VYATTA, INC. | Vyatta System

Security

REFERENCE GUIDE

Firewall
Intrusion Protection System
Traffic Filtering
URL Filtering

Vyatta
Suite 200
1301 Shoreway Road
Belmont, CA 94002
vyatta.com
650 413 7200
1 888 VYATTA 1 (US and Canada)

COPYRIGHT

Vyatta reserves the right to make changes to software, hardware, and documentation without notice. For the most recent version of documentation, visit the Vyatta web site at vyatta.com.

PROPRIETARY NOTICES

ISSUE DATE: February 2009

DOCUMENT REVISION. VC5 v03

RELEASED WITH: VC5.0.2

PART NO. A0-0119-10-0002

Table of Contents

Quick Reference to Commands

Use this section to help you quickly locate a command.

Quick List of Examples

Use this list to help you locate examples you'd like to try or look at.

Preface

This guide explains how to deploy security features of the Vyatta system. It describes the available commands and provides configuration examples.

This preface provides information about using this guide. The following topics are covered:

- Intended Audience
- Organization of This Guide
- Document Conventions
- Vyatta Publications

Intended Audience

This guide is intended for experienced system and network administrators. Depending on the functionality to be used, readers should have specific knowledge in the following areas:

- Networking and data communications
- TCP/IP protocols
- General router configuration
- Routing protocols
- Network administration
- Network security

Organization of This Guide

This guide has the following aid to help you find the information you are looking for:

- **Quick Reference to Commands**

  Use this section to help you quickly locate a command.

- **Quick Reference to Commands**

  Use this section to help you quickly locate a command.

This guide has the following chapters and appendixes:

Chapter	Description	Page
Chapter 1: Firewall	This chapter explains how to use the firewall feature of the Vyatta system.	1
Chapter 2: Intrusion Protection System	This chapter lists the commands for setting up intrustion detection and prevention, and traffic filtering on the Vyatta system.	88
Chapter 3: Traffic Filtering	This chapter lists the commands for setting up traffic filtering on the Vyatta system.	107
Chapter 4: URL Filtering	This chapter explains how to set up URL filtering on the Vyatta system.	111
Chapter A: ICMP Types	This appendix lists the ICMP types defined by the Internet Assigned Numbers Authority (IANA).	140
Glossary of Acronyms		143

Document Conventions

This guide contains advisory paragraphs and uses typographic conventions.

Advisory Paragraphs

This guide uses the following advisory paragraphs:

Warnings alert you to situations that may pose a threat to personal safety, as in the following example:

WARNING *Risk of injury. Switch off power at the main breaker before attempting to connect the remote cable to the service power at the utility box.*

Cautions alert you to situations that might cause harm to your system or damage to equipment, or that may affect service, as in the following example:

CAUTION *Risk of loss of service. Restarting a running system will interrupt service.*

Notes provide information you might need to avoid problems or configuration errors:

NOTE *You must create and configure network interfaces before enabling them for routing protocols.*

Typographic Conventions

This document uses the following typographic conventions:

`Courier`	Examples, command-line output, and representations of configuration nodes.
`boldface Courier`	In an example, your input: something you type at a command line.
boldface	In-line commands, keywords, and file names .
italics	Arguments and variables, where you supply a value.
<key>	A key on your keyboard. Combinations of keys are joined by plus signs (“+”). An example is <Ctrl>+<Alt>+<Del>.
[*arg1* | *arg2*]	Enumerated options for completing a syntax. An example is [enable | disable].

num1–numN	A inclusive range of numbers. An example is 1–65535, which means 1 through 65535.
arg1..*argN*	A range of enumerated values. An example is eth0..eth3, which means eth0, eth1, eth2, and eth3.
arg [*arg* ...] *arg*,[*arg*,...]	A value that can optionally represent a list of elements (a space-separated list in the first case, and a comma-separated list in the second case).

Vyatta Publications

More information about the Vyatta system is available in the Vyatta technical library, and on www.vyatta.com and www.vyatta.org.

Full product documentation is provided in the Vyatta technical library. To see what documentation is available for your release, see the *Guide to Vyatta Documentation*. This guide is posted with every release of Vyatta software and provides a great starting point for finding what you need.

Chapter 1: Firewall

This chapter explains how to use the firewall feature of the Vyatta system.

This chapter presents the following topics:

- Firewall Configuration
- Firewall Commands

Firewall Configuration

This section describes how to configure firewall protection on the Vyatta system.

This section presents the following topics:

- Firewall Overview
- Firewall Configuration Examples
- Viewing Firewall Information

Firewall Overview

The Vyatta system's firewall functionality analyzes and filters IP packets between network interfaces. The most common application of this 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. It provides:

- Packet filtering can be performed for traffic traversing the router, using "in" and "out" on an interface. Packets destined to the router itself can be filtered using the "local" keyword.
- Criteria that can be defined for packet-matching rules include source IP address, destination IP address, source port, destination port, IP protocol, and ICMP type.
- General detection on IP options such as source routing and broadcast packets

The Vyatta firewall features stateful packet inspection and can provide significant additional protection in a layered security strategy. The system can intercept network activity, categorize it against its configured database of permitted traffic, and allow or deny the attempt. This adds add an extra layer of security when used in conjunction with stateful packet-filtering devices.

To use the firewall feature, you define a firewall rule set as a named firewall instance. You then apply the firewall instance to interfaces, where the instance acts as a packet filter. The firewall instance will filter packets in one of the following ways, depending on what you specify when you apply the firewall instance:

- **in.** If you apply the rule set as **in**, the firewall will filter packets entering the interface.
- **out.** If you apply the rule set as **out**, the firewall will filter packets leaving the interface.
- **local.** If you apply the rule set as **local**, the firewall will filter packets destined for the Vyatta system.

For each interface, you can apply up to three firewall instances: one firewall **in** instance, one firewall **out** instance, and one firewall **local** instance.

Note that after the final user-defined rule in a rule set is executed, an implicit rule of **reject all** takes effect.

Make sure the firewall instance you apply to an interface is already defined, or you may experience unintended results. If you apply a firewall instance that does not exist to an interface, the implicit firewall rule of **allow all** will be applied.

