule 40 {
                                action accept
                                protocol ipv6-icmp
                        }
                        rule 100 {
                                action accept
                                protocol udp
                        }
                        rule 200 {
                                action accept
                                destination {
                                        port 80
                                }
                                protocol tcp
                        }
                        rule 400 {
                                action accept
                        }
                }
        }
}
```

Example steps to configure a global firewall policy to allow all return traffic

The following example shows the steps to configure a firewall globally to allow all return traffic. In addition, the firewall allows any traffic (such as FTP data) that is related to allowed traffic in the original direction. The firewall drops invalid traffic.

To configure this global stateful behavior, perform the following steps in configuration mode.

Table 11: Setting a global state policy

Step	Command

Configure global state policy.	`vyatta@R1# set security firewall global-state-policy icmp`
	
	`vyatta@R1# set security firewall global-state-policy tcp`
	
	`vyatta@R1# set security firewall global-state-policy udp`

Commit the configuration.	`vyatta@R1# commit`

Step	Command

Show the state policy configuration.	`vyatta@R1# show security firewall global-state-policy`
	`security {`
	`  firewall {`
	`  global-state-policy {`
	`  icmp`
	`  tcp`
	`  udp`
	`  }`
	`  }`
	`}`

Changes in global-state-policy behavior

This section describes changes in global-state-policy behavior prior to Release 5.1 and gives an example of how to achieve similar functionality for later releases.

Prior to Release 5.1, the vRouter would add implicit rules when global state policies were defined. From release 5.1 onwards this no longer occurs. The reason for the change is to ensure that firewalls are not "opened up" unintentionally. The details of the behavior change are as follows.

Prior to Release 5.1, a rule group named "default_state_group" would be added after all rule groups configured on interfaces, in both the "out" and the "in" directions. Its contents would depend on what values were set for "global-state-policy" (possible values are one or more of "icmp", "tcp", and "udp"). If all three were configured, i.e.

```
set global-state-policy icmp
set global-state-policy tcp
set global-state-policy udp
```

Then the following would be its contents:

```
rule 100 - allow stateful proto tcp
rule 200 - allow stateful proto udp
rule 300 - allow stateful proto icmp
```

If a protocol was not set as global-state-policy, then an entry would not appear for that protocol.

If with release 5.1 and greater, you would like similar functionality as earlier releases, an explicit group of rules needs to be created which should be applied to each interface and direction (e.g. "in" and "out") after all rule groups you matched earlier (if any).

For example, if the configuration has the lines:

```
set global-state-policy icmp
set global-state-policy tcp
set global-state-policy udp
```

then similar functionality can be achieved by the added configuration:

```
set security firewall name DEFAULT-FW rule 100 action accept
set security firewall name DEFAULT-FW rule 100 protocol tcp
set security firewall name DEFAULT-FW rule 200 action accept
set security firewall name DEFAULT-FW rule 200 protocol udp
set security firewall name DEFAULT-FW rule 300 action accept
set security firewall name DEFAULT-FW rule 300 protocol icmp
```

for each interface **`IF-NAME`** where firewall groups were applied in the "in" direction, configure the following *after* all other firewall groups on the interface:

```
set interfaces dataplane IF-NAME firewall in DEFAULT-FW
```

and for each interface **IF-NAME** where firewall groups were applied in the "out" direction, configure the following *after* all other firewall groups on the interface:

```
set interfaces dataplane IF-NAME firewall out DEFAULT-FW
```

Using firewall with VRRP interfaces

A Virtual Router Redundancy Protocol (VRRP) interface is a logical abstraction that allows the system to implement RFC 3768-compliant MAC address behavior. VRRP can be configured with or without VRRP interfaces. To achieve the expected results when filtering traffic, it is important to understand how traffic flows on systems that use VRRP.

- If no VRRP interface is designed, traffic flows in and out through a physical interface or virtual interface.
- If a VRRP interface is designed, traffic flows in through the VRRP interface and out through the physical interface or virtual interface.

This traffic flow affects how you design and attach firewall rule sets.

Applying a rule set to a VRRP interface

When a host sends a packet to the router, the packet ingresses through the VRRP interface. But when the router sends traffic to the host, traffic egresses through the parent interface or virtual interface.

The firewall rule sets for the VRRP interface and the physical interface are independent. Specifically, packet-filtering rules applied to incoming traffic on the parent interface are not applied to traffic arriving on the VRRP interface. When designing firewall rule sets for incoming traffic, make sure you apply an appropriate rule set for your VRRP interface; otherwise, all incoming traffic is unfiltered.

The example in Filtering on source IP address *(page 19)* shows how to define a simple firewall rule set, FWTEST-1, which filters on source IP address. The following example shows how to apply the same rule set to inbound traffic on the VRRP interface. In this example, the dp0p1p3 interface is already configured. Specifically:

- It is a member of VRRP group 15.
- It has rule set FWTEST-1 applied for inbound traffic.

To apply the rule set to the VRRP interface, perform the following steps in configuration mode.

Table 12: Applying a firewall rule set to a VRRP interface

Step	Command

View the initial configuration for the interfaces.	`vyatta@R1# show interfaces`
	
	`dataplane dp0p160p1 {`
	`  address 10.1.32.73/24`
	`  mtu 1500`
	`}`
	`dataplane dp0p192p1 {`
	`  address 10.10.10.3/24`
	`  address 2014:14::3/64`
	`  mtu 1500`
	`  vrrp {`
	`  vrrp-group 10 {`
	`  virtual-address 10.10.10.50`
	`  }`
	`  }`
	`}`
	`dataplane dp0p224p1 {`
	`  address 192.168.1.1/24`
	`  ip {`
	`  }`
	`  mtu 1500`
	`}`
	`dataplane dp0p256p1 {`
	`  address 20.20.20.3/24`
	`  address 2020:20::3/64`
	`  mtu 1500`
	`}`
	`loopback lo {`
	`  ipv6 {`
	`  }`
	`}`

Attach the same FW-TEST1 rule set for inbound traffic on the VRRP interface.	`vyatta@R1# set interfaces dataplane dp0p192p1 firewall in NEGATED-EXAMPLE`
Commit the configuration.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show interfaces dataplane dp0p192p1`
	
	`address 172.16.1.20/24`
	`firewall {`
	`  in FWTEST-1`
	`}`
	`mtu 1500`
	`vrrp {`
	`  vrrp-group 15 {`
	`  advertise-interval 1`
	`  preempt true`
	`  sync-group test`
	`  virtual-address 172.16.1.25`
	`  }`
	`}`

Using VRRP with a zone-based firewall

When a physical interface or virtual interface has a VRRP interface defined, all incoming traffic arrives through the VRRP interface. Zone-based firewalls drop all traffic in and out unless explicitly allowed. Therefore, if you are using VRRP interfaces with a zone-based firewall, you must make sure you include the VRRP interfaces in your zone.

To use VRRP interface in a zone you must attach the physical interface on which VRRP is enabled. The configuration is the same as zone configuration on a physical interface, the only difference is that VRRP is running on this interface.

Enabling firewall denial of service protection

To configure firewall denial of service protection, perform the steps in the following examples in configuration mode.

> **Note:** The vRouter software automatically calculates the rate-limit interval from the rate and burst values as follows: interval (milliseconds) = (burst*1000)/rate.

Example 1: Limit only inbound max-halfopen TCP sessions

Complete the following steps to limit only the inbound max-halfopen TCP sessions on the dp0p1s1 interface:

1. Configure the dp0p1s1 data plane interface and assign FW1 as the inbound firewall:

```
vyatta@R1# set interfaces dataplane dp0p1s1 address 10.10.1/24
vyatta@R1# set interfaces dataplane dp0p1s1 firewall in FW1
```

2. Configure the dp0p1s2 data plane interface:

```
vyatta@R1# set interfaces dataplane dp0p1s2 address 10.10.1/24
```

3. Configure FW1 as the firewall for the configuration:

```
vyatta@R1# set security firewall name FW1 rule 10 action accept
```

4. Configure the firewall rule to be stateful:

```
vyatta@R1# set security firewall name FW1 rule 10 session
```

5. Configure the system session limit parameter name as MAX_HALFOPEN_200 and set the limit to a maximum of 200 half-open sessions:

```
vyatta@R1# set system session limit parameter name MAX_HALFOPEN_200 max-halfopen 200
```

6. Configure PROTOTCP as the system session group name for the dp0p1s1 interface:

```
vyatta@R1# set system session limit group name PROTOTCP interface dp0p1s1
```

> **Note:** The session limiter is configured on the dp0p1s1 interface, which means it is applied to both inbound and outbound sessions created on that interface. However, because there is only an inbound firewall on dp0p1s1 the session limiter works only with inbound sessions.

7. Configure the rule parameters for PROTOTCP:

```
vyatta@R1# set system session limit group name PROTOTCP rule 10 parameter MAX_HALFOPEN_200
```

8. Configure the rule protocol for PROTOTCP:

```
vyatta@R1# set system session limit group name PROTOTCP rule 10 protocol tcp
```

9. Save the configuration:

```
vyatta@R1# commit
```

10. Display the configured firewall DoS protection:

```
vyatta@R1# show session limit parameter MAX_HALFOPEN_200
Session limit parameter "MAX_HALFOPEN_200":
    Sessions allowed                                                          200
    Sessions blocked                                                          100
    Current session counts (estab/half-open/terminating)                  [0:200:0]
    Max session counts (estab/half-open/terminating)                      [0:200:0]
    Time since last session created                                           23.0s
    Sessions per sec avg (1sec/1min/5mins)                                  [0:0:0]
    Max sessions per sec avg (1sec/1min/5mins)                              [0:0:0]
    Time since max sessions per sec (1sec/1min/5mins)             [never:never:never]
    Time since last session blocked                                           23.0s
    Max sessions blocked per sec avg (1sec/1min/5mins)                      [0:0:0]
    Features                                                             max-halfopen
    Max half-open sessions
        Maximum                                                               200
        Sessions blocked                                                      100

Session limit group "PROTOTCP":
    Active on (dp0p1s1)
    rule    parameter           proto           allowed         blocked
    ----    ---------           -----           -------         -------
    10      MAX_HALFOPEN_200    tcp             200             100
    condition - proto tcp
```

Example 2: Rate-limit sessions for different types of protocols while maintaining separate counts for each protocol

Complete the following steps to rate-limit TCP, UDP, and ICMP sessions with a single rate-limit parameter, while maintaining separate counts for each protocol.

1. Configure the dp0p1s1 data plane interface and assign FW1 as the inbound firewall:

```
vyatta@R1# set interfaces dataplane dp0p1s1 address 10.10.1/24
vyatta@R1# set interfaces dataplane dp0p1s1 firewall in FW1
```

2. Configure FW1 as the firewall for the configuration:

```
vyatta@R1# set security firewall name FW1 rule 10 action accept
```

3. Configure the firewall rule to be stateful:

```
vyatta@R1# set security firewall name FW1 rule 10 session
```

4. Configure the system session limit parameter name as PARAM1 and set the rate limit to 4 sessions:

```
vyatta@R1# set system session limit parameter name PARAM1 rate-limit 4
```

5. Configure GROUP1 as the system session group name for the dp0p1s1 interface:

```
vyatta@R1# set system session limit group name GROUP1 interface dp0p1s1
```

6. Configure the rule 10 parameters for GROUP1:

```
vyatta@R1# set system session limit group name GROUP1 rule 10 parameter PARAM1
```

7. Configure the rule protocol to UDP for GROUP1:

```
vyatta@R1# set system session limit group name GROUP1 rule 10 protocol udp
```

8. Configure the rule 20 parameters for GROUP1:

```
vyatta@R1# set system session limit group name GROUP1 rule 20 parameter PARAM1
```

9. Configure the rule protocol to TCP for GROUP1:

```
vyatta@R1# set system session limit group name GROUP1 rule 20 protocol tcp
```

10. Configure the rule 30 parameters for GROUP1:

```
vyatta@R1# set system session limit group name GROUP1 rule 30 parameter PARAM1
```

11. Configure the rule protocol to ICMP for GROUP1:

```
vyatta@R1# set system session limit group name GROUP1 rule 30 protocol icmp
```

12. Save the configuration:

```
vyatta@R1# commit
```

13. After sending 100 packets each of UDP, TCP and ICMP (with different ports, source addresses, or both), display the configured firewall DoS protection:

```
vyatta@R1# show session limit parameter PARAM1
Session limit parameter "PARAM1":
    Sessions allowed                                                    111
    Sessions blocked                                                    189
    Current session counts (estab/half-open/terminating)              [0:0:0]
    Max session counts (estab/half-open/terminating)                 [0:74:0]
    Time since last session created                                     1.9m
    Sessions per sec avg (1sec/1min/5mins)                            [0:0:0]
    Max sessions per sec avg (1sec/1min/5mins)                        [4:0:0]
    Time since max sessions per sec (1sec/1min/5mins)     [1.9m:never:never]
    Time since last session blocked                                     1.9m
    Max sessions blocked per sec avg (1sec/1min/5mins)                [7:0:0]
    Features                                                       rate-limit
    Rate limit
        Rate sessions/second                                              4
        Max burst                                                         4
        Interval (milliseconds)                                        1000
        Sessions blocked                                                189

Session limit group "GROUP1":
    Active on (dp0p1s1)
    rule    parameter  proto          allowed         blocked
    ----    ---------  -----          -------         -------
    10      PARAM1     udp            37              63
    condition - proto udp
```

```
    20      PARAM1     tcp             37              63
    condition - proto tcp

    30      PARAM1     icmp            37              63
    condition - proto icmp
```

Viewing firewall information

This section describes how to display active firewalls applied to interfaces and zones.

Showing active firewall rule sets

You can see active firewall rule sets by using the **`show firewall`** *interface* command in operational mode and specifying the name of an interface. If no interface is specified, then all firewall rule sets for all interfaces are displayed.

The following example shows how to display information for all interfaces.

```
vyatta@R1:~$ show firewall

------------------------------------
Rulesets Information: Firewall
------------------------------------
-------------------------------------------------------------------------------
Firewall "fw_1":
Active on (dp0p192p1, in)
rule    action  proto   packets         bytes
----    ------  -----   -------         -----
1       allow   tcp     0               0
  condition - stateful proto tcp flags S/FSRA all

8       allow   any     0               0
  condition - stateful to 20.20.20.0/24
```

Showing firewall configuration on interfaces

You can view firewall information in configuration nodes by using the `show` command in configuration mode. The following example shows how to display firewall configuration in configuration mode.

```
vyatta@R1# show security firewall

    name FWTEST-1 {
        rule 1 {
            action accept
            source {
                address 172.16.0.26
            }
        }
    }
    name FWTEST-2 {
        rule 1 {
            action accept
            destination {
                address 10.10.40.101
            }
            source {
                address 10.10.30.46
            }
        }
    }
    name FWTEST-3 {
```

```
        rule 1 {
            action accept
            destination {
                port telnet
            }
            protocol tcp
            source {
                address 10.10.30.46
            }
        }
    }
    name FWTEST-4 {
        rule 1 {
            action accept
            destination {
                address 172.16.0.0/24
            }
            source {
                address 10.10.40.0/24
            }
        }
    }
    vyatta@R1#
```

Global Firewall Commands

clear firewall

Clears firewall statistics.

Syntax:
`clear firewall [ bridge ]`

bridge
Specifies clearing firewall bridge statistics only.

Operational mode

Use this command to clear firewall statistics.

show firewall

Displays statistics for a firewall rule set for an interface or for all firewall rule sets.

Syntax:
`show firewall [ interface ]`

When used with no option, the command shows information for all configured firewall rule sets.

interface
A type of interface. For more information about the supported interface name formats, refer to Supported Interface Types *(page 102)*.

Operational mode

Use this command to display statistics about configured firewall rule sets.

The following example shows how to display statistics for firewall rule sets.

```
vyatta@R1# show firewall
-------------------------------------
Rulesets Information: Firewall
-------------------------------------
--------------------------------------------------------------------------------
Firewall "fw_1":
Active on (dp0p192p1, in)
rule    action  proto   packets         bytes
----    ------  -----   -------         -----
1       allow   tcp     0               0
  condition - stateful proto tcp flags S/FSRA all
8       allow   any     0               0
  condition - stateful to 20.20.20.0/24
```

Firewall Commands

clear session limit

Clears session related data.

Syntax:
`clear session limit [ group | parameter parameter ]`

group
Clears session limit group information.
parameter
Clears session limit parameter information for the specified parameter.

Operational mode

Use this command to clear session related data.

Clears maximum half-open, established, and terminating counts; maximum 1s, 1m, and 5m rates; maximum 1s, 1m, and 5m drops; rate-limit blocked counts, and half-open blocked counts. If "group" is specified, the per-rule allowed and blocked counts are reset. There is no option to clear specific groups.

Each session limit parameter maintains counts for sessions in the New, Established, and Terminating states. These sessions are protocol dependent. The table below shows how the state is determined for the four main session protocol types.

Note: There is no Terminating state equivalent for UDP, ICMP Echo, and so on.

	New	Established	Terminating
TCP	syn-sent, simsyn-sent, syn-received	Established	fin-sent, fin-received, close-wait, fin-wait, closing, last-ack, time-wait
UDP	One or more packets in forward direction	One or more packets seen in each direction	Not applicable
ICMP echo	Echo request in forward direction	Echo reply in backward direction	Not applicable
Other	One or more packets in forward direction	One or more packets seen in each direction	Not applicable

interfaces dataplane <interface> firewall local <ruleset>

Enables control plane policing (CPP) on a data plane interface by applying a firewall instance or rule set.

Syntax:
`set interfaces dataplane interface firewall local ruleset`

Syntax:
`delete interfaces dataplane interface firewall local ruleset`

Syntax:
`show interfaces dataplane interface firewall local ruleset`

interface
The name of a data plane interface.

ruleset
A firewall rule set to be applied when packets are received on the interface and are destined to the vRouter.

Configuration mode

```
interfaces {
    dataplane interface {
        firewall {
            local ruleset
        }
    }
}
```

Use this command to enable CPP on a data plane interface by applying a firewall instance or rule set.

CPP has no effect on traffic that is traversing the vRouter or destined to the vRouter until the firewall rule set has been applied to the data plane by using this command.

To use CPP, you must first define a firewall rule set as a named firewall instance and then apply the firewall instance to a data plane interface by using this command. After the firewall instance or rule set is applied to the **local** keyword, the firewall is enabled to filter packets that are destined for the system itself.

Use the `set` form of this command to enable CPP on a data plane interface.

Use the `delete` form of this command to disable CPP on a data plane interface.

Use the `show` form of this command to display CPP configuration on a data place interface.

interfaces loopback <interface> firewall local <ruleset>

Applies a firewall rule set to a loopback interface.

Syntax:
`set interfaces loopback` *interface* **`firewall local`** *ruleset*

Syntax:
`delete interfaces loopback` *interface* **`firewall local`** *ruleset*

Syntax:
`show interfaces loopback` *interface* **`firewall local`**

interface
The name of a loopback interface. The value of this parameter is **`lo`**.
`local` ***ruleset***
Applies the ruleset for packets destined to the vRouter arriving on any interface.

Configuration mode

```
interfaces {
    loopback lo {
        firewall {
            local ruleset
        }
    }
}
```

Use this command to apply a firewall rule set to all interfaces.

Note: The use of the lo interface indicates that the rules must be applied on all interfaces, for packets destined for the vRouter.

If an interface also has local rule sets applied directly on the interface, then those rule sets are run first. Only if there is no match will it then run the ones attached to the loopback lo interface.

To use the firewall feature, you must define a firewall rule set as a named firewall instance by using the **security firewall name <name>** command. You then apply the firewall rule set to the loopback interface.

Use the **set** form of this command to apply a firewall rule set to the loopback interface.

Use the **delete** form of this command to delete a firewall rule set from the loopback interface.

Use the **show** form of this command to display the configuration of a firewall ruleset on the loopback interface.

monitor firewall

Monitors firewall activity.

Syntax:
monitor firewall name *firewall-name* [**rule** *rule-number*]

Monitoring applies to all rules for the specified firewall.

firewall-name
Specifies the firewall by name.
rule-number
Restricts monitoring to a rule in the firewall.

Operational mode

Use this command to monitor activity for a specified firewall. Include a firewall rule to limit monitoring to that rule.

The following example shows how to monitor activity for firewall fw1.

```
vyatta@vyatta:~$ monitor firewall name fw1
FIREWALL: fw rule fw1:10000 block tcp(6) src=dp0s10/9e:b0:fb:23:3:8c/10.0.1.1(1000)
 dst=/52:54:0:13:af:c9/10.0.2.1(80) len=40 ttl=64 window=512 res=0x00 SYN urgp=0
FIREWALL: fw rule fw1:10000 block tcp(6) src=dp0s10/9e:b0:fb:23:3:8c/10.0.1.1(1001)
 dst=/52:54:0:13:af:c9/10.0.2.1(80) len=40 ttl=64 window=512 res=0x00 SYN urgp=0
FIREWALL: fw rule fw1:10000 block tcp(6) src=dp0s10/9e:b0:fb:23:3:8c/10.0.1.1(1002)
 dst=/52:54:0:13:af:c9/10.0.2.1(80) len=40 ttl=64 window=512 res=0x00 SYN urgp=0
FIREWALL: fw rule fw1:10000 block tcp(6) src=dp0s10/9e:b0:fb:23:3:8c/10.0.1.1(1003)
 dst=/52:54:0:13:af:c9/10.0.2.1(80) len=40 ttl=64 window=512 res=0x00 SYN urgp=0
FIREWALL: fw rule fw1:10000 block tcp(6) src=dp0s10/9e:b0:fb:23:3:8c/10.0.1.1(1004)
 dst=/52:54:0:13:af:c9/10.0.2.1(80) len=40 ttl=64 window=512 res=0x00 SYN urgp=0
...
^C
vyatta@vyatta:~$
```

security application firewall name <name> description <description>

Provides a description of a firewall application rule set.

Syntax:
set security application firewall name *name* **description** *description*

Syntax:
delete security application firewall name *name* **description** *description*

Syntax:
show security application firewall name *name* **description**

name

The name of a firewall application rule set.

description

A brief description of the application rule set. If the description contains spaces, it must be enclosed in double quotation marks.

Configuration mode

```
security {
        application {
                firewall {
                        name name {
                                description text
                        }
                }
        }
}
```

Use the `set` form of this command to describe the firewall application rule set.

Use the `delete` form of this command to remove the description of the firewall application rule set.

Use the `show` form of this command to display the description.

security application firewall name <name> no-match-action

Defines the no-match action for a firewall application rule set.

Syntax:

`set security application firewall name` *name* **`no-match-action`** { **`accept`** | **`drop`** }

Syntax:

`delete security application firewall name` *name* **`no-match-action`** { **`accept`** | **`drop`** }

Syntax:

`show security application firewall name` *name* **`no-match-action`**

name

The name of a firewall application rule set.

`accept`

Accepts the packet. To be performed when the application does not match any other rule in the rule set.

`drop`

Drops the packet silently. To be performed when the application does not match any other rule in the rule set. This is also the action performed if "no-match-action" is not set for a rule set.

Configuration mode

```
security {
        application {
                firewall {
                        name name {
                                no-match-action {
                                    accept
                                    drop
                                }
                        }
                }
        }
}
```

Use the `set` form of this command to define the no-match action for a firewall application rule set.

Use the `delete` form of this command to delete the no-match action from a firewall application rule set.

Use the `show` form of this command to display the no-match action for a firewall application rule set.

security application firewall name <name> rule <rule-number>

Defines a rule for a firewall application rule set.

Syntax:

```
set security application firewall name name rule rule-number
```

Syntax:

```
delete security application firewall name name rule rule-number
```

Syntax:

```
show security application firewall name name rule rule-number
```

name
The name of a firewall rule set.

rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.

Configuration mode

```
security {
       application {
              firewall {
                     name name {
                            rule rule-number
                     }
              }
       }
}
```

Use this command to define a rule within a firewall application rule set.

A firewall rule set consists as many as 9,999 configurable rules.

To avoid having to renumber firewall rules, a good practice is to number rules in increments of 10. This increment allows room for the insertion of new rules within the rule set.

Use the `set` form of this command to define a rule within a firewall application rule set.

Use the `delete` form of this command to delete a rule from a firewall application rule set.

Use the `show` form of this command to display a rule from a firewall application rule set.

security application firewall name <name> rule <rule-number> action <action>

Defines the actions for a firewall application rule.

Syntax:

```
set security application firewall name name rule rule-number action { accept | drop }
```

Syntax:

```
delete security application firewall name name rule rule-number action { accept | drop }
```

Syntax:
`show security application firewall name` *name* **`rule`** *rule-number* **`action`**

name
The name of a firewall application rule set.
rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.
`accept`
Accepts the packet when it satisfies the match criteria.
Exactly one action must be specified.
`drop`
Drops the packet silently when it satisfies the match criteria.
Exactly one action must be specified.

Configuration mode

```
security {
       application {
              firewall {
                     name name {
                            rule rule-number {
                                   action {
                                      accept
                                      drop
                                   }
                            }
                     }
              }
       }
}
```

Use the `set` form of this command to define the action for a firewall application rule.
Use the `delete` form of this command to delete the action from a firewall application rule.
Use the `show` form of this command to display the action for a firewall application rule set.

security application firewall name <name> rule <rule-number> description <description>

Provides a brief description of a firewall application rule.

Syntax:
`set security application firewall name` *name* **`rule`** *rule-number* **`description`** *description*

Syntax:
`delete security application firewall name` *name* **`rule`** *rule-number* **`description`** *description*

Syntax:
`show security application firewall name` *name* **`rule`** *rule-number* **`description`**

name
The name of a firewall application rule set.
rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.
description

A brief description of the rule. If the description contains spaces, it must be enclosed in double quotation marks.

Configuration mode

```
security {
        application {
                firewall {
                        name name {
                                rule rule-number {
                                        description description
                                }
                        }
                }
        }
}
```

Use the `set` form of this command to provide a brief description of a firewall application rule.

Use the `delete` form of this command to delete the description of a firewall application rule.

Use the `show` form of this command to display the description of a firewall application rule.

security application firewall name <name> rule <rule-number> name <app-name>

Specifies match by application name for a firewall application rule.

Syntax:

`set security application firewall name` *name* **`rule`** *rule-number* **`name`** *app-name*

Syntax:

`delete security application firewall name` *name* **`rule`** *rule-number* **`name`** *app-name*

Syntax:

`show security application firewall name` *name* **`rule`** *rule-number* **`name`**

name

The name of a firewall rule set.

rule-number

The numeric identifier of a rule. The identifier ranges from 1 through 9999.

app-name

The name of an application. You can configure a single application name to be matched from a list of DPI engine applications at the most granular level.

Configuration mode