Firewall Configuration Examples

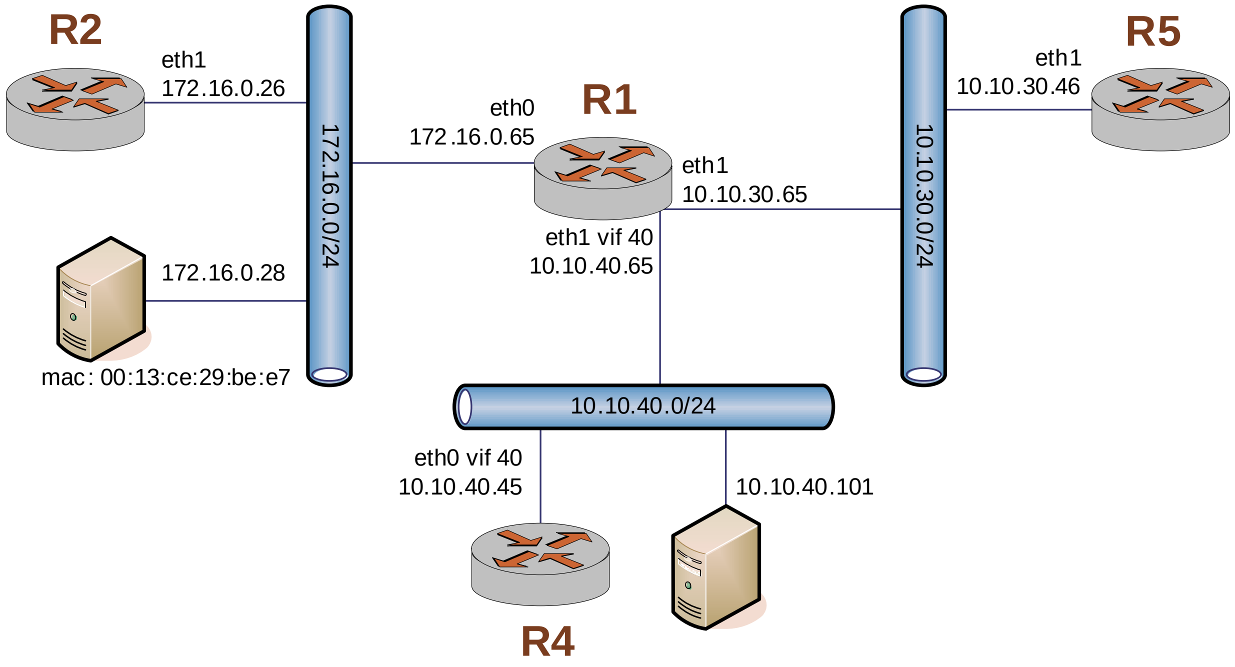
This section sets up a basic firewall configuration. To configure the firewall:

1. You define a number of named firewall rule sets. This includes:
   - Specifying match conditions for traffic.
   - Specifying the action to be taken if traffic matches the specified criteria. Traffic can be **accepted**, silently **dropped**, or **rejected** with a TCP reset.
2. You apply the named rule sets to an interface as packet filters. You can apply one named rule set to each of the following:
   - **in.** If you apply the rule set as **in**, the firewall will filter packets entering the interface.
   - **out.** If you apply the rule set as **out**, the firewall will filter packets leaving the interface.
   - **local.** If you apply the rule set as **local**, the firewall will filter packets destined for the Vyatta system.

Note that after the final user-defined rule in a rule set is executed, an implicit rule of **reject all** takes effect.

This section presents a sample configuration for firewall. When you have finished, the firewall will be configured on router R1 as shown in Figure 1-1.

Figure 1-1 Firewall



This section includes the following examples:

- Example 1-1 Filtering on source IP
- Example 1-2 Filtering on source and destination IP
- Example 1-3 Filtering on source IP and destination protocol
- Example 1-4 Defining a network-to-network filter
- Example 1-5 Filtering on source MAC address
- Example 1-6 Excluding an address

Filter on Source IP

Example 1-1 defines a firewall rule set containing one rule, which filters on source IP address only. This rule will deny packets coming from router R2. It then applies the firewall rule set to packets inbound on interface eth0.

To create a rule set that filters on source IP, perform the following steps in configuration mode:

Example 1-1 Filtering on source IP

Step	Command

Create the configuration node for FWTEST-1 and its rule Rule 1. This rule rejects traffic matching the specified criteria.	`vyatta@R1#` **`set firewall name FWTEST-1 rule 1 action reject`**
	`[edit]`

This rule applies to traffic that has 176.16.0.26 as the source.	`vyatta@R1#` **`set firewall name FWTEST-1 rule 1 source address 172.16.0.26`**
	`[edit]`

Apply FWTEST-1 to inbound packets on eth0.	`vyatta@R1#` **`set interfaces ethernet eth0 firewall in name FWTEST-1`**
	`[edit]`

Commit the configuration.	`vyatta@R1#` **`commit`**
	`[edit]`

Filter on Source and Destination IP

Example 1-2 defines another firewall rule set. It contains one rule, which filters on both source and destination IP address. This rule accepts packets leaving R5 through eth1 using 10.10.30.46, and destined for 10.10.40.101. It then applies the firewall rule set to packets outbound from vif 1 on interface eth1.

To create a rule set that filters on source and destination IP, perform the following steps in configuration mode:

Example 1-2 Filtering on source and destination IP

Step	Command

Create the configuration node for FWTEST-2 and its rule Rule 1. This rule accepts traffic matching the specified criteria.	`vyatta@R1#` **`set firewall name FWTEST-2 rule 1 action accept`**
	`[edit]`

Example 1-2 Filtering on source and destination IP

This rule applies to traffic that has 10.10.30.46 as the source.	vyatta@R1# **set firewall name FWTEST-2 rule 1 source address 10.10.30.46** [edit]
This rule applies to traffic that has 10.10.40.101 as the destination.	vyatta@R1# **set firewall name FWTEST-2 rule 1 destination address 10.10.40.101** [edit]
Apply FWTEST-2 to outbound packets on eth1 vif 40.	vyatta@R1# **set interfaces ethernet eth1 vif 40 firewall out name FWTEST-2** [edit]
Commit the configuration.	vyatta@R1# **commit** [edit]

Filter on Source IP and Destination Protocol

Example 1-3 defines 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 rule set is applied to local packets (that is, packets destined for this router, R1) through eth1.

To create a rule set that filters on source IP and destination protocol, perform the following steps in configuration mode:

Example 1-3 Filtering on source IP and destination protocol

Step	Command
Create the configuration node for FWTEST-3 and its rule Rule 1. This rule accepts traffic matching the specified criteria.	vyatta@R1# **set firewall name FWTEST-3 rule 1 action accept** [edit]
This rule applies to traffic that has 10.10.30.46 as the source.	vyatta@R1# **set firewall name FWTEST-3 rule 1 source address 10.10.30.46** [edit]
This rule applies to TCP traffic.	vyatta@R1# **set firewall name FWTEST-3 rule 1 protocol tcp** [edit]
This rule applies to traffic that is destined for the Telnet service.	vyatta@R1# **set firewall name FWTEST-3 rule 1 destination port telnet** [edit]
Apply FWTEST-3 to packets bound for this router arriving on eth1.	vyatta@R1# **set interfaces ethernet eth1 firewall local name FWTEST-3** [edit]

Example 1-3 Filtering on source IP and destination protocol

Commit the configuration.	vyatta@R1# **commit** [edit]

Defining a Network-to-Network Filter

Example 1-4 creates 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 rule set to packets inbound through vif 40 on interface eth1.