```
security {
        application {
                firewall {
                        name name {
                                rule rule-number {
                                        name app-name
                                }
                        }
                }
        }
}
```

You can specify a application name match for a firewall rule in this command, or specify a match by protocol using the **security application firewall name <name> rule <rule-number> protocol <protocol>** command. Use a protocol rule if you want to match any applications that use that protocol, and use an application rule if you want to match only a specific named application.

Use the `set` form of this command to specify match by application name for a firewall application rule.

Use the `delete` form of this command to delete match by application name for a firewall application rule.

Use the `show` form of this command to display the match criterion for a firewall application rule.

security application firewall name <name> rule <rule-number> protocol <protocol>

Specifies match by application protocol for a firewall application rule.

Syntax:
`set security application firewall name` *name* **`rule`** *rule-number* **`protocol`** *protocol*

Syntax:
`delete security application firewall name` *name* **`rule`** *rule-number* **`protocol`** *protocol*

Syntax:
`show security application firewall name` *name* **`rule`** *rule-number* **`protocol`**

name
The name of a firewall rule set.
rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.
protocol
Matches packets by protocol. A protocol is the name of an application that runs directly over UDP or TCP. You can configure a single protocol name to be matched from a list of DPI engine applications at the most granular level.

Configuration mode

```
security {
       application {
              firewall {
                     name name {
                            rule rule-number {
                                   protocol protocol
                            }
                     }
              }
       }
}
```

You can specify a protocol match for a firewall rule in this command, or specify a match by application name using the **security application firewall name <name> rule <rule-number> name <app-name>** command. Use a protocol rule if you want to match any applications that use that protocol, and use an application rule if you want to match only a specific named application.

Use the `set` form of this command to specify match by application protocol for a firewall application rule.

Use the `delete` form of this command to delete match by application protocol for a firewall application rule.

Use the `show` form of this command to display application protocol match for a firewall application rule.

security application firewall name <name> rule <rule-number> type <type>

Specifies match by application type for a firewall application rule.

Syntax:
```
set security application firewall name name rule rule-number type type
```

Syntax:
```
delete security application firewall name name rule rule-number type type
```

Syntax:
```
show security application firewall name name rule rule-number type
```

name
The name of a firewall rule set.

rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.

type
Matches packets by application type. The application type provides access to less granular groups of DPI classifications such as analytics, database, and social networking. An application can have multiple application types. You can configure a single application type to be matched from a list of DPI application types at the most granular level.

Configuration mode

```
security {
       application {
              firewall {
                     name name {
                            rule rule-number {
                                   type type
                            }
                     }
              }
       }
}
```

Use the `set` form of this command to specify match by application type for a firewall application rule.

Use the `delete` form of this command to delete match by application type for a firewall application rule.

Use the `show` form of this command to display the application type match for a firewall application rule.

security firewall all-ping <state>

Enables or disables responses to all ICMP echo request (ping) messages.

Syntax:
```
set security firewall all-ping { disable | enable }
```

Syntax:
```
delete security firewall all-ping [ disable | enable ]
```

Syntax:
```
show security firewall all-ping
```

Responses to ICMP echo request messages are enabled.

disable
Disables responses to ICMP echo request messages.

enable
Enables responses to ICMP echo request messages.

Configuration mode

```
security {
    firewall {
        all-ping
            disable
            enable
    }
}
```

Use this command to specify whether the system responds to ICMP echo request messages (pings). These messages include all ping messages: unicast, broadcast, or multicast.

Pings are a network tool that help establish the reachability of a device from the local system. Pings are often disallowed because they are a potential means of denial of service (DoS) attacks.

Use the `set` form of this command to enable or disable responses to pings.

Use the `delete` form of this command to restore the default behavior of responding to pings.

Use the `show` form of this command to display the state of responding to pings.

security firewall broadcast-ping <state>

Enables or disables response to broadcast ICMP echo request and time-stamp request messages.

Syntax:
```
set security firewall broadcast-ping { disable | enable }
```

Syntax:
```
delete security firewall broadcast-ping [ disable | enable ]
```

Syntax:
```
show security firewall broadcast-ping
```

ICMP echo and time-stamp request messages do not receive responses.

disable
Disables responses to broadcast ICMP echo and time-stamp request messages.

enable
Enables responses to broadcast ICMP echo and time-stamp request messages.

Configuration mode

```
security {
    firewall {
        broadcast-ping
            disable
            enable
    }
}
```

Use this command to specify whether the system responds to broadcast ICMP echo request and broadcast ICMP time-stamp request messages.

Pings are a network tool that help establish the reachability of a device from the local system. Pings, particularly broadcast pings, are often disallowed because they are a potential means for denial of service (DoS) attacks. Time-stamp requests are used to query another device for the current date and time. Time-stamp requests are

also often disallowed both because they are a potential means for a DoS attack and because the query allows an attacker to learn the date set on the queried machine.

Use the `set` form of this command to specify whether the system responds to broadcast ICMP ICMP echo and time-stamp request messages.

Use the `delete` form of this command to restore the default behavior of not responding to broadcast ICMP ICMP echo and time-stamp request messages.

Use the `show` form of this command to display the behavior to broadcast ICMP ICMP echo and time-stamp request messages.

security firewall config-trap <state>

Enables the generation of Simple Network Message Protocol (SNMP) traps regarding firewall configuration changes.

Syntax:

```
set security firewall config-trap { disable | enable }
```

Syntax:

```
delete security firewall config-trap [ disable | enable ]
```

Syntax:

```
show security firewall config-trap
```

Disabled.

disable
: Disables the generation of SNMP traps regarding a firewall configuration change.

enable
: Enables the generation of SNMP traps regarding a firewall configuration change.

Configuration mode

```
security {
    firewall {
        config-trap
            disable
            enable
    }
}
```

A device uses SNMP traps to notify, without solicitation, the manager of the device about significant events, such as firewall configuration changes.

Use the `set` form of this command to enable the generation of SNMP traps when a firewall configuration change is made.

Use the `delete` form of this command to restore the default behavior.

Use the `show` form of this command to display the state regarding the generation of SNMP traps on firewall configuration changes.

security firewall global-state-policy <protocol>

Configures the global state parameters for firewall.

Syntax:

```
set security firewall global-state-policy { icmp | tcp | udp }
```

Syntax:

```
delete security firewall global-state-policy [ icmp | tcp | udp ]
```

Syntax:

```
show security firewall global-state-policy
```

If this statement is not configured, the firewall is stateless. In this case, specific rules governing statefulness can be configured within the rule set.

icmp
Enable ICMP state monitoring for firewall.

tcp
Enable TCP state monitoring for firewall.

udp
Enable UDP state monitoring for firewall.

Configuration mode

```
security {
    firewall {
        global-state-policy {
            icmp
            tcp
            udp
        }
    }
}
```

Setting this configuration node makes the firewall globally stateful.

When configured to be stateful, the firewall tracks the state of network connections and traffic flows and allows or restricts traffic based on whether its connection state is known and authorized. For example, when an initiation flow is allowed in one direction, the stateful firewall automatically allows responder flows in the return direction.

The statefulness policy that is configured applies to all IPv4 and IPv6 traffic, traversing the interface that the rule set is attached to. After the firewall is configured to be globally stateful, this setting overrides any state rules configured within rule sets.

Use the `set` form of this command to configure a global statefulness policy for firewall.

Use the `delete` form of this command to delete a global statefulness policy for firewall.

Use the `show` form of this command to display a global statefulness policy for firewall.

security firewall name <name> default-action <action>

Defines the default action for a firewall rule.

Syntax:

```
set security firewall name name default-action { accept | drop }
```

Syntax:

```
delete security firewall name name default-action [ accept | drop ]
```

Syntax:

```
show security firewall name name default-action
```

name
Multi-node. The name of a firewall rule set. The name must not contain a space or any other of the following special characters: |, ;, &, $, <, or >. The name can be as many as 28 characters long.

You can define more than one firewall rule set by creating more than one `name` configuration node.

accept

Accepts the default action for the specified rule set.

drop

Denies the default action for the specified rule set.

Configuration mode

```
security {
    firewall {
        name name {
            default-action
                accept
                drop
        }
    }
}
```

A firewall rule set is a named collection of as many as 9,999 packet-filtering rules. If default-action is not set, or is set to drop, then an implicit rule performs the drop. If default-action is set to accept, then a default rule is added to the end of the rule set that matches all packets and has action accept.

Use the `set` form of this command to define an IP firewall rule.

Use the `delete` form of this command to delete a firewall rule.

Use the `show` form of this command to display a firewall rule.

security firewall name <name> default-log

Defines an IP firewall rule set to log packets that reach the default action.

Syntax:

```
set security firewall name name default-log
```

Syntax:

```
delete security firewall name name default-log
```

Syntax:

```
show security firewall name name default-log
```

name

Multi-node. The name of a firewall rule set. The name must not contain a space or any other of the following special characters: |, ;, &, $, <, or >. The name can be as many as 28 characters long.

You can define more than one firewall rule set by creating more than one `name` configuration node.

Configuration mode

```
security {
    firewall {
        name name {
            default-log
        }
    }
}
```

Use this command to specify that the default action will be logged.

A firewall rule set is a named collection of as many as 9999 packet-filtering rules. Following the numbered rules may be a hidden rule, 10000, which can be set to deny or accept all traffic. There are a set of implicit actions that may be applied if rule 10000 is not present. These actions do not occur if rule 10000 is present, and do not occur if **default-log** or **default-action** is specified. See the Implicit Action *(page 11)* section in this guide.

If a **default-log** action is applied to a rule set but the default action for the firewall has not been configured, the default action for logging is to drop packets that do not match any rule. To have packets that match a default

log rule logged and accepted, you must configure **default-action** as **accept**. Refer to the security firewall name <name> default-action <action> *(page 52)* command.

If multiple rule sets are applied to an interface, and the first rule set makes use of **default-log** or **default-action**, subsequent rules for the interface are not processed (they are ignored).

Use the `set` form of this command to enable logging for the default action.

Use the `delete` form of this command to disable logging for the default action.

Use the `show` form of this command to display the default logging configuration for the rule set.

security firewall name <name> description <description>

Provides a brief description for a firewall rule set.

Syntax:
`set security firewall name` *name* **`description`** *description*

Syntax:
`delete security firewall name` *name* **`description`** *description*

Syntax:
`show security firewall name` *name* **`description`**

name
The name of a firewall rule set.

description
A brief description of the rule set. If the description contains spaces, it must be enclosed in double quotation marks.

Configuration mode

```
security {
    firewall {
        name name {
            description description
        }
    }
}
```

Providing a description for a firewall rule set can help you to quickly determine the purpose of the rule set when viewing the configuration.

Use the `set` form of this command to provide brief description of a firewall rule set.

Use the `delete` form of this command to delete a description.

Use the `show` form of this command to display a description.

security firewall name <name> rule <rule-number>

Defines a rule for a firewall rule set.

Syntax:
`set security firewall name` *name* **`rule`** *rule-number*

Syntax:
`delete security firewall name` *name* **`rule`** *rule-number*

Syntax:
`show security firewall name` *name* **`rule`**

name
The name of a firewall rule set.

rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.

Configuration mode

```
security {
    firewall {
        name name {
            rule rule-number
        }
    }
}
```

Use this command to define a rule within a firewall rule set.

A firewall rule set consists as many as 9,999 configurable rules.

To avoid having to renumber firewall rules, a good practice is to number rules in increments of 10. This increment allows room for the insertion of new rules within the rule set.

Use the `set` form of this command to define a firewall rule within a firewall rule set.

Use the `delete` form of this command to delete a rule from a firewall rule set.

Use the `show` form of this command to display a rule from a firewall rule set.

security firewall name <name> rule <rule-number> action <action>

Defines the action for a firewall rule.

Syntax:
`set security firewall name` *name* **`rule`** *rule-number* **`action`** { **`accept`** | **`drop`** }

Syntax:
`delete security firewall name` *name* **`rule`** *rule-number* **`action`** { **`accept`** | **`drop`** }

Syntax:
`show security firewall name` *name* **`rule`** *rule-number* **`action`**

name
The name of a firewall rule set.

rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.

`accept`
Accepts the packet when it satisfies the match criteria.

Exactly one action must be specified.

`drop`
Drops the packet silently when it satisfies the match criteria.

Exactly one action must be specified.

Configuration mode

```
=security {
        firewall {
                name name {
                        rule rule-number {
                                action accept
                                action drop
```

```
                        }
                }
        }
}
```

Use the `set` form of this command to define an action for a firewall rule within a firewall rule set.

Use the `delete` form of this command to delete an action for a rule from a firewall rule set.

Use the `show` form of this command to display an action for a rule from a firewall rule set.

security firewall name <name> rule <rule-number> description <description>

Provides a brief description for a firewall rule.

Syntax:
`set security firewall name` *name* **`rule`** *rule-number* **`description`** *description*

Syntax:
`delete security firewall name` *name* **`rule`** *rule-number* **`description`**

Syntax:
`show security firewall name` *name* **`rule`** *rule-number*

name
The name of a firewall rule set.

rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.

description
A brief description of the rule. If the description contains spaces, it must be enclosed in double quotation marks.

Configuration mode

```
security {
    firewall {
        name name {
            rule rule-number {
                description description
            }
        }
    }
}
```

Providing a description for a firewall rule can help you to quickly determine the purpose of the rule when viewing the configuration.

Use the `set` form of this command to provide a brief description of a firewall rule.

Use the `delete` form of this command to delete the description of a firewall rule.

Use the `show` form of this command to display the description of a firewall rule.

security firewall name <name> rule <rule-number> destination <destination>

Defines the destination address, MAC address, or destination port for a firewall rule.

Syntax:

```
set security firewall name name rule rule-number destination { address address | mac-address address
| port port }
```

Syntax:

```
delete security firewall name name rule rule-number destination [ address | mac-address | port ]
```

Syntax:

```
show security firewall name name rule rule-number destination
```

name
The name of a firewall rule set.

rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.

`address` ***address***
Specifies a destination address to match. Address formats are as follows:

ip-address: An IPv4 address.

ip-address/prefix: A network address, where 0.0.0.0/0 matches any network.

`!` *ip-address*: All IP addresses except the one specified.

`!` *ip-address/prefix*: All network addresses except the one specified.

ipv6-address: An IPv6 address; for example, fe80::20c:29fe:fe47:f89.

ip-address/prefix: A network address, where ::/0 matches any network; for example, fe80::20c:29fe:fe47:f88/64.

`!` *ipv6-address*: All IP addresses except the one specified.

`!` *ip-address/prefix*: All network addresses except the one specified.

address-group: The name of an address group containing a list of addresses to match.

When both an address and a port are specified, the packet is considered a match only if both the address and the port match.

`mac-address` ***address***
Matches the media access control (MAC) address in the source address. The address format is six 8-bit numbers, separated by colons, in hexadecimal; for example, 00:0a:59:9a:f2:ba.

`port` ***port***
Specifies a destination port to match. Port formats are as follows:

port-name: The name of an IP service; for example, `http`. You can specify any service name in the `/etc/services` file.

port-number: A port number. The number ranges from 1 through 65535.

start-end: A range of ports; for example, 1001-1005.

port-group: The name of a port group containing a list of ports to match.

When both an address and a port are specified, the packet is considered a match only if both the address and the port match.

Configuration mode

```
security {
    firewall {
        name name {
            rule rule-number
                destination {
                    address address
                    mac-address address
                    port port
            }
        }
    }
```

```
}
```

Use the `set` form of this command to define a destination address, MAC address, or destination port within a firewall rule.

Use the `delete` form of this command to delete a destination address, MAC address, or destination port from a firewall rule.

Use the `show` form of this command to display a destination address, MAC address, or destination port from a firewall rule.

security firewall name <name> rule <rule-number> disable

Disables the specified firewall rule.

Syntax:

```
set security firewall name name rule rule-number disable
```

Syntax:

```
delete security firewall name name rule rule-number disable
```

Syntax:

```
show security firewall name name rule rule-number
```

The rule is enabled.

name
The name of a firewall rule set.

rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.

Configuration mode

```
security {
    firewall {
        name name {
            rule rule-number {
                disable
            }
        }
    }
}
```

Use this command to disable a firewall rule. Disabling a firewall rule is a useful way to test how the firewall performs minus a specific rule without having to delete and then re-enter the rule.

Use the `set` form of this command to disable a firewall rule

Use the `delete` form of this command to delete a firewall rule.

Use the `show` form of this command to display a firewall rule.

security firewall name <name> rule <rule-number> dscp <value>

Specifies the Differentiated Services Code Point (DSCP) value for a firewall rule.

Syntax:

```
set security firewall name name rule rule-number dscp value
```

Syntax:

```
delete security firewall name name rule rule-number dscp
```

Syntax:
`show security firewall name` *name* **`rule`** *rule-number* **`dscp`**

`dscp` ***value***
Specifies the DSCP value to match in the incoming IP header. For the value, enter one of the following:

number: A DSCP number ranges from 0 through 63. DSCP matches packets with headers that include this DSCP value. If this option is not set, the DSCP field retains its original value.

classifier: The traffic classifier for the per-hop behavior defined by the DS field in the IP header.

- **`default`**: The Default Class (00000) for best-effort traffic.
- **`af`** *number*: The Assured Forwarding Class for assurance of delivery as defined in RFC 2597. Depending on the forwarding class and the drop precedence, the class can be one of the following values: **`af11`** through **`af13`**, **`af21`** through **`af23`**, **`af31`** through **`af33`**, or **`af41`** through **`af43`**.
- **`cs`** *number*: Class Selector for network devices that use the Precedence field in the IPv4 header. The number ranges from 1 to 7 and indicates the precedence, for example cs1.
- **`ef`**: Expedited Forwarding, per-hop behavior.
- **`va`**: Voice Admit, Capacity-Admitted Traffic.

Configuration mode

```
security {
    firewall {
        name name {
            rule rule-number {
                dscp value
            }
        }
    }
}
```

Use the `set` form of this command to define the DSCP value to match.

Use the `delete` form of this command to delete the DSCP value.

Use the `show` form of this command to display the DSCP value for a firewall rule.

security firewall name <name> rule <rule-number> ethertype <type>

Specifies the Ethernet type for a firewall rule.

Syntax:
`set security firewall name` *name* **`rule`** *rule-number* **`ethertype`** *type*

Syntax:
`delete security firewall name` *name* **`rule`** *rule-number* **`ethertype`**

Syntax:
`show security firewall name` *name* **`rule`** *rule-number* **`ethertype`**

By default, the firewall allows the transmission of known Ethernet-type packets in the network.

`ethertype` ***type***
Specifies matching for the Ethernet type.

type: The Ethernet type; for example, IPv4. You can specify any Ethernet name listed in the **/etc/ethertypes** file. You can also enter the hexadecimal or decimal value for the Ethernet type.

Configuration mode

```
security {
    firewall {
        name name {
            rule rule-number {
                ethertype type
            }
        }
    }
}
```

Use this command to configure the firewall to accept or drop specified types of Ethernet packets.

After you define a firewall rule set with the Ethernet type, you must apply it to an interface as a packet filter by using the firewall-related interface commands. Until you apply a firewall rule set to an interface, the set has no effect on traffic destined for or traversing the system.

Use the `set` form of this command to define the Ethernet type to match.

Use the `delete` form of this command to delete the Ethernet type.

Use the `show` form of this command to display the Ethernet type for a firewall rule.

security firewall name <name> rule <rule-number> fragment

Defines fragmented packets for a firewall rule.

Syntax:

```
set security firewall name name rule rule-number fragment
```

Syntax:

```
delete security firewall name name rule rule-number fragment
```

Syntax:

```
show security firewall name name rule rule-number [ fragment ]
```

name
The name of a firewall rule.

rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.

`fragment`
Specifies matching for fragmented packets. This option only works for rule sets applied to bridges (l2 direction) or QoS. It does not work elsewhere, as IPv4 and IPv6 fragments are reassembled before being processed by the firewall.

Configuration mode

```
security {
    firewall {
        name name {
            rule rule-number
                fragment
        }
    }
}
```

Use the `set` form of this command to define the matching of fragmented packets within a firewall rule.

Use the `delete` form of this command to delete the matching of fragmented packets from a firewall rule.

Use the `show` form of this command to display the matching of fragmented packets from a firewall rule.

security firewall name <name> rule <rule-number> icmp

Specifies an IPv4 ICMP type number, code number, name, or group for a firewall rule.

Syntax:

```
set security firewall name name rule rule-number icmp { type number [ code number ] | name name | group group }
```

Syntax:

```
delete security firewall name name rule rule-number icmp [ type [ number code ] | name | group ]
```

Syntax:

```
show security firewall name name rule rule-number icmp [ type [ number code ] | name | group ]
```

name
The name of a firewall rule set.

rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.

type *number*
Specifies matching for numeric ICMP types. Types range from 0 through 255; for example, 8 (echo request) or 0 (echo Reply). For a list of ICMP codes and types, refer to ICMP Types *(page 97)*.

code *number*
Specifies matching for numeric ICMP codes. Codes range from 0 through 255. For a list of ICMP codes and types, refer to ICMP Types *(page 97)*.

name *name*
Specifies matching for ICMP type names. For a list of ICMP codes and types, refer to ICMP Types *(page 97)*.

group *group*
Specifies an IPv4 ICMP group.

Configuration mode

```
security {
    firewall {
        name name {
            rule rule-number {
                icmp {
                    type number {
                        code number
                    }
                    name name
                    group group
                }
            }
        }
    }
}
```

Use the `set` form of this command to define an ICMP firewall rule within a firewall rule.

Use the `delete` form of this command to delete an ICMP firewall rule from a firewall rule.

Use the `show` form of this command to display an ICMP firewall rule from a firewall rule.

security firewall name <name> rule <rule-number> icmpv6

Specifies an IPv6 ICMP type number, code number, name, or group for a firewall rule.

Syntax:

`set security firewall name` *name* **`rule`** *rule-number* **`icmpv6`** { **`type`** *number* [**`code`** *number*] | **`name`** *name* | **`group`** *group* }

Syntax:
`delete security firewall name` *name* **`rule`** *rule-number* **`icmpv6`** [**`type`** [*number* **`code`**] | **`name`** | **`group`**]

Syntax:
`show security firewall name` *name* **`rule`** *rule-number* **`icmpv6`** [**`type`** [*number* **`code`**] | **`name`** | **`group`**]

name
The name of a firewall rule set.
rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.
`type` ***number***
Specifies matching for numeric ICMPv6 types. Types range from 0 through 255. For a list of ICMPv6 codes and types, refer to ICMPv6 Types *(page 100)*.
`code` ***number***
Specifies matching for numeric ICMPv6 codes. Codes range from 0 through 255. For a list of ICMPv6 codes and types, refer to ICMPv6 Types *(page 100)*.
`name` ***name***
Specifies matching for ICMPv6 type names. For a list of ICMPv6 codes and types, refer to ICMPv6 Types *(page 100)*.
`group` ***group***
Specifies an IPv6 ICMP group.

Configuration mode

```
security {
    firewall {
        name name {
            rule rule-number {
                icmpv6 {
                    type number {
                        code number
                    }
                    name name
                    group group
                }
            }
        }
    }
}
```

Use this command to specify the IPv6 ICMP type within a firewall rule.

Use the `set` form of this command to define an IPv6 ICMP firewall rule within a firewall rule.

Use the `delete` form of this command to delete an IPv6 ICMP firewall rule from a firewall rule.

Use the `show` form of this command to display an IPv6 ICMP firewall rule from a firewall rule.

security firewall name <name> rule <rule-number> ipv6-route type <number>

Specifies the IPv6 route type number for a firewall rule.

Syntax:
`set security firewall name` *name* **`rule`** *rule-number* **`ipv6-route type`** *number*

Syntax:

```
delete security firewall name name rule rule-number ipv6-route type
```

Syntax:

```
show security firewall name name rule rule-number ipv6-route type
```

type *number*

Specifies matching for numeric IPv6 route types. Route types range from 0 through 255.

Configuration mode

```
security {
    firewall {
        name name {
            rule rule-number {
                ipv6-route {
                    type number
                }
            }
        }
    }
}
```

Note: This command can be used to block Type 0 Routing Headers in IPv6. RFC 5095 deprecates the use of Type 0 Routing Headers in IPv6 because they are a security risk.

Use the `set` form of this command to define the IPv6 route type for a firewall rule.

Use the `delete` form of this command to delete the IPv6 route type for a firewall rule.

Use the `show` form of this command to display the IPv6 route type for a firewall rule.

security firewall name <name> rule <rule-number> log

Enables or disables per-packet logging of firewall rule actions. Use only for debugging purposes.

Syntax:

```
set security firewall name name rule rule-number log
```

Syntax:

```
delete security firewall name name rule rule-number log
```

Syntax:

```
show security firewall name name rule rule-number
```

Actions are not logged.

name

The name of a firewall rule set.

rule-number

The numeric identifier of a rule. The identifier ranges from 1 through 9999.

Configuration mode

```
security {
    firewall {
        name name {
            rule rule-number {
                log
            }
        }
    }
```

```
}
```

Use the `set` form of this command to enable or disable logging of firewall rule actions.

Use this type of logging only for debugging purposes. Per-packet logging occurs in the forwarding paths and can greatly reduce the throughput of the system and dramatically increase the disk space used for the log files. For all operational purposes, use stateful session logging instead of per-packet logging (see security firewall session-log <protocol> *(page 74)*).

Use the `delete` form of this command to delete the logging value for a rule.

Use the `show` form of this command to display the logging value for a rule.

security firewall name <name> rule <rule-number> mark <action>

Specifies the DSCP or Priority Code Point (PCP) packet marking action for a firewall rule.

Syntax:

```
set security firewall name name rule rule-number mark { dscp dscp-value | pcp pcp-number }
```

Syntax:

```
delete security firewall name name rule rule-number mark [ dscp | pcp ]
```

Syntax:

```
show security firewall name name rule rule-number mark
```

name

The name of a firewall rule set.

rule-number

The numeric identifier of a rule. The identifier ranges from 1 through 9999.

dscp *dscp-value*

Specifies the DSCP value. For the value, enter one of the following:

number: A DSCP number ranges from 0 through 63. DSCP matches packets with headers that include this DSCP value. If this option is not set, the DSCP field retains its original value.

classifier: The traffic classifier for the per-hop behavior defined by the DS field in the IP header.

- **default**: The Default Class (00000) for best-effort traffic.
- **af** *number*: the Assured Forwarding Class for assurance of delivery as defined in RFC 2597. Depending on the forwarding class and the drop precedence, the class can be one of the following values: **af11** through **af13**, **af21** through **af23**, **af31** through **af33**, or **af41** through **af43**.
- **cs** *number*: Class Selector for network devices that use the Precedence field in the IPv4 header. The number ranges from 1 to 7 and indicates the precedence, for example cs1.
- **ef**: Expedited Forwarding, Per-Hop Behavior.
- **va**: Voice Admit, Capacity-Admitted Traffic.

pcp *pcp-number*

The 802.1 priority-code point number. The number can range from 0 through 7.

Configuration mode

```
security {
    firewall {
        name name {
            rule rule-number {
                mark {
                    dscp dscp-value
                    pcp pcp-number
                }
            }
```