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

Example 1-4 Defining a network-to-network filter

Step	Command
Create the configuration node for FWTEST-4 and its rule Rule 1. This rule accepts traffic matching the specified criteria.	vyatta@R1# **set firewall name FWTEST-4 rule 1 action accept** [edit]
This rule applies to traffic coming from the network 10.10.40.0/24.	vyatta@R1# **set firewall name FWTEST-4 rule 1 source address 10.10.40.0/24** [edit]
This rule applies to traffic destined for the network 172.16.0.0/24.	vyatta@R1# **set firewall name FWTEST-4 rule 1 destination address 172.16.0.0/24** [edit]
Apply FWTEST-4 to packets bound for this router arriving through vif 40 on eth1.	vyatta@R1# **set interfaces ethernet eth1 vif 40 firewall in name FWTEST-4** [edit]
Commit the configuration.	vyatta@R1# **commit** [edit]

Filter on Source MAC Address

Example 1-5 defines a firewall rule set containing one rule, which filters on source MAC address only. This rule will allow packets coming from a specific computer, identified by its MAC address rather than its IP address. The rule set is applied to packets inbound on interface eth0.

To create a rule set that filters on source MAC address, perform the following steps in configuration mode:

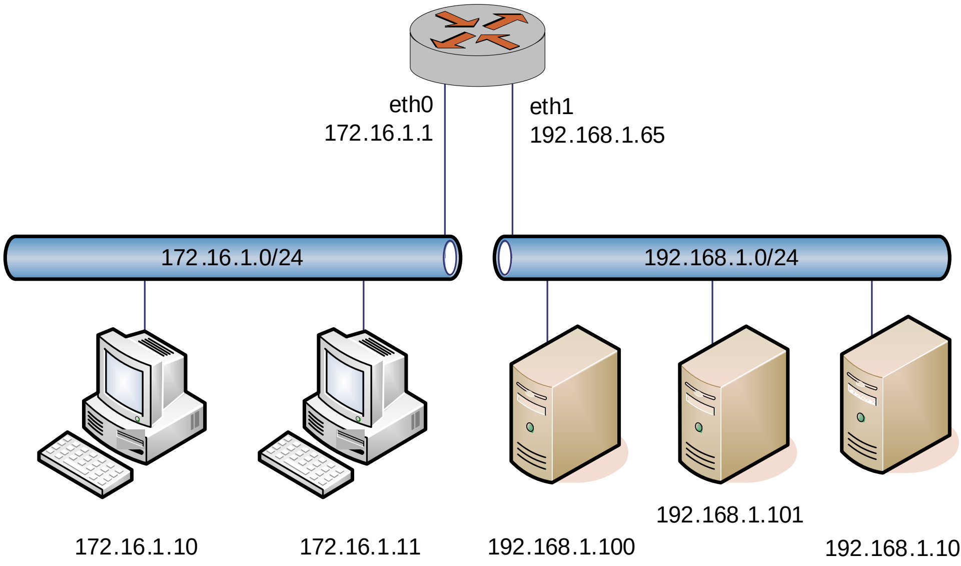
Example 1-5 Filtering on source MAC address

Step	Command
Create the configuration node for FWTEST-5 and its rule Rule 1. This rule accepts traffic matching the specified criteria.	vyatta@R1# **set firewall name FWTEST-5 rule 1 action accept** [edit]
This rule applies to traffic that has 00:13:ce:29:be:e7 as the source MAC address.	vyatta@R1# **set firewall name FWTEST-5 rule 1 source mac-address 00:13:ce:29:be:e7** [edit]
Apply FWTEST-5 to inbound packets on eth0.	vyatta@R1# **set interfaces ethernet eth0 firewall in name FWTEST-5** [edit]
Commit the configuration.	vyatta@R1# **commit** [edit]

Excluding an Address

The firewall rule shown in Example 1-6 allows all traffic from the 172.16.1.0/24 network except to server 192.168.1.100.

Figure 1-2 Excluding an address



To create a rule set that excludes an address, perform the following steps in configuration mode:

Example 1-6 Excluding an address

Step	Command

Create the configuration node for FWTEST-5 and its rule 10. Give a description for the rule.	`vyatta@R1# set firewall name NEGATED-EXAMPLE rule 10 description "Allow all traffic from LAN except to server 192.168.1.100"`
	`[edit]`

All traffic that matches the rule will be accepted.	`vyatta@R1# set firewall name NEGATED-EXAMPLE rule 10 action accept`
	`[edit]`

Any traffic from network 172.16.1.0/24 matches the rule.	`vyatta@R1# set firewall name NEGATED-EXAMPLE rule 10 source address 172.16.1.0/24`
	`[edit]`

Traffic destined anywhere EXCEPT 192.168.1.100 matches the rule. That traffic does not match the rule, and invokes the implicit “reject all” rule.	`vyatta@R1# set firewall name NEGATED-EXAMPLE rule 10 destination address !192.168.1.100`
	`[edit]`

Apply the rule set NEGATED-EXAMPLE to inbound packets on eth0.	`vyatta@R1# set interfaces ethernet eth0 firewall in name NEGATED-EXAMPLE`
	`[edit]`

Commit the configuration.	`vyatta@R1# commit`
	`[edit]`

Example 1-6 Excluding an address

Show the configuration.

```
vyatta@R1# show 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
         }
      }
}
[edit]

vyatta@R1# show interfaces ethernet eth0
      address 172.16.1.1/24
      firewall {
         in {
            name NEGATED-EXAMPLE
         }
      }
      hw-id 00:0c:29:99:d7:74
[edit]
```

Viewing Firewall Information

This section includes the following examples:

- Example 1-7 Showing firewall rule sets
- Example 1-8 Showing firewall configuration on an interface
- Example 1-9 Displaying the “firewall” configuration node

Showing Firewall Rule Set Information

You can see how firewall rule sets are set up by using the **show firewall** command in operational mode and specifying the name of the rule set. If no rule set is specified then all defined rule sets are displayed.

Example 1-7 shows the information you configured for firewall rule set FWTEST-1 and FWTEST-3.