```
        }
    }
}
```

Use the `set` form of this command to define the packet marking action within a firewall rule.

Use the `delete` form of this command to delete the packet marking action within a firewall rule.

Use the `show` form of this command to display the packet marking action within a firewall rule.

security firewall name <name> rule <rule-number> pcp <number>

Specifies the 802.1 Priority Code Point (PCP) to match for a firewall rule.

Syntax:
`set security firewall name` *name* **`rule`** *rule-number* **`pcp`** *pcp-number*

Syntax:
`delete security firewall name` *name* **`rule`** *rule-number* **`pcp`**

Syntax:
`show security firewall name` *name* **`rule`** *rule-number* **`pcp`**

name
The name of a firewall rule set.
rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.
`pcp` ***pcp-number***
The 802.1 priority-code point number. The number can range from 0 through 7.

Configuration mode

```
security {
    firewall {
        name name {
            rule rule-number {
                pcp pcp-number
            }
        }
    }
}
```

Use the `set` form of this command to define the PCP within a firewall rule.

The following notes apply to PCP matching and marking:

- Matching on PCP for a firewall rule should be done only in the "in" direction in L2, because the PCP of a forwarded packet is cleared.
- Marking of the PCP value on outgoing packets in a firewall rule can be done only for bridging (in the L2 direction).
- If a PCP setting is required for routed packets, QoS must be used. Refer to the AT&T Vyatta Network Operating System QoS Configuration Guide for more information.

Use the `delete` form of this command to delete the PCP within a firewall rule.

Use the `show` form of this command to display the PCP within a firewall rule.

security firewall name <name> rule <rule-number> police <limiting-method>

Specifies the type of packet rate limiting method.

Syntax:

```
set security firewall name name rule rule-number police { bandwidth limit | burst size | ratelimit limit | then { action drop | mark { dscp dscp-value | pcp pcp-number } } }
```

Syntax:

```
delete security firewall name name rule rule-number police [ { bandwidth limit | burst size | ratelimit | then { action drop | mark { dscp | pcp } } } ]
```

Syntax:

```
show security firewall name name rule rule-number police [ { bandwidth | burst | ratelimit | then { action | mark } ]
```

The action is to drop packets when rule is matched.

name
The name of a firewall rule set.

rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.

bandwidth ***limit***
The bandwidth rate as a number followed by no space and a scaling suffix representing the rate (for example, 10mbit).

The following suffixes are supported:

No suffix: Kilobits per second.

mbit: Megabits per second.

mbps: Megabytes per second.

gbit: Gigabits per second.

kbps: Kilobytes per second.

gbps: Gigabytes per second.

burst ***limit***
The burst size limit in number of bytes. The number can range from 1 through 312500000.

ratelimit ***limit***
The number of packets that can be sent in a second.

n: Number of packets per second.

*n***kpps**: Thousands of packets per second.

*n***mpps**: Millions packets per second.

dscp ***dscp-value***
Specifies the DSCP number. The supported values are **af11** through **af13**, **af21** through **af23**, **af31** through **af33**, **af41** through **af43**, **cs1** through **cs7**, **default**, **ef**, and **va**.

Packets are marked with the given value if policing is exceeded.

pcp ***pcp-number***
The 802.1 priority-code point number. The number can range from 0 through 7.

Packets are marked with the given value if policing is exceeded.

Configuration mode

```
security {
```

```
    firewall {
        name name {
            rule rule-number {
                police {
                    bandwidth limit
                    burst size
                    then {
                        action drop
                        mark {
                            dscp dscp-value
                            pcp pcp-number
                        }
                    }
                }
            }
        }
    }
}
```

If no **then** action is specified, then the default action is to drop the packet if police limits are exceeded.

Use the `set` form of this command to enable or disable policing of firewall rule actions.

Use the `delete` form of this command to delete the policing value for a rule.

Use the `show` form of this command to display the policing value for a rule.

security firewall name <name> rule <rule-number> protocol

Specifies the protocol to match for a firewall rule.

Syntax:

```
set security firewall name name rule rule-number protocol protocol
```

Syntax:

```
delete security firewall name name rule rule-number protocol
```

Syntax:

```
show security firewall name name rule rule-number protocol
```

protocol *protocol*

Matches packets by protocol. Any protocol literals or numbers listed in the `/etc/protocols` file can be specified.

Configuration mode

```
security {
    firewall {
        name name {
            rule rule-number {
                protocol protocol
            }
        }
    }
}
```

Use the `set` form of this command to define the protocol type to match for a firewall rule.

Use the `delete` form of this command to delete the protocol type to match for a firewall rule.

Use the `show` form of this command to display the protocol type to match for a firewall rule.

security firewall name <name> rule <rule-number> session application firewall <app-firewall>

Specify match by application firewall for a firewall rule within a session.

Syntax:
`set security firewall name` *name* **`rule`** *rule-number* **`session application firewall`** *app-firewall*

Syntax:
`delete security firewall name` *name* **`rule`** *rule-number* **`session application firewall`** *app-firewall*

Syntax:
`show security firewall name` *name* **`rule`** *rule-number* **`session application firewall`**

name
The name of a firewall rule set.
rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.
app-firewall
Matches packets by application firewall. The name of the application firewall is configured by using the **security application firewall name** command.

Configuration mode

```
security {
       firewall {
              name name {
                     rule rule-number {
                            session {
                                   application {
                                          firewall app-firewall

                                   }
                            }
                     }
              }
       }
}
```

Use the `set` form of this command to specify the application firewall to run for a firewall rule within a session.

When this rule is matched, a session will be created and the named application firewall will be run. The application firewall will return either a "match" or "no-match". If "match" is returned, then packets are forwarded for the session, otherwise they are dropped. Note the packets will be forwarded until the DPI function has decided it has enough information to determine the application name.

Use the `delete` form of this command to delete the application firewall to run for a firewall rule within a session.

Use the `show` form of this command to display the application firewall for a firewall rule within a session.

security firewall name <name> rule <rule-number> session application name <app-name>

For a session, specifies match by application name for a firewall application rule.

Syntax:
`set security firewall name` *name* **`rule`** *rule-number* **`session application name`** *app-name*

Syntax:
`delete security firewall name` *name* **`rule`** *rule-number* **`session application name`** *app-name*

Syntax:
`show security firewall name` *name* **`rule`** *rule-number* **`session application name`** *app-name*

name
The name of a firewall rule set.
rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.
app-name
The name of an application. You can configure a single application name to be matched from a list of DPI engine applications at the most granular level.

Configuration mode

```
security {
       firewall {
              name name {
                     rule rule-number {
                            session {
                                   application {
                                          name name
                                   }
                            }
                     }
              }
       }
}
```

Use the `set` form of this command to specify match by application name for a firewall application rule within a session. For an application specified in this command, the rule matches the last application in the path. For a protocol specified in the **security firewall name <name> rule <rule-number> session protocol <protocol>** command, the rule matches the application that comes after TCP/UDP in the path of protocols.

Use the `delete` form of this command to delete match by application name for a firewall application rule within a session.

Use the `show` form of this command to display the application name match for a firewall application rule.

security firewall name <name> rule <rule-number> session application protocol <protocol>

For a session, specifies match by application protocol for a firewall rule.

Syntax:
`set security firewall name` *name* **`rule`** *rule-number* **`session application protocol`** *protocol*

Syntax:
`delete security firewall name` *name* **`rule`** *rule-number* **`session application protocol`** *protocol*

Syntax:
`show security firewall name` *name* **`rule`** *rule-number* **`session application protocol`**

name
The name of a firewall rule set.
rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.
protocol

Matches packets by protocol. A protocol is the name of an application which runs directly over UDP or TCP.

Configuration mode

```
security {
       firewall {
              name name {
                     rule rule-number {
                            session {
                                   application {
                                          protocol protocol
                                   }
                            }
                     }
              }
       }
}
```

Use the `set` form of this command to specify match by application protocol for a firewall rule within a session. For a protocol specified in this command, the rule matches the application that comes after TCP/UDP in the path of protocols. For an application specified in the **security firewall name <name> rule <rule-number> session application name <app-name>** command, the rule matches the last application in the path.

Use the `delete` form of this command to delete match by application protocol for a firewall rule within a session.

Use the `show` form of this command to display application protocol match for a firewall rule within a session.

security firewall name <name> rule <rule-number> session application type <type>

For a session, specifies match by application type for a firewall rule.

Syntax:
`set security firewall name` *name* **`rule`** *rule-number* **`session application type`** *type*

Syntax:
`delete security firewall name` *name* **`rule`** *rule-number* **`session application type`** *type*

Syntax:
`show security firewall name` *name* **`rule`** *rule-number* **`session application type`**

name
The name of a firewall rule set.

rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.

type
Matches packets by application type. The application type provides access to less granular groups of DPI classifications such as analytics, database, social networking. An application can have multiple application types. You can configure a single application type to be matched from a list of DPI engine application types at the most granular level.

Configuration mode

```
security {
       firewall {
              name name {
                     rule rule-number {
                            session {
                                 application {
```

```
                                    type type
                                }
                            }
                        }
                    }
                }
            }
```

Use the `set` form of this command to specify match by application type for a firewall rule within a session. When this rule is matched, a session will be created and the DPI function will try and match on the type of the application. If it matches the application type, then packets are forwarded for the session, otherwise they are dropped. Note the packets will be forwarded until the DPI function has decided it has enough information to determine the application type.

Use the `delete` form of this command to delete match by application type for a firewall rule within a session.

Use the `show` form of this command to display the application type match for a firewall rule within a session.

security firewall name <name> rule <rule-number> source <source>

Defines the source address, MAC address, or source port for a firewall rule.

Syntax:
`set security firewall name` *name* **`rule`** *rule-number* **`source`** { **`address`** *address* | **`mac-address`** *address* | **`port`** *port* }

Syntax:
`delete security firewall name` *name* **`rule`** *rule-number* **`source`** [**`address`** *address* | **`mac-address`** *address* | **`port`** *port*]

Syntax:
`show security firewall name` *name* **`rule`** *rule-number* **`source`**

name
The name of a firewall rule set.

rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.

`address` ***address***
Specifies a source address to match. Address formats are as follows:

ip-address: An IPv4 address.

ip-address/prefix: A network address, where 0.0.0.0/0 matches any network.

`!` *ip-address*: All IP addresses except the one specified.

`!` *ip-address/prefix*: All network addresses except the one specified.

ipv6-address: An IPv6 address; for example, fe80::20c:29fe:fe47:f89.

ipv6-address/prefix: A network address, where ::/0 matches any network; for example, fe80::20c:29fe:fe47:f88/64.

`!` *ipv6-address*: All IP addresses except the one specified.

`!` *ipv6-address/prefix*: All network addresses except the one specified.

address-group: The name of an address group containing a list of addresses to match.

When both an address and a port are specified, the packet is considered a match only if both the address and the port match.

`mac-address` ***address***
Matches the media access control (MAC) address in the source address. The address format is six 8-bit numbers, separated by colons, in hexadecimal; for example, 00:0a:59:9a:f2:ba.

port *port*

Specifies a source port to match. Port formats are as follows:

port-name: The name of an IP service; for example, http. You can specify any service name in the `/etc/services` file.

port-number: A port number. The number ranges from 1 through 65535.

start-end: A range of ports; for example, 1001-1005.

port-group: The name of a port group containing a list of ports to match.

When both an address and a port are specified, the packet is considered a match only if both the address and the port match.

Configuration mode

```
security {
    firewall {
        name name {
            rule rule-number
                source {
                    address address
                    mac-address address
                    port port
                }
        }
    }
}
```

Use the `set` form of this command to define a source address, MAC address, or source port within a firewall rule.

Use the `delete` form of this command to delete a source address, MAC address, or source port from a firewall rule.

Use the `show` form of this command to display a source address, MAC address, or source port from a firewall rule.

security firewall name <name> rule <rule-number> state <state>

Defines whether to match packets related to existing connections for the firewall rule.

Syntax:

```
set security firewall name name rule rule-number state { disable | enable }
```

Syntax:

```
delete security firewall name name rule rule-number state
```

Syntax:

```
show security firewall name name rule rule-number state
```

name

The name of a firewall rule set.

rule-number

The numeric identifier of a rule. The identifier ranges from 1 through 9999.

state

Related packets are packets related to existing connections.

Values for *state* are as follows:

`enable`: Matches related flows.

`disable`: Does not match related flows.

Configuration mode

```
security {
    firewall {
        name name {
            rule rule-number {
                state state
            }
        }
    }
}
```

Use the `set` form of this command to enable or disable stateful processing for the firewall rule.

Use the `delete` form of this command to delete stateful processing of a firewall rule.

Use the `show` form of this command to display stateful processing configuration of a firewall rule.

security firewall name <name> rule <rule-number> tcp flags <flags>

Defines the TCP flags to match for a firewall rule.

Syntax:

```
set security firewall name name rule rule-number tcp flags flags
```

Syntax:

```
delete security firewall name name rule rule-number tcp [ flags flags ]
```

Syntax:

```
show security firewall name name rule rule-number tcp
```

name

The name of a firewall rule set.

rule-number

The numeric identifier of a rule. The identifier ranges from 1 through 9999.

flags

Matches the specified TCP flags in a packet. The keywords are SYN, ACK, FIN, RST, URG, and PSH. You can specify more than one flag, separated by commas, in a list. Prefixing the flag name with the negation operator (!) matches packets with the specified flag unset. For example, the list of SYN, !ACK, !FIN, !RST matches only packets with the SYN flag set and the ACK, FIN, and RST flags unset.

Configuration mode

```
security {
    firewall {
        name name {
            rule rule-number {
                tcp {
                    flags flags
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the TCP flags in a packet of a firewall rule.