Example 1-7 Showing firewall rule sets

```
vyatta@R1:~$ show firewall FWTEST-1

Active on (eth0, IN)

State Codes: E - Established, I - Invalid, N - New, R - Related

rule  action  source            destination        proto  state
----  ------  ------            -----------        -----  -----
1     REJECT  172.16.0.26       0.0.0.0/0           all    any
1025  DROP    0.0.0.0/0         0.0.0.0/0           all    any


vyatta@R1:~$ show firewall FWTEST-3

Active on (eth1, LOCAL)

State Codes: E - Established, I - Invalid, N - New, R - Related

rule  action  source            destination        proto  state
----  ------  ------            -----------        -----  -----
1     ACCEPT  10.10.30.46         0.0.0.0/0          tcp    any
                                  dst ports: telnet
1025  DROP    0.0.0.0/0           0.0.0.0/0          all    any


vyatta@R1:~$
```

Showing Firewall Configuration on Interfaces

Example 1-8 shows how firewall rule set FWTEST-1 is applied to interface eth0.

Example 1-8 Showing firewall configuration on an interface

```
vyatta@R1# show interfaces ethernet eth0 firewall
   in {
      name FWTEST-1
   }
[edit]
vyatta@R1#
```

Showing Firewall Configuration

You can always view the information in configuration nodes by using the **show** command in configuration mode. In this case you can view firewall configuration by using the **show firewall** command in configuration mode, as shown in Example 1-9.

Example 1-9 Displaying the “firewall” configuration node

```
vyatta@R1# show frewall
   name FWTEST-1 {
      rule 1 {
         action reject
         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
         }
      }
   }
   name FWTEST-5 {
      rule 1 {
         action accept
         source {
            mac-addr 00:13:ce:29:be:e7
         }
      }
   }
[edit]
vyatta@R1#
```

Firewall Commands

This chapter contains the following commands.

Configuration Commands	
Global Firewall Configuration	
firewall	Enables firewall on the system.
firewall broadcast-ping <state>	Specifies whether the system will respond to ICMP Echo request messages sent to an IP broadcast address.
firewall ip-src-route <state>	Specifies whether to permit or deny packets with the Loose Source Route or Strict Source Route IP options.
firewall log-martians <state>	Specifies whether to log packets with impossible addresses.
firewall receive-redirects <state>	Specifies whether to accept ICMP redirects.
firewall send-redirects <state>	Specifies whether to allow sending of ICMP redirects.
firewall syn-cookies <state>	Specifies whether to enable the TCP SYN cookies option.
Firewall Instances (Rule Set)	
firewall name <name>	Defines a firewall instance, or rule set.
firewall name <name> description <desc>	Specifies a brief description for a firewall rule set.
Firewall Rules	
firewall name <name> rule <rule-num>	Specifies a firewall rule within a rule set.
firewall name <name> rule <rule-num> action <action>	Specifies the action to perform on packets that match the criteria specified in this firewall rule.
firewall name <name> rule <rule-num> description <desc>	Specifies a brief description for a firewall rule.
firewall name <name> rule <rule-num> destination	Specifies the destination address and port to match in a firewall rule.
firewall name <name> rule <rule-num> icmp	Specifies ICMP code and type settings for a firewall rule.
firewall name <name> rule <rule-num> ipsec	Specifies IPSEC packet matching.
firewall name <name> rule <rule-num> log <state>	Enables or disables logging of firewall rule actions.

firewall name <name> rule <rule-num> protocol <protocol>	Specifies the protocol to which a firewall rule applies.
firewall name <name> rule <rule-num> source	Specifies the source address and port to match in a firewall rule.
firewall name <name> rule <rule-num> state	Specifies the kinds of packets to which this rule is applied.
Firewall on Ethernet Interfaces	
interfaces ethernet <ethx> firewall	Applies a firewall instance to an Ethernet interface.
Firewall on Ethernet Vifs	
interfaces ethernet <ethx> vif <vlan-id> firewall	Applies a firewall instance to an Ethernet vif.
Firewall on Ethernet Link Bonding Interfaces	
interfaces bonding <bondx> firewall	Applies a firewall instance to an Ethernet link bonding interface.
interfaces bonding <bondx> vif <vlan-id> firewall	Applies a firewall instance to an Ethernet link bonding interface vif.
Firewall on PPPoE Interfaces	
interfaces ethernet <ethx> pppoe <num> firewall	Applies a firewall instance to a PPPoE interface.
Firewall on Serial Interfaces	
interfaces serial <wanx> cisco-hdlc vif 1 firewall	Applies a firewall instance to a Cisco HDLC–encapsulated serial interface.
interfaces serial <wanx> frame-relay vif <dlci> firewall	Applies a firewall instance to a Frame Relay–encapsulated serial interface.
interfaces serial <wanx> ppp vif 1 firewall	Applies a firewall instance to a PPP-encapsulated serial interface.
Firewall on ADSL Interfaces	
interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet firewall	Applies a firewall instance to an ADSL PVC with RFC 1483 Bridged Ethernet encapsulation.
interfaces adsl <adslx> pvc <pvc-id> classical-ipoa firewall	Applies a firewall instance to an ADSL PVC with RFC 1577 Classical IPOA encapsulation.
interfaces adsl <adslx> pvc <pvc-id> pppoa <num> firewall	Applies a firewall instance to an ADSL PVC with PPPoA encapsulation.
interfaces adsl <adslx> pvc <pvc-id> pppoe <num> firewall	Applies a firewall instance to an ADSL PVC with PPPoE encapsulation.
Firewall on Tunnel Interfaces	

interfaces tunnel <tunx> firewall	Applies named firewall instances (packet-filtering rule sets) to a tunnel interface.
Firewall on OpenVPN Interfaces	
interfaces openvpn <vtunx> firewall	Applies a firewall instance to an OpenVPN interface.
Firewall on Wireless Modem Interfaces	
interfaces wirelessmodem <wlmx> firewall	Applies named firewall instances (packet-filtering rule sets) to a wirelessmodem interface.
Operational Commands	
clear firewall name <name> counters	Clears all statistics associated with the specified firewall instance.
show firewall	Displays rules associated with a firewall instance.
show firewall <name> statistics	Displays statistics information for a firewall instance.

clear firewall name <name> counters

Clears all statistics associated with the specified firewall instance.

Syntax

clear firewall name *name* **counters**

Command Mode

Operational mode.

Parameters

name	The name of the firewall instance where statistics are to be cleared.

Default

None.

Usage Guidelines

Use this command to clear the statistics associated with a specific firewall instance.

firewall

Enables firewall on the system.

Syntax

set firewall

delete firewall

show firewall

Command Mode

Configuration mode.

Configuration Statement

```
firewall {
}
```

Parameters

None.

Default

None.

Usage Guidelines

A firewall has no effect on traffic traversing the system or destined to the system until it has been applied to an interface using the **interfaces ethernet <ethx> firewall** command (see page 66).

Note that after the final user-defined rule in a rule set is executed, an implicit rule of **reject all** takes effect.

Use this command to specify a firewall configuration.

Use the **set** form of this command to create the firewall configuration.

Use the **delete** form of this command to remove the firewall configuration.

Use the **show** form of this command to view the firewall configuration.

firewall broadcast-ping <state>

Specifies whether the system will respond to ICMP Echo request messages sent to an IP broadcast address.

Syntax

set firewall broadcast-ping {**enable** | **disable**}

delete firewall broadcast-ping

show firewall broadcast-ping

Command Mode

Configuration mode.

Configuration Statement

```
firewall {
   broadcast-ping [enable|disable]
}
```

Parameters

enable	The system will respond to ICMP Echo requests sent to the broadcast address.
disable	The system will ignore ICMP Echo requests sent to the broadcast address.

Default

The default is **disable**.

Usage Guidelines

Use this command to specify whether the system will respond to ICMP Echo request messages sent to an IP broadcast address.

Use the **set** form of this command to specify whether the system will respond to ICMP Echo request messages sent to an IP broadcast address.

Use the **delete** form of this command to remove the specified value.

Use the **show** form of this command to view the specified value.

firewall ip-src-route <state>

Specifies whether to permit or deny packets with the Loose Source Route or Strict Source Route IP options.

Syntax

set firewall ip-src-route {**enable** | **disable**}

delete firewall ip-src-route

show firewall ip-src-route

Command Mode

Configuration mode.

Configuration Statement

```
firewall {
   ip-src-route [enable|disable]
}
```

Parameters

enable	Permits packets with source routing IP options set.
disable	Drops packets with source routing IP options set.

Default

The default is **disable**.

Usage Guidelines

Source routing allows applications to override the routing tables and specify one or more intermediate destinations for outgoing datagrams. This capability is sometimes used for troubleshooting, but renders the network vulnerable to attacks where network traffic is transparently directed to a centralized collection point for packet capture.

Use this command to specify whether to permit or deny packets with the Loose Source Route or Strict Source Route IP options.

Use the **set** form of this command to specify whether to permit or deny packets with the Loose Source Route or Strict Source Route IP options.

Use the **delete** form of this command to remove the specified value.

Use the **show** form of this command to view the specified value.

firewall log-martians <state>

Specifies whether to log packets with impossible addresses.

Syntax

set firewall log-martians {**enable** | **disable**}

delete firewall log-martians

show firewall log-martians

Command Mode

Configuration mode.

Configuration Statement

```
firewall {
    log-martians [enable|disable]
}
```

Parameters

enable	Records packets with impossible addresses in the log.
disable	Does not record packets with impossible addresses in the log.

Default

The default is **enable**.

Usage Guidelines

Use this command to specify whether to log packets with impossible addresses.

Use the **set** form of this command to specify whether to log packets with impossible addresses.

Use the **delete** form of this command to remove the specified value.

Use the **show** form of this command to view the specified value.

firewall name <name>

Defines a firewall instance, or rule set.

Syntax

set firewall name *name*

delete firewall name [*name*]

show firewall name [*name*]

Command Mode

Configuration mode.

Configuration Statement

```
firewall {
   name text {
   }
}
```

Parameters

name	Mandatory. The name of the firewall instance.

Default

None.

Usage Guidelines

Use this command to specify the name of a firewall instance.

A firewall instance is a named packet-filtering rule sets consisting of up to 1024 rules. Following the 1024 configurable rules is an implicit “deny all” rule.

Use the **set** form of this command to specify the name of a firewall instance.

Use the **delete** form of this command to remove the instance identified by the specified name.

Use the **show** form of this command to view the instance identified by the specified name.

firewall name <name> description <desc>

Specifies a brief description for a firewall rule set.

Syntax

set firewall name *name* **description** *desc*

delete firewall name *name* **description**

show firewall name *name* **description**

Command Mode

Configuration mode.

Configuration Statement

```
firewall {
   name text {
      description text
   }
}
```

Parameters

name	Mandatory. The name of the firewall instance.
desc	A description of the rule set. If the description contains spaces, it must be enclosed in double quotes.

Default

None.

Usage Guidelines

Use this command to specify a description of a firewall instance.

Use the **set** form of this command to specify a description of the firewall instance identified by the specified name.

Use the **delete** form of this command to remove the description of the instance identified by the specified name.

Use the **show** form of this command to view the description of the instance identified by the specified name.

firewall name <name> rule <rule-num>

Specifies a firewall rule within a rule set.

Syntax

set firewall name *name* **rule** *rule-num*

delete firewall name *name* **rule** [*rule-num*]

show firewall name *name* **rule** [*rule-num*]

Command Mode

Configuration mode.

Configuration Statement

```
firewall {
   name text {
      rule 1-1024 {
      }
   }
}
```

Parameters

name	Mandatory. The name of the firewall instance.
rule-num	Mandatory. Defines a firewall rule within the rule set. The rule number specifies the order in which this rule appears in the firewall rule table. Each rule must have a unique rule number. The range is 1 to 1024.

Default

None.

Usage Guidelines

Use this command to define a firewall instance. A firewall instance consists of a rule set of up to 1024 rules. Following the 1024 configurable rules is an implicit “deny all” rule.

Firewall rules are evaluated in sequence according to rule number. This is different from NAT, where rules are evaluated in the order in which they were configured, regardless of rule number.

Keep in mind that once assigned, a rule number cannot be changed because it is the identifier of the configuration node. If you think you might want to insert rules into your rule set later on, a good practice is to number rules in increments of 10. This leaves room for the addition of other rules.

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

Use the **set** form of this command to specify a firewall rule.

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

Use the **show** form of this command to view a firewall rule.

firewall name <name> rule <rule-num> action <action>

Specifies the action to perform on packets that match the criteria specified in this firewall rule.

Syntax

set firewall name *name* **rule** *rule-num* **action** *action*

delete firewall name *name* **rule** *rule-num* **action**

show firewall name *name* **rule** *rule-num* **action**

Command Mode

Configuration mode.

Configuration Statement

```
firewall {
   name text {
      rule 1-1024 {
         action [accept|drop|reject]
      }
   }
}
```

Parameters

name	Mandatory. The name of the firewall instance.
rule-num	Mandatory. The identifier of a firewall rule within the rule set. The range is 1 to 1024.
action	The action to be taken when the rule is matched. Supported values are as follows: **accept**: Accepts and forwards packets matching the criteria. **drop**: Silently drops packets matching the criteria. **reject**: Drops packets matching the criteria with a TCP reset.

Default

None.

Usage Guidelines

Use this command to specify the action to perform on packets that match the criteria specified in this firewall rule. Only one action can be defined for a rule.

Use the **set** form of this command to specify the action to perform on packets that match the criteria specified in this firewall rule.

Use the **delete** form of this command to remove the action.

Use the **show** form of this command to view the action.

firewall name <name> rule <rule-num> description <desc>

Specifies a brief description for a firewall rule.

Syntax

set firewall name *name* **rule** *rule-num* **description** *desc*

delete firewall name *name* **rule** *rule-num* **description**

show firewall name *name* **rule** *rule-num* **description**

Command Mode

Configuration mode.