Use the `delete` form of this command to delete the TCP flags in a packet of a firewall rule.

Use the `show` form of this command to display the TCP flags in a packet of a firewall rule.

security firewall session-log <protocol>

Specifies the logging that should be performed for selected state changes for the given protocol.

Syntax:

```
set security firewall session-log { icmp icmp-state | other other-state | udp udp-state | tcp tcp-state }
```

Syntax:

```
delete security firewall session-log { icmp | other | udp | tcp }
```

Syntax:

```
show security firewall session-log
```

Session logging is disabled.

icmp-state

Enables Internet Control Message Protocol (ICMP) for messaging for the session log.

- **closed**: Entering the closed state.
- **established**: Entering the established state.
- **new**: Entering the new state.
- **timeout**: Entering the timeout state.

other-state

To use protocols other than TCP, UDP, or ICMP for session logging. Accepts the same parameters as ICMP.

ucp-state

To use User Datagram Protocol (UDP) for session logging. Accepts the same parameters as ICMP.

tcp-state

Enables Transmission Control Protocol (TCP) for session logging.

- **closed-wait**: Entering the closed-wait state.
- **closing**: Entering the closing state.
- **established**: Entering the established state.
- **fin-received**: Entering the fin-received state.
- **fin-sent**: Entering the fin-sent state.
- **fin-wait**: Entering the fin-wait state.
- **last-ack**: Entering the last-ack state.
- **simsyn-sent**: Entering the simsyn-sent state.
- **syn-received**: Entering the syn-received state.
- **syn-sent**: Entering the syn-sent state.
- **time-wait**: Entering the time-wait state.
- **timeout**: Entering the timeout state.

Configuration mode

```
security {
    firewall {
        session-log {
            icmp
               {
                closed
                established
                new
                timeout
                }
            other
```

```
                {
                 closed
                 established
                 new
                 timeout
                 }
             udp
                {
                 closed
                 established
                 new
                 timeout
                 }
             tcp
                {
                 closed-wait
                 closing
                 established
                 fin-received
                 fin-sent
                 fin-wait
                 last-ack
                 simsyn-sent
                 syn-received
                 syn-sent
                 time-wait
                 timeout
                 }
            }
        }
    }
```

Use the `set` form of this command to log packets when entering in the state matching what was configured.

If a stateful firewall rule or a NAT rule is matched in a flow and this command is configured, a log message is generated when the session transitions to the state that is set in the configuration.

Use the `delete` form of this command to delete the logging of transitions into the selected state for the given protocol.

Use the `show` form of this command to display the logging that is enabled for the various protocols.

security firewall syn-cookies

Enables or disables the use of TCP SYN cookies with IPv4.

Syntax:

```
set security firewall syn-cookies  { disable | enable }
```

Syntax:

```
delete security firewall syn-cookies  [ disable | enable ]
```

Syntax:

```
show security firewall syn-cookies
```

If this statement is not configured, then it takes the default of SYN cookies being enabled. When SYN cookies are enabled the Linux kernel will enable a method to defeat SYN flood attacks, otherwise this method is not enabled.

disable
: Disables TCP SYN cookies with IPv4.

enable
: Enables TCP SYN cookies with IPv4.

Configuration mode

```
security {
    firewall {
        syn-cookies {
          enable
          disable
        }
    }
}
```

Use the `set` form of this command to enable or disable TCP SYN cookies with IPv4.

Use the `delete` form of this command to delete the configuration of TCP SYN cookies.

Use the `show` form of this command to display the current setting for TCP SYN cookies.

security firewall tcp-strict

Configures strict checking of TCP state for all stateful rules.

Syntax:
```
set security firewall tcp-strict
```

Syntax:
```
delete security firewall tcp-strict
```

Syntax:
```
show security firewall tcp-strict
```

If this is not configured, then the checking of state for any TCP session is not performed in a strict manner.

`tcp-strict`
Enables strict TCP state checking for all sessions created.

Configuration mode

```
security {
    firewall {
        tcp-strict
    }
}
```

Use the `set` form of this command to enable TCP strict tracking of stateful firewall rules for traffic associated with sessions. This command enables the user to toggle between loose or strict stateful behaviors for TCP. To do so, stateful tracking must be enabled through either a state rule or global rule.

Use the `delete` form of this command to disable TCP strict tracking of stateful firewall rules.

Use the `show` form of this command to display the configuration of TCP strict tracking of stateful firewall rules.

show log firewall

Displays the firewall log.

Syntax:
`show log firewall` **`name`** *firewall-name* [**`rule`** *rule-number*]

Logs are displayed for all rules for the specified firewall.

firewall-name
Specifies the firewall by name.

rule-number
Restricts the output to a firewall rule.

Operational mode

Use this command to display the log for a specified firewall. Include a firewall rule to restrict the output to that rule.

For this command to work, the syslog level must be set to (**notice**, **info**, **or debug)** by using the `set system syslog global facility dataplane level` command.

The following example shows how to display the log for firewall fw1.

```
vyatta@vyatta:~$ show log firewall name fw1
2016-05-23T14:17:19.332976+00:00 localhost dataplane[16115]: fw rule fw1:10000 block tcp(6)
 src=dp0s10/2a:db:9c:f4:a2:a0/10.0.1.1(1000) dst=/52:54:0:13:af:c9/10.0.2.1(80) len=40 ttl=64
 window=512 res=0x00 SYN urgp=0
2016-05-23T14:17:19.432974+00:00 localhost dataplane[16115]: fw rule fw1:10000 block tcp(6)
 src=dp0s10/2a:db:9c:f4:a2:a0/10.0.1.1(1001) dst=/52:54:0:13:af:c9/10.0.2.1(80) len=40 ttl=64
 window=512 res=0x00 SYN urgp=0
2016-05-23T14:17:19.533278+00:00 localhost dataplane[16115]: fw rule fw1:10000 block tcp(6)
 src=dp0s10/2a:db:9c:f4:a2:a0/10.0.1.1(1002) dst=/52:54:0:13:af:c9/10.0.2.1(80) len=40 ttl=64
 window=512 res=0x00 SYN urgp=0
2016-05-23T14:17:19.633260+00:00 localhost dataplane[16115]: fw rule fw1:10000 block tcp(6)
 src=dp0s10/2a:db:9c:f4:a2:a0/10.0.1.1(1003) dst=/52:54:0:13:af:c9/10.0.2.1(80) len=40 ttl=64
 window=512 res=0x00 SYN urgp=0
2016-05-23T14:17:19.733200+00:00 localhost dataplane[16115]: fw rule fw1:10000 block tcp(6)
 src=dp0s10/2a:db:9c:f4:a2:a0/10.0.1.1(1004) dst=/52:54:0:13:af:c9/10.0.2.1(80) len=40 ttl=64
 window=512 res=0x00 SYN urgp=0
...
^C
vyatta@vyatta:~$
```

show session limit group

Displays session limit group information.

Syntax:

`show session limit group` [*group-name*]

group-name

The name of a configured session limit group.

Operational mode

Use the `show session limit group` command to display information about the session limit group.

The `show session limit group` command displays the following information.

Output field	Description
Active on	Interface on which this ruleset applies. Applies to sessions created inbound or outbound.
rule	Rule number.
parameter	Session limit parameter.
pronto	Protocol in match criteria.
allowed	Number of sessions that matched this rule and were permitted by the session limit parameter.
blocked	Number of sessions that matched this rule and that were blocked by the session limit parameter.

The following example shows how to display session limit group information.

```
prompt# show session limit group
Session limit group "GROUP1":
    Active on (dp0p1s1)
    rule    parameter  proto           allowed         blocked
    ----    ---------  -----           -------         -------
    10      PARAM1     any             333             726
    condition - all
```

show session limit parameter

Displays detailed session limit parameter information.

Syntax:

`show session limit parameter` [**`global`** | *parameter-name*]

`global`
Specifies the global session limit parameters.

parameter-name
The name of a configured system session limit parameter.

Operational mode

Use the `show session limit parameter` command to display detailed information about the global session limit parameters or a specific system session limit parameter.

The `show session limit parameter` command displays the following information.

Output field	Description
Sessions allowed	Total number of sessions allowed.
Sessions blocked	Total number of sessions blocked (sum of rate-limit sessions blocked and max-halfopen sessions blocked)
Current session counts	Number of current established, half-open, and terminating sessions that matched rules assigned to this session.
Max session counts	Maximum values of session counts since the last counter reset.
Time since last session created	Time since the last session was created.
Sessions per second average	Session per-second rates averaged over the last one second, one minute, and five minutes.
Max sessions per second	Highest value of sessions per-second rates since the last counter reset.
Time since max sessions per sec	Time since the **Max sessions per second** value occurred.
Time since last session blocked	Time since the last session was blocked.
Max sessions blocked per sec avg	Highest values of **Sessions per second average**.
Features	Features enabled on the session policy (rate-limit, max-halfopen).
Rate sessions/second	Rate-limit feature maximum sessions per second since configuration time.
Max burst	Rate-limit feature maximum burst of sessions created since configuration time.

Output field	Description
Interval (milliseconds)	Rate-limit interval derived from the configured rate and burst values.
Sessions blocked	Sessions blocked due to the rate-limit rate being exceeded.
Max halfopen maximum	Max-halfopen sessions limit. After the number of halfopen sessions reaches this value, no further sessions are created.
Sessions blocked	Sessions blocked due to the max-halfopen value being exceeded.

The following example shows how to display detailed session limit parameter information.

```
vyatta@R1# show session limit parameter PARAM1
Session limit parameter "PARAM1":
    Sessions allowed                                                    111
    Sessions blocked                                                    189
    Current session counts (estab/half-open/terminating)              [0:0:0]
    Max session counts (estab/half-open/terminating)                 [0:74:0]
    Time since last session created                                     1.9m
    Sessions per sec avg (1sec/1min/5mins)                            [0:0:0]
    Max sessions per sec avg (1sec/1min/5mins)                        [4:0:0]
    Time since max sessions per sec (1sec/1min/5mins)      [1.9m:never:never]
    Time since last session blocked                                     1.9m
    Max sessions blocked per sec avg (1sec/1min/5mins)                [7:0:0]
    Features                                                       rate-limit
    Rate limit
        Rate sessions/second                                              4
        Max burst                                                         4
        Interval (milliseconds)                                        1000
        Sessions blocked                                                189

Session limit group "GROUP1":
    Active on (dp0p1s1)
    rule    parameter  proto           allowed         blocked
    ----    ---------  -----           -------         -------
    10      PARAM1     udp             37              63
    condition - proto udp

    20      PARAM1     tcp             37              63
    condition - proto tcp

    30      PARAM1     icmp            37              63
    condition - proto icmp
```

show session limit parameter brief

Displays brief session limit parameter information.

Syntax:

```
show session limit parameter brief name parameter-name
```

parameter-name

The name of a configured system session limit parameter.

Operational mode

Use this command to display summary information for each currently configured system session limit parameter.

The `show session limit parameter brief` command displays the following information for each currently configured system session limit parameter.

Output field	Description
Name	Name of the system session limit parameter.
Sessions	Number of currently established, half-open, and terminating sessions.
Max	Maximum number of sessions ever established, half-open, and terminating.
Rate	Current creation rate of one-minute sessions.
HO Blocks	Sessions blocked as a result of the maximum half-open limit being reached.
RL Blocks	Sessions blocked as a result of the rate-limit rate being reached.

The following example shows how to display brief session limit parameter information.

```
prompt# show session limit parameter brief
Name             Sessions         Max              Rate (1min)  HO Blocks    RL Blocks
PARAM1           [0:7:0]          [0:111:0]        4            -            726
PARAM2           [0:2:0]          [0:211:0]        -            1432         -
PARAM3           [0:39:0]         [0:561:0]        20           0            0
```

system session limit global max-halfopen

Configures the limit for the global maximum number of half-open sessions.

Syntax:

`set system session limit global` **`max-halfopen`** *number*

Syntax:

`delete system session limit global` **`max-halfopen`** *number*

number

A number from 1 through 100000000.

Configuration mode.

```
system {
        session {
                limit {
                        global {
                                max-halfopen <1..100000000>
                                }
                        }
                }
        }
```

Use this command to stop the creation of sessions when the max-halfopen number of sessions created is exceeded.

Note: The global state limit configuration only applies to sessions created in interfaces that do not have any interface session limit configured.

Use the `set` form of this command to configure a max-halfopen limit.

Use the `delete` form of this command to remove a max-halfopen limit configuration.

system session limit global rate-limit

Configures the session limit global rate-limit rate and burst.

Syntax:

```
set system session limit global  rate-limit { rate rate-number | burst burst-number }
```

Syntax:

```
delete system session limit global  rate-limit { rate rate-number | burst burst-number }
```

rate-number

A number from 1 through 4294967295 (the maximum for at a uint32 data object).

burst-number

A number from 1 through 100000000.

Configuration mode.

```
system {
        session {
                limit {
                        global {
                                rate-limit {
                                    rate <1..4294967295>
                                    burst <0..100000000>
                                }
                        }
                }
        }
}
```

Use this command to stop the creation of sessions when a rate limit is exceeded.

Note: The global state limit configuration only applies to sessions created in interfaces that do not have any interface session limit configured.

Use the `set` form of this command to configure a rate limit.

Use the `delete` form of this command to remove a rate-limit configuration.

system session limit group name interface

Configures a session limit group name for an interface.