Configuration Statement

```
firewall {
   name text {
      rule 1-1024 {
         description text
      }
   }
}
```

Parameters

name	Mandatory. The name of the firewall instance.
rule-num	Mandatory. The identifier of a firewall rule within the rule set. The range is 1 to 1024.
desc	A brief description for this rule. If the description contains spaces, it must be enclosed in double quotes.

Default

None.

Usage Guidelines

Use this command to specify a brief description for a firewall rule.

Use the **set** form of this command to set the description.

Use the **delete** form of this command to remove the description.

Use the **show** form of this command to view description configuration.

firewall name <name> rule <rule-num> destination

Specifies the destination address and port to match in a firewall rule.

Syntax

set firewall name *name* **rule** *rule-num* **destination** [**address** *address* | **port** *port*]

delete firewall name *name* **rule** *rule-num* **destination** [**address** | **port**]

show firewall name *name* **rule** *rule-num* **destination** [**address** | **port**]

Command Mode

Configuration mode.

Configuration Statement

```
firewall {
   name text {
      rule 1-1024 {
         destination {
            address text
            port text
         }
      }
   }
}
```

Parameters

name	Mandatory. The name of the firewall instance.
rule-num	Mandatory. The identifier of a firewall rule within the rule set. The range is 1 to 1024.

address	The destination address to match. The following formats are valid: *ip-address*: Matches the specified IP address. *ip-address*/*prefix*: A network address, where 0.0.0.0/0 matches any network. *ip-address*–*ip-address*: Matches a range of contiguous IP addresses; for example, 192.168.1.1–192.168.1.150. **!***ip-address*: Matches all IP addresses except the one specified. **!***ip-address*/*prefix*: Matches all network addresses except the one specified. **!***ip-address*–*ip-address*: Matches all IP addresses except those in the specified range.
port	Applicable only when the protocol is TCP or UDP. The destination port to match. The following formats are valid: *port-name*: Matches the name of an IP service; for example, **http**. You can specify any service name in the file **etc/services**. *port-num*: Matches a port number. The range is 1 to 65535. start–end: Matches the specified range of ports; for example, 1001–1005. You can use a combination of these formats in a comma-separated list. You can also negate the entire list by prepending it with an exclamation mark (“!”); for example,**!22,telnet,http,123,1001-1005**.

Default

None.

Usage Guidelines

Use this command to specify the destination to match in a firewall rule.

Note that you should take care in using more than one “exclusion” rule (that is, a rule using the negation operation (“!”) in combination. NAT rules are evaluated sequentially, and a sequence of exclusion rules could result in unexpected behavior.

Use the **set** form of this command to create a firewall destination.

Use the **delete** form of this command to remove a firewall destination.

Use the **show** form of this command to view firewall destination configuration.

firewall name <name> rule <rule-num> icmp

Specifies ICMP code and type settings for a firewall rule.

Syntax

set firewall name *name* **rule** *rule-num* **icmp** {**type** *type* | **code** *code*}

delete firewall name *name* **rule** *rule-num* **icmp** [**type** | **code**]

show firewall name *name* **rule** *rule-num* **icmp** [**type** | **code**]

Command Mode

Configuration mode.

Configuration Statement

```
firewall {
   name text {
      rule 1-1024 {
         icmp {
            type u32
            code u32
         }
      }
}
```

Parameters

name	Mandatory. The name of the firewall instance.
rule-num	Mandatory. The identifier of a firewall rule within the rule set. The range is 1 to 1024.
type	A valid ICMP type code from 0 to 255; for example, 8 (Echo Request), or **0** (Echo Reply), or the keyword **all**. The default is all. For a list of ICMP codes and types, see “Appendix A: ICMP Types.”
code	The ICMP type code associated with this ICMP type. The range is 0 to 255. For a list of ICMP codes and types, see “Appendix A: ICMP Types.”

Default

None.

Usage Guidelines

Use this command to define the ICMP types this rule applies to—for example Echo Request or Echo Reply. Packets having this ICMP type will “match” the rule.

Use the **set** form of this command to specify the ICMP code and type for the specified rule

Use the **delete** form of this command to remove the ICMP code or type value for the specified rule.

Use the **show** form of this command to view the ICMP code or type value for the specified rule.

firewall name <name> rule <rule-num> ipsec

Specifies IPSEC packet matching.

Syntax

set firewall name *name* **rule** *rule-num* **ipsec {match-ipsec|match-none}**

delete firewall name *name* **rule** *rule-num* **ipsec [match-ipsec|match-none]**

show firewall name *name* **rule** *rule-num* **ipsec**

Command Mode

Configuration mode.

Configuration Statement

```
firewall {
   name text {
      rule 1-1024 {
         ipsec {
            match-ipsec
            match-none
         }
      }
   }
}
```

Parameters

name	Mandatory. The name of the firewall instance.
rule-num	Mandatory. The identifier of a firewall rule within the rule set. The range is 1 to 1024.
match-ipsec	Match inbound IPsec packets.
match-none	Match inbound non-IPsec packets.

Default

None.

Usage Guidelines

Use this command to specify whether to match IPsec or non-IPsec packets.

Use the **set** form of this command to specify which type of packets to match.

Use the **delete** form of this command to remove the configuration.

Use the **show** form of this command to view the configuration.

firewall name <name> rule <rule-num> log <state>

Enables or disables logging of firewall rule actions.

Syntax

set firewall name *name* **rule** *rule-num* **log** *state*

delete firewall name *name* **rule** *rule-num* **log**

show firewall name *name* **rule** *rule-num* **log**

Command Mode

Configuration mode.

Configuration Statement

```
firewall {
    name text {
        rule 1-1024 {
            log [enable|disable]
        }
    }
}
```

Parameters

name	Mandatory. The name of the firewall instance.
rule-num	Mandatory. The identifier of a firewall rule within the rule set. The range is 1 to 1024.
state	Enables or disables logging of firewall actions. Supported values are as follows: **enable**: Log when action is taken. **disable**: Do not log when action is taken.

Default

Actions are not logged.

Usage Guidelines

Use this command to enable or disable logging for the specified rule. When enabled, any actions taken will be logged.

Use the **set** form of this command to specify logging for the specified rule

Use the **delete** form of this command to remove the logging value for the specified rule.

Use the **show** form of this command to view the logging value for the specified rule.

firewall name <name> rule <rule-num> protocol <protocol>

Specifies the protocol to which a firewall rule applies.

Syntax

set firewall name *name* **rule** *rule-num* **protocol** *protocol*

delete firewall name *name* **rule** *rule-num* **protocol**

show firewall name *name* **rule** *rule-num* **protocol**

Command Mode

Configuration mode.