Syntax:

```
set system session limit group  name group-name  interface interface-name
```

Syntax:

```
delete system session limit group  name group-name  interface interface-name
```

group-name

A name for the session limit group.

interface-name

An interface name.

Configuration mode.

```
system {
```

```
        session {
                limit {
                       group {
                               name <group-name> {
                                   interface <interface-name>
                               }
                       }
                }
        }
}
```

Use this command to create a session limit group name and apply the session limit group name to an interface.

Use the `set` form of this command to configure a session limit group name.

Use the `delete` form of this command to remove a session limit group name configuration.

system session limit group name rule destination

Configures a rule set with a destination for a session limit group.

Syntax:

`set system session limit group` **`name`** *group-name* **`rule`** *rule-number* **`destination`** { **`address`** *value* | **`port`** *value* }

Syntax:

`delete system session limit group` **`name`** *group-name* **`rule`** *rule-number* **`destination`** { **`address`** *value* | **`port`** *value* }

group-name

A name for the session limit group.

rule-number

A number for the rule.

`address` ***value***

Specifies an IP address.

`port` ***value***

Specifies a port number.

Configuration mode.

```
system {
      session {
              limit {
                       group {
                               name <group-name> {
                                  rule <rule-number> {
                                          destination {
                                                address <value>
                                                port <value>
                                          }
                                  }
                               }
                       }
              }
        }
}
```

Use the `set` form of this command to configure a rule set with a destination for a session limit group.

Use the `delete` form of this command to remove a rule set with a destination from a session limit group configuration.

system session limit group name rule icmp

Configures a rule set with ICMP for a session limit group.

Syntax:

```
set system session limit group  name group-name  rule rule-number  icmp { group group-value | name
name | type type-value  code value }
```

Syntax:

```
delete system session limit group  name group-name  rule rule-number  icmp { group group-value | name
name | type type-number  code value }
```

group-name

A name for the session limit group.

rule-number

A number for the rule.

group-value

A value for the ICMP group.

name

The ICMP name can be:

- TOS-host-redirect
- TOS-host-unreachable
- TOS-network-redirect
- TOS-network-unreachable
- address-mask-reply
- address-mask-request
- communication-prohibited
- destination-unreachable
- echo-reply
- echo-request
- fragmentation-needed
- host-precedence-violation
- host-prohibited
- host-redirect
- host-unknown
- host-unreachable
- ip-header-bad
- network-prohibited
- network-redirect
- network-unknown
- network-unreachable
- parameter-problem
- port-unreachable
- precedence-cutoff
- protocol-unreachable
- redirect
- required-option-missing
- router-advertisement
- router-solicitation
- source-quench

- source-route-failed
- time-exceeded
- timestamp-reply
- timestamp-request
- ttl-zero-during-reassembly
- ttl-zero-during-transit

.

type-number

A number for the ICMP type.

value

A value for the code.

Configuration mode.

```
system {
      session {
              limit {
                       group {
                               name <group-name> {
                                       rule <rule-number> {
                                               icmp {
                                                    group <group-value>
                                                    name <name>
                                                    type <type-number> code <value>
                                               }
                                       }
                               }
                       }
                }
        }
}
```

Use the `set` form of this command to configure a rule set with ICMP for a session limit group.

Use the `delete` form of this command to remove a rule set with ICMP from a session limit group configuration.

system session limit group name rule icmpv6

Configures a rule set with ICMPv6 for a session limit group.

Syntax:

`set system session limit group` **`name`** *group-name* **`rule`** *rule-number* **`icmpv6`** { **`group`** *group-value* | **`name`** *name* | **`type`** *type-value* **`code`** *value* }

Syntax:

`delete system session limit group` **`name`** *group-name* **`rule`** *rule-number* **`icmpv6`** { **`group`** *group-value* | **`name`** *name* | **`type`** *type-number* **`code`** *value* }

group-name

A name for the session limit group.

rule-number

A number for the rule.

group-value

A value for the ICMPv6 group.

name

The ICMPv6 name can be:

- address-unreachable
- bad-header
- communication-prohibited
- destination-unreachable
- echo-reply
- echo-request
- mobile-prefix-advertisement
- mobile-prefix-solicitation
- multicast-listener-done
- multicast-listener-query
- multicast-listener-report
- neighbor-advertisement
- neighbor-solicitation
- no-route
- packet-too-big
- parameter-problem
- port-unreachable
- redirect
- router-advertisement
- router-solicitation
- time-exceeded
- ttl-zero-during-reassembly
- ttl-zero-during-transit
- unknown-header-type
- unknown-option

type-number

A number for the ICMPv6 type.

value

A value for the code.

Configuration mode.

```
system {
      session {
              limit {
                       group {
                               name <group-name> {
                                       rule <rule-number> {
                                               icmpv6 {
                                                    group <group-value>
                                                    name <name>
                                                    type <type-number> code <value>
                                               }
                                       }
                               }
                       }
               }
        }
}
```

Use the `set` form of this command to configure a rule set with ICMPv6 for a session limit group.

Use the `delete` form of this command to remove a rule set with ICMPv6 from a session limit group configuration.

system session limit group name rule parameter

Configures a rule set with a parameter for a session limit group.

Syntax:
`set system session limit group` **`name`** *group-name* **`rule`** *rule-number* **`parameter`** *system-session-limit-parameter-name*

Syntax:
`delete system session limit group` **`name`** *group-name* **`rule`** *rule-number* **`parameter`** *system-session-limit-parameter-name*

group-name
A name for the session limit group.

rule-number
A number for the rule.

system-session-limit-parameter-name
A name for the system session limit parameter.

Configuration mode.

```
system {
      session {
              limit {
                       group {
                               name <group-name> {
                                       rule <rule-number> {
                                               parameter <system-limit-parameter-name>
                                               }
                               }
                       }
               }
        }
}
```

This command must be entered after the `set system session limit parameter` command is entered because the `set session limit group name rule parameter` command uses the name of the system session limit parameter configured with the `set system session limit parameter` command.

Use the `set` form of this command to configure a rule set with a parameter for a session limit group.

Use the `delete` form of this command to remove a rule set with a parameter from a session limit group configuration.

system session limit group name rule protocol

Configures a rule set with a protocol for a session limit group.

Syntax:
`set system session limit group` **`name`** *group-name* **`rule`** *rule-number* **`protocol`** *protocol-name*

Syntax:
`delete system session limit group` **`name`** *group-name* **`rule`** *rule-number* **`protocol`** *protocol-name*

group-name
A name for the session limit group.

rule-number
A number for the rule.

protocol-name

A name of a protocol.

Configuration mode.

```
system {
      session {
              limit {
                       group {
                               name <group-name> {
                                  rule <rule-number> {
                                          protocol <protocol-name>
                                          }
                               }
                       }
                }
        }
}
```

Use the `set` form of this command to configure a rule set with a protocol for a session limit group.

Use the `delete` form of this command to remove a rule set with a protocol from a session limit group configuration.

system session limit group name rule source

Configures a rule set with a source for a session limit group.

Syntax:

`set system session limit group` **`name`** *group-name* **`rule`** *rule-number* **`source`** { **`address`** *value* | **`port`** *value* }

Syntax:

`delete system session limit group` **`name`** *group-name* **`rule`** *rule-number* **`source`** { **`address`** *value* | **`port`** *value* }

group-name

A name for the session limit group.

rule-number

A number for the rule.

`address` ***value***

Specifies an IP address.

`port` ***value***

Specifies a port number.

Configuration mode.

```
system {
      session {
              limit {
                       group {
                               name <group-name> {
                                  rule <rule-number> {
                                          source {
                                                address <value>
                                                port <value>
                                          }
                                  }
                               }
                       }
```

```
                }
        }
}
```

Use the `set` form of this command to configure a rule set with a source for a session limit group.

Use the `delete` form of this command to remove a rule set with a source from a session limit group configuration.

system session limit group name rule tcp flags

Configures a rule set with TCP flags for a session limit group.

Syntax:
`set system session limit group` **`name`** *group-name* **`rule`** *rule-number* **`tcp flags`** *value*

Syntax:
`delete system session limit group` **`name`** *group-name* **`rule`** *rule-number* **`tcp flags`** *value*

group-name
A name for the session limit group.

rule-number
A number for the rule.

value
TCP flag value.

Configuration mode.

```
system {
      session {
              limit {
                       group {
                               name <group-name> {
                                       rule <rule-number> {
                                               tcp flags <value>
                                               }
                               }
                       }
               }
        }
}
```

Use the `set` form of this command to configure a rule set with a TCP flag for a session limit group.

Use the `delete` form of this command to remove a rule set with a TCP flag from a session limit group configuration.

system session limit parameter name max-halfopen

Configures a number of max-halfopen sessions for a system session limit parameter.

Syntax:
`set system session limit parameter` **`name`** *system-session-limit-parameter-name* **`max-halfopen`** *number*

Syntax:
`delete system session limit parameter` **`name`** *system-session-limit-parameter-name* **`max-halfopen`** *number*

system-session-limit-parameter-name
A name for the system session limit parameter.

number

A number of max-halfopen sessions from 1 through 100000000.

Configuration mode.

```
system {
      session {
              limit {
                       parameter {
                               name <system-session-limit-parameter-name> {
                                  max-halfopen <1..100000000> {
                                  }
                               }
                       }
               }
        }
}
```

Use the `set` form of this command to configure a number of max-halfopen sessions for a system session limit parameter.

Use the `delete` form of this command to remove a number of max-halfopen sessions for a system session limit parameter configuration.

system session limit parameter name rate-limit

Configures a rate limit for a system session limit parameter.

Syntax:

`set system session limit parameter` **`name`** *system-session-limit-parameter-name* **`rate-limit`** { **`rate`** *rate-number* | **`burst`** *number* }

Syntax:

`delete system session limit parameter` **`name`** *system-session-limit-parameter-name* **`rate-limit`** { **`rate`** *rate-number* | **`burst`** *number* }

system-session-limit-parameter-name

A name for the system session limit parameter.

rate-number

A number used to rate-limit the sessions from 1 through 4294967295.

number

A number of burst sessions from 0 through 100000000.

Configuration mode.

```
system {
      session {
              limit {
                       parameter {
                               name <system-session-limit-parameter-name> {
                                  rate-limit {
                                      rate <1 ..max}
                                      burst <1..100000000>
                                  }
                               }
                       }
               }
        }
}
```

Use the `set` form of this command to configure a session rate limit for a system session limit parameter.

Use the `delete` form of this command to remove a session rate limit for a system session limit parameter configuration.

Zone-Based Firewall Commands

clear zone-policy

Clears firewall zone statistics.

Syntax:
```
clear zone-policy
```

Statistics are cleared on all firewall zones.

Operational mode

Use this command to clear statistics for firewall rules that are applied to zones.

show zone-policy

Displays the security zone policy for a security zone or security zone policies for all security zones.

Syntax:
```
show zone-policy [ zone zone ]
```

Security zone policies for all security zones are displayed.

zone *zone*
The name of a security zone.

Operational mode

Use this command to display the security zone policy for a security zone or security policies for all security zones.

The following example shows how to display security zone policies for all security zones on the R1 router.

```
vyatta@R1:~$ show zone-policy
-------------------
Name: LAN1
Interfaces: dp0p256p1
To Zone:
  name                                  firewall
  ----                                  --------
  LAN2                                  fw_1
-------------------
Name: LAN2
Interfaces: dp0p192p1
To Zone:
  name                                  firewall
  ----                                  --------
  LAN1                                  fw_2
```

The following example shows how to display security zone policies for a specific security zone (inside) on the R1 router.

```
vyatta@R1:~$ show zone-policy zone inside
-------------------
Name: inside *description*
```

```
Interfaces: peth0 peth1 peth2 peth3

To Zone:
  name                                  firewall
  ----                                  --------
  outside                               local-to-inside local-to-inside-6
```

security zone-policy zone <zone>

Defines a security zone policy.

Syntax:
`set security zone-policy zone` *zone*

Syntax:
`delete security zone-policy zone [` *zone* `]`

Syntax:
`show security zone-policy`

zone
The name of a security zone.

You can define more than one security zone by creating more than one `zone-policy zone` configuration node.

Configuration mode

```
security {
    zone-policy {
        zone zone {
        }
    }
}
```

In the vRouter, a zone is defined as a group of interfaces that have the same security level. After a zone is defined, firewall rule sets can be applied to traffic flowing between zones.

By default, traffic to a zone is dropped unless a policy has been defined for the zone sending the traffic. Traffic flowing within a zone is not filtered.

When defining a zone, keep the following in mind:

- An interface can be a member of only one zone.
- An interface that is a member of a zone cannot have a firewall rule set directly applied to it.
- For interfaces not assigned to a zone, traffic is unfiltered by default. These interfaces can have rule sets directly applied to them.

Use the `set` form of this command to define a security zone.

Use the `delete` form of this command to delete a security zone.

Use the `show` form of this command to display the configuration of a security zone. See show zone-policy *(page 91)*.

security zone-policy zone <zone> default-action <action>

Defines the default action for traffic leaving a security zone.

Syntax:

```
set security zone-policy zone zone default-action { accept | drop }
```

Syntax:

```
delete security zone-policy zone zone default-action [ accept | drop ]
```

Syntax:

```
show security zone-policy zone zone default-action
```

Traffic is dropped silently.

zone
The name of a security zone for which traffic is destined.

accept
Accepts traffic. The action to be taken for traffic leaving the zone and does not match any firewall rule sets.

drop
Drops traffic silently. The action to be taken for traffic leaving the zone and does not match any firewall rule sets.

Note: This is the default action if default-action is not set.