Configuration Statement

```
firewall {
   name text {
      rule 1-1024 {
         protocol text
      }
   }
}
```

Parameters

name	Mandatory. The name of the firewall instance.
rule-num	Mandatory. The identifier of a firewall rule within the rule set. The range is 1 to 1024.
protocol	Mandatory. Any protocol literals or numbers listed in the file **/etc/protocols** can be used. The keyword **all** is also supported. Prefixing the protocol name with the exclamation mark character (“!”) matches every protocol except the specified protocol. For example, **!tcp** matches all protocols except TCP.

Default

The default is **all**.

Usage Guidelines

Use this command to define to which protocol a firewall rule applies. Packets using this protocol will “match” the rule.

Note that you should take care in using more than one “exclusion” rule (that is, a rule using the negation operation (“!”) in combination. NAT rules are evaluated sequentially, and a sequence of exclusion rules could result in unexpected behavior.

Use the **set** form of this command to specify the protocol to match for the specified rule

Use the **delete** form of this command to remove the protocol value for the specified rule.

Use the **show** form of this command to view the protocol value for the specified rule.

firewall name <name> rule <rule-num> source

Specifies the source address and port to match in a firewall rule.

Syntax

set firewall name *name* **rule** *rule-num* **source** [**address** *address* | **port** *port* | **mac-address** *mac-addr*]

delete firewall name *name* **rule** *rule-num* **source** [**address** | **port**]

show firewall name *name* **rule** *rule-num* **source** [**address** | **port**]

Command Mode

Configuration mode.

Configuration Statement

```
firewall {
   name text {
      rule 1-1024 {
         source {
            address text
            port text
         }
      }
   }
}
```

Parameters

name	Mandatory. The name of the firewall instance.
rule-num	Mandatory. The identifier of a firewall rule within the rule set. The range is 1 to 1024.

address	The source address to match. The following formats are valid: *ip-address*: Matches the specified IP address. *ip-address*/*prefix*: A network address, where 0.0.0.0/0 matches any network. *ip-address*–*ip-address*: Matches a range of contiguous IP addresses; for example, 192.168.1.1–192.168.1.150. **!***ip-address*: Matches all IP addresses except the one specified. **!***ip-address*/*prefix*: Matches all network addresses except the one specified. **!***ip-address*–*ip-address*: Matches all IP addresses except those in the specified range.
port	The source port to match. The following formats are valid: *port-name*: Matches the name of an IP service; for example, **http**. You can specify any service name in the file **etc/services**. *port-num*: Matches a port number. The range is 1 to 65535. start–end: Matches the specified range of ports; for example, 1001–1005. You can use a combination of these formats in a comma-separated list. You can also negate the entire list by prepending it with an exclamation mark (“!”); for example, **!22,telnet,http,123,1001-1005**.
mac-addr	The media access control (MAC) address to match. The format is 6 colon-separated 8-bit numbers in hexadecimal; for example, 00:0a:59:9a:f2:ba.

Default

None.

Usage Guidelines

Use this command to specify the source to match in a firewall rule.

Note that you should take care in using more than one “exclusion” rule (that is, a rule using the negation operation (“!”) in combination. NAT rules are evaluated sequentially, and a sequence of exclusion rules could result in unexpected behavior.

Use the **set** form of this command to create a firewall source.

Use the **delete** form of this command to remove a firewall source.

Use the **show** form of this command to view firewall source configuration.

firewall name <name> rule <rule-num> state

Specifies the kinds of packets to which this rule is applied.

Syntax

set firewall name *name* **rule** *rule-num* **state** {**established** *state* | **invalid** *state* | **new** *state* | **related** *state*}

delete firewall name *name* **rule** *rule-num* **state**

show firewall name *name* **rule** *rule-num* **state**

Command Mode

Configuration mode.

Configuration Statement

```
firewall {
   name text {
      rule 1-1024 {
         state {
               established [enable|disable]
               invalid [enable|disable]
               new [enable|disable]
               related [enable|disable]
         }
      }
   }
}
```

Parameters

name	Mandatory. The name of the firewall instance.
rule-num	Mandatory. The identifier of a firewall rule within the rule set. The range is 1 to 1024.
established *state*	Specifies whether or not the rule will be applied to established packets. Supported values are as follows: **enable**: Applies the rule to established packets. **disable**: Does not apply the rule to established packets.

invalid *state*	Specifies whether or not the rule will be applied to invalid packets. Supported values are as follows: **enable**: Applies the rule to invalid packets. **disable**: Does not apply the rule to invalid packets.
new *state*	Specifies whether or not the rule will be applied to new packets. Supported values are as follows: **enable**: Applies the rule to new packets. **disable**: Does not apply the rule to new packets.
related *state*	Specifies whether or not the rule will be applied to related packets. Supported values are as follows: **enable**: Applies the rule to related packets. **disable**: Does not apply the rule to related packets.

Default

The rule is applied to all packets, regardless of state.

Usage Guidelines

Use this command to specify the kind of packets this rule will be applied to.

- *Established* packets are packets that are part of a connection that has seen packets in both directions; for example, a reply packet, or an outgoing packet on a connection that has been replied to.
- *Invalid* packets are packets that could not be identified for some reason. These might include the system running out of resource, or ICMP errors that do not correspond to any known connection. Generally these packets should be dropped.
- *New* packets are packets creating new connections. For TCP, this will be packets with the SYN flag set.
- *Related* packets are packets related to existing connections.

Use the **set** form of this command to specify the kind of packets a firewall rule will be applied to.

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

Use the **show** form of this command to view state configuration for a firewall rule.

firewall receive-redirects <state>

Specifies whether to accept ICMP redirects.

Syntax

set firewall receive-redirects *state*

delete firewall receive-redirects

show firewall receive-redirects

Command Mode

Configuration mode.

Configuration Statement

```
firewall {
   receive-redirects [enable|disable]
}
```

Parameters

state	Permits or denies receiving packets with ICMP redirects. Supported values are as follows: **enable**: Permits packets with ICMP redirects to be received. **disable**: Denies packets with ICMP redirects to be received.

Default

The default is **disable**.

Usage Guidelines

Use this command to specify whether to accept ICMP redirects. ICMP redirects can allow an arbitrary sender to forge packets and alter the system's routing table. This can leave the system open to a man-in-the-middle attack.

Use the **set** form of this command to specify whether to accept ICMP redirects.

Use the **delete** form of this command to remove the specified value.

Use the **show** form of this command to view the specified value.

firewall send-redirects <state>

Specifies whether to allow sending of ICMP redirects.

Syntax

set firewall send-redirects *state*

delete firewall send-redirects

show firewall send-redirects

Command Mode

Configuration mode.