Configuration mode

```
security {
    zone-policy {
        zone zone {
            default-action
                accept
                drop
        }
    }
}
```

This action is taken for all traffic leaving a zone where the traffic does not match any firewall rules.

Use the `set` form of this command to set the default action for traffic leaving a security zone.

Use the `delete` form of this command to restore the default action, that is, traffic is dropped silently.

Use the `show` form of this command to display the configuration of the default action.

security zone-policy zone <zone> description <description>

Provides a description for a security zone.

Syntax:

```
set security zone-policy zone zone description description
```

Syntax:

```
delete security zone-policy zone zone description
```

Syntax:

```
show security zone-policy zone zone description
```

zone
The name of a security zone for which traffic is destined.

description
A brief description for the security zone. If the description contains spaces, it must be enclosed in double quotation marks.

Configuration mode

```
security {
    zone-policy {
        zone zone {
            description description
        }
    }
}
```

Use the `set` form of this command to provide a description.

Use the `delete` form of this command to delete a description.

Use the `show` form of this command to display the description.

security zone-policy zone <from-zone> to <to-zone>

Specifies the destination zone for a given source zone.

Syntax:

`set security zone-policy zone` *from-zone* **`to`** *to-zone*

Syntax:

`delete security zone-policy zone` *from-zone* **`to`** *to-zone*

Syntax:

`show security zone-policy`

from-zone

The name of a security zone from which traffic is originating.

to-zone

The name of a security zone for which traffic is destined.

Configuration mode

```
security {
    zone-policy {
        zone from-zone {
            to to-zone
        }
    }
}
```

Use this command to specify a destination zone for a given source zone.

Use the `set` form of this command to specify the source and destination zones.

Use the `delete` form of this command to delete the source and destination zones.

Use the `show` form of this command to display the configuration of a source zone.

security zone-policy zone <from-zone> to <to-zone> firewall <name>

Applies a firewall rule set to the packet flow between two zones.

Syntax:

`set security zone-policy zone` *from-zone* **`to`** *to-zone* **`firewall name`** *name*

Syntax:

`delete security zone-policy zone` *from-zone* **`to`** *to-zone* **`firewall name`**

Syntax:

```
show security zone-policy zone from-zone to to-zone firewall name
```

from-zone
The name of a security zone from which traffic is originating.

to-zone
The name of a security zone for which traffic is destined.

name
The name of a firewall rule set.

Configuration mode

```
security {
    zone-policy {
        zone from-zone {
            to to-zone {
                firewall name
            }
        }
    }
}
```

You can apply multiple rulesets by running this command multiple times and specifying differing rule set names.

Use the `set` form of this command to define a rule set that filters packets flowing from one zone to another.

Use the `delete` form of this command to delete a packet-filtering rule set.

Use the `show` form of this command to display the configured rule sets.

security zone-policy zone <zone> interface <interface-name>

Adds an interface to a security zone.

Syntax:

```
set security zone-policy zone zone interface interface-name
```

Syntax:

```
delete security zone-policy zone zone interface interface-name
```

Syntax:

```
show security zone-policy zone zone interface interface-name
```

zone
The name of a security zone for which traffic is destined.

interface-name
The name of an interface; for example, dp0p1p1, wan1, or ppp1. You can add multiple interfaces by running this command multiple times and specifying differing interface names.

Configuration mode

```
security {
    zone-policy {
        zone zone {
            interface interface-name
        }
    }
}
```

All interfaces in the zone have the same security level; traffic arriving to those interfaces from other zones is all treated in the same way. Traffic flowing between interfaces in the same security zone is not filtered.

Use the `set` form of this command to add an interface to a zone.

Use the `delete` form of this command to delete an interface from a zone.

Use the `show` form of this command to display which interfaces are members of a zone.

ICMP Types

This appendix lists the Internet Control Messaging Protocol (ICMP) types defined by the Internet Assigned Numbers Authority (IANA).

The IANA has developed a standard that maps a set of integers onto ICMP types. The following table lists the ICMP types and codes defined by the IANA and maps them to the literal strings that are available in the vRouter system.

Table 13: ICMP types

ICMP Type	Code	Literal	Description
0 - Echo reply	0	echo-reply	Echo reply (pong)
3 - Destination unreachable		destination- unreachable	Destination is unreachable
	0	network-unreachable	Destination network is unreachable
	1	host-unreachable	Destination host is unreachable
	2	protocol-unreachable	Destination protocol is unreachable
	3	port-unreachable	Destination port is unreachable
	4	fragmentation-needed	Fragmentation is required
	5	source-route-failed	Source route has failed
	6	network-unknown	Destination network is unknown
	7	host-unknown	Destination host is unknown
	9	network-prohibited	Network is administratively prohibited
	10	host-prohibited	Host is administratively is prohibited
	11	ToS-network-unreachable	Network is unreachable for ToS
	12	ToS-host-unreachable	Host is unreachable for ToS

ICMP Type	Code	Literal	Description
	13	communication-prohibited	Communication is administratively prohibited
	14	host-precedence-violation	Requested precedence is not permitted.
	15	precedence-cutoff	Precedence is lower than the required minimum.
4 - Source quench	0	source-quench	Source is quenched (congestion control)
5 - Redirect message		redirect	Redirected message
	0	network-redirect	Datagram is redirected for the network
	1	host-redirect	Datagram is redirected for the host
	2	ToS-network-redirect	Datagram is redirected for the ToS and network
	3	ToS-host-redirect	Datagram is redirected for the ToS and host
8 - Echo request	0	echo-request	Echo request (ping)
9 - Router advertisement	0	router-advertisement	Router advertisement
10 - Router solicitation	0	router-solicitation	Router solicitation
11 - Time exceeded		time-exceeded	Time to live (TTL) has exceeded
	0	ttl-zero-during-transit	TTL has expired in transit
	1	ttl-zero-during-reassembly	Fragment reassembly time has exceeded
12 - Parameter problem: Bad IP header		parameter-problem	Bad IP header
	0	ip-header-bad	Pointer that indicates an error
	1	required-option-missing	Missing required option
13 - Timestamp	0	timestamp-request	Request for a timestamp
14 - Timestamp reply	0	timestamp-reply	Reply to a request for a timestamp

ICMP Type	Code	Literal	Description
15 - Information request	0		Information request
16 - Information reply	0		Information reply
17 - Address mask request	0	address-mask-request	Address mask request
18 - Address mask reply	0	address-mask-reply	Address mask reply

ICMPv6 Types

This appendix lists the ICMPv6 types defined by the Internet Assigned Numbers Authority (IANA).

The Internet Assigned Numbers Authority (IANA) has developed a standard that maps a set of integers onto ICMPv6 types. The following table lists the ICMPv6 types and codes defined by the IANA and maps them to the strings literal strings available in the AT&T Vyatta vRouter system.

Table 14: ICMPv6 types

ICMPv6 Type	Code	Literal	Description
1 - Destination unreachable		destination- unreachable	
	0	no-route	No route to destination
	1	communication-prohibited	Communication with destination administratively prohibited
	2		Beyond scope of source address
	3	address-unreachable	Address unreachable
	4	port-unreachable	Port unreachable
	5		Source address failed ingress/egress policy
	6		Reject route to destination
2 - Packet too big	0	packet-too-big	
3 - Time exceeded		time-exceeded	
	0	ttl-zero-during-transit	Hop limit exceeded in transit
	1	ttl-zero-during-reassembly	Fragment reassembly time exceeded
4 - Parameter problem		parameter-problem	
	0	bad-header	Erroneous header field encountered
	1	unknown-header-type	Unrecognized Next Header type encountered

ICMPv6 Type	Code	Literal	Description
	2	unknown-option	Unrecognized IPv6 option encountered
128 - Echo request	0	echo-request (ping)	Echo request
129 - Echo reply	0	echo-reply (pong)	Echo reply
133 - Router solicitation	0	router-solicitation	Router solicitation
134 - Router advertisement	0	router-advertisement	Router advertisement
135 - Neighbor solicitation	0	neighbor-solicitation (neighbour-solicitation)	Neighbor solicitation
136 - Neighbor advertisement	0	neighbor-advertisement (neighbour-advertisement)	Neighbor advertisement

Supported Interface Types

The following table shows the syntax and parameters of supported interface types. Depending on the command, some of these types may not apply.

Interface Type	Syntax	Parameters
Bridge	`bridge` *brx*	*brx*: The name of a bridge group. The name ranges from br0 through br999.

Interface Type	Syntax	Parameters
Data plane	`dataplane` *interface-name*	*interface-name*: The name of a data plane interface. Following are the supported formats of the interface name: • `dp`*x*`p`*y*`p`*z*—The name of a data plane interface, where — `dp`*x* specifies the data plane identifier (ID). Currently, only dp0 is supported. — `p`*y* specifies a physical or virtual PCI slot index (for example, p129). — `p`*z* specifies a port index (for example, p1). For example, dp0p1p2, dp0p160p1, and dp0p192p1. • `dp`*x*`em`*y* —The name of a data plane interface on a LAN-on-motherboard (LOM) device that does not have a PCI slot, where `em`*y* specifies an embedded network interface number (typically, a small number). For example, dp0em3. • `dp`*x*`s`*y*—The name of a data plane interface in a system in which the BIOS identifies the network interface card to reside in a particular physical or virtual slot *y*, where *y* is typically a small number. For example, for the dp0s2 interface, the BIOS identifies slot 2 in the system to contain this interface. • `dp`*x*`P`*n*`p`*y*`p`*z* —The name of a data plane interface on a device that is installed on a secondary PCI bus, where `P`*n* specifies the bus number. You can use this format to name data plane interfaces on large physical devices with multiple PCI buses. For these devices, it is possible to have network interface cards installed on different buses with these cards having the same slot ID. The value of *n* must be an integer greater than 0. For example, dp0P1p162p1 and dp0P2p162p1.

Interface Type	Syntax	Parameters
Data plane vif	`dataplane` *interface-name* `vif` *vif-id* [`vlan` *vlan-id*]	*interface-name*: Refer to the preceding description. *vif-id*: A virtual interface ID. The ID ranges from 1 through 4094. *vlan-id*: The VLAN ID of a virtual interface. The ID ranges from 1 through 4094.
Loopback	`loopback lo` or `loopback lo`*n*	*n*: The name of a loopback interface, where *n* ranges from 1 through 99999.
OpenVPN	`openvpn` *vtunx*	*vtunx*: The identifier of an OpenVPN interface. The identifier ranges from vtun0 through vtun*x*, where *x* is a nonnegative integer.
Tunnel	`tunnel` *tunx* or `tunnel` *tunx* `parameters`	*tunx*: The identifier of a tunnel interface you are defining. The identifier ranges from tun0 through tun*x*, where *x* is a nonnegative integer.
Virtual tunnel	`vti` *vtix*	*vtix*: The identifier of a virtual tunnel interface you are defining. The identifier ranges from vti0 through vti*x*, where *x* is a nonnegative integer. **Note:** Before you can configure a vti interface, you must configure a corresponding vpn. **Note:** This interface does not support IPv6.
VRRP	*parent-interface* `vrrp vrrp-group` *group*	*parent-interface*: The type and identifier of a parent interface; for example, data plane dp0p1p2 or bridge br999. *group*: A VRRP group identifier. The name of a VRRP interface is not specified. The system internally constructs the interface name from the parent interface identifier plus the VRRP group number; for example, dp0p1p2v99. Note that VRRP interfaces support the same feature set as does the parent interface.

List of Acronyms

Acronym	Description
ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
AH	Authentication Header
AMI	Amazon Machine Image
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
AWS	Amazon Web Services
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CCMP	AES in counter mode with CBC-MAC
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6
DLCI	data-link connection identifier
DMI	desktop management interface
DMVPN	dynamic multipoint VPN
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
EBS	Amazon Elastic Block Storage
EC2	Amazon Elastic Compute Cloud
EGP	Exterior Gateway Protocol
ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Output
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers

Acronym	Description
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocol
IPS	Intrusion Protection System
IKE	Internet Key Exchange
IP	Internet Protocol
IPOA	IP over ATM
IPsec	IP Security
IPv4	IP Version 4
IPv6	IP Version 6
ISAKMP	Internet Security Association and Key Management Protocol
ISM	Internet Standard Multicast
ISP	Internet Service Provider
KVM	Kernel-Based Virtual Machine
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol
LLDP	Link Layer Discovery Protocol
MAC	medium access control
mGRE	multipoint GRE
MIB	Management Information Base
MLD	Multicast Listener Discovery
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
NBMA	Non-Broadcast Multi-Access
ND	Neighbor Discovery
NHRP	Next Hop Resolution Protocol
NIC	network interface card
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OSPFv2	OSPF Version 2
OSPFv3	OSPF Version 3
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PAT	Port Address Translation
PCI	peripheral component interconnect
PIM	Protocol Independent Multicast
PIM-DM	PIM Dense Mode
PIM-SM	PIM Sparse Mode
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM

Acronym	Description
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PTMU	Path Maximum Transfer Unit
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RHEL	Red Hat Enterprise Linux
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
RP	Rendezvous Point
RPF	Reverse Path Forwarding
RSA	Rivest, Shamir, and Adleman
Rx	receive
S3	Amazon Simple Storage Service
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SPT	Shortest Path Tree
SSH	Secure Shell
SSID	Service Set Identifier
SSM	Source-Specific Multicast
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TBF	Token Bucket Filter
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service
TSS	TCP Maximum Segment Size
Tx	transmit
UDP	User Datagram Protocol
VHD	virtual hard disk
vif	virtual interface
VLAN	virtual LAN
VPC	Amazon virtual private cloud
VPN	virtual private network
VRRP	Virtual Router Redundancy Protocol
WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access