Configuration Statement

```
firewall {
   send-redirects [enable|disable]
}
```

Parameters

state	Permits or denies transmitting packets with ICMP redirects. Supported values are as follows: **enable**: Permits packets with ICMP redirects to be sent. **disable**: Denies packets with ICMP redirects to be sent.

Default

The default is **disable**.

Usage Guidelines

Use this command to specify whether to allow sending of ICMP redirects. Sending a redirect will potentially alter the routing table of the host or router to which the redirect is sent.

Use the **set** form of this command to specify whether to permit or deny the sending ICMP redirects.

Use the **delete** form of this command to remove the specified value.

Use the **show** form of this command to view the specified value.

firewall syn-cookies <state>

Specifies whether to enable the TCP SYN cookies option.

Syntax

set firewall syn-cookies *state*

delete firewall syn-cookies

show firewall syn-cookies

Command Mode

Configuration mode.

Configuration Statement

```
firewall {
   syn-cookies [enable|disable]
}
```

Parameters

state	Enables or disables TCP SYN cookies option. Supported values are as follows: **enable**: Enables TCP SYN cookies option. **disable**: Disables TCP SYN cookies option.

Default

The default is **disable**.

Usage Guidelines

Use this command to specify whether to use the TCP SYN cookies option. Enabling this option can help protect the system from a TCP SYN Flood Denial of Service (DoS) attack.

To start a TCP connection, a source sends a SYN (synchronize/start) packet. The destination sends back a SYN ACK (synchronize acknowledge). Then the source sends an ACK (acknowledge), and the connection is established. This is referred to as the “TCP three-way handshake.”

After a destination server sends a SYN ACK, it uses a connection queue to keep track of the connections waiting to be completed. An attacker can fill up the connection queue by generating phony TCP SYN packets from random IP addresses at a rapid rate. When the connection queue is full, all subsequent TCP services are denied.

When this option is enabled, the system creates a hash entry when it receives a SYN packet, and returns a SYN ACK cookie only, without retaining all the SYN information. When it receives the ACK from the client, it validates it against the hash and, if it is valid, rebuilds the SYN packet information and accepts the packet.

Use the **set** form of this command to specify whether to enable or disable the TCP SYN cookies option.

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

Use the **show** form of this command to view TCP SYN cookies option configuration.

interfaces adsl <adslx> pvc <pvc-id> bridged-ethernet firewall

Applies a firewall instance to an ADSL PVC with RFC 1483 Bridged Ethernet encapsulation.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **bridged-ethernet firewall** {**in name** *fw-name* | **local name** *fw-name* | **out name** *fw-name*}

delete interfaces adsl *adslx* **pvc** *pvc-id* **bridged-ethernet firewall** [**in** | **local** | **out**]

show interfaces adsl *adslx* **pvc** *pvc-id* **bridged-ethernet firewall** [**in** | **local** | **out**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535 | auto] {
         bridged-ethernet {
            firewall {
               in {
                  name text
               }
               local {
                  name text
               }
               out {
                  name text
               }
            }
         }
      }
   }
}
```

Parameters

adslx	Mandatory. The name of the interface. This can be the name of a PPPoA-, PPPoE-, Classical IPOA-, or Bridged Ethernet-encapsulated DSL interface; that is the interface name can be **pppoa***x*, **pppoe***x*, or **adsl***x*.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
in name *fw-name*	Applies the specified firewall instance to inbound traffic on the specified interface.
local name *fw-name*	Applies the specified firewall instance to traffic arriving on the specified interface and bound for the local system.
out name *fw-name*	Applies the specified firewall instance to outbound traffic on the specified interface.

Default

None.

Usage Guidelines

Use this command to apply a firewall instance, or rule set, to a PVC with RFC 1483 Bridged Ethernet encapsulation on an ADSL interface.

A firewall has no effect on traffic traversing the system or destined to the system until a firewall rule set has been applied to an interface or a vif using this command.

To use the firewall feature, you define a firewall rule set as a named firewall instance, using the **firewall** command (see page 18). You then apply the firewall instance to interfaces and/or vifs using a statement like this one. Once applied, the instance acts as a packet filter.

The firewall instance will filter packets in one of the following ways, depending on what you specify when you apply it:

- **in.** If you apply the rule set as **in**, the firewall will filter packets entering the interface.
- **out.** If you apply the rule set as **out**, the firewall will filter packets leaving the interface.
- **local.** If you apply the rule set as **local**, the firewall will filter packets destined for the system itself.

For each interface, you can apply up to three firewall instances: one firewall **in** instance, one firewall **out** instance, and one firewall **local** instance.

Make sure the firewall instance you apply to an interface is already defined, or you may experience unintended results. If you apply a firewall instance that does not exist to an interface, the implicit firewall rule of **allow all** will be applied.

Use the **set** form of this command to apply a firewall instance to a PVC with Bridged Ethernet encapsulation on an ADSL interface.

Use the **delete** form of this command to remove a firewall instance from a PVC with Bridged Ethernet encapsulation on an ADSL interface.

Use the **show** form of this command to view a firewall configuration for a PVC with Bridged Ethernet encapsulation on an ADSL interface.

interfaces adsl <adslx> pvc <pvc-id> classical-ipoa firewall

Applies a firewall instance to an ADSL PVC with RFC 1577 Classical IPOA encapsulation.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **classical-ipoa firewall** {**in name** *fw-name* | **local name** *fw-name* | **out name** *fw-name*}

delete interfaces adsl *adslx* **pvc** *pvc-id* **classical-ipoa firewall** [**in** | **local** | **out**]

show interfaces adsl *adslx* **pvc** *pvc-id* **classical-ipoa firewall** [**in** | **local** | **out**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    adsl adslx {
        pvc [0-255/0-65535 | auto] {
            classical-ipoa {
                firewall {
                    in {
                        name text
                    }
                    local {
                        name text
                    }
                    out {
                        name text
                    }
                }
            }
        }
    }
}
```

Parameters

adslx	Mandatory. The name of the interface. This can be the name of a PPPoA-, PPPoE-, Classical IPOA-, or Bridged Ethernet-encapsulated DSL interface; that is the interface name can be **pppoa***x*, **pppoe***x*, or **adsl***x*.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
in name *fw-name*	Applies the specified firewall instance to inbound traffic on the specified interface.
local name *fw-name*	Applies the specified firewall instance to traffic arriving on the specified interface and bound for the local system.
out name *fw-name*	Applies the specified firewall instance to outbound traffic on the specified interface.

Default

None.

Usage Guidelines

Use this command to apply a firewall instance, or rule set, to a PVC with RFC 1577 Classical IP over Asynchronous Transfer Mode (IPOA) encapsulation on an ADSL interface.

A firewall has no effect on traffic traversing the system or destined to the system until a firewall rule set has been applied to an interface or a vif using this command.

To use the firewall feature, you define a firewall rule set as a named firewall instance, using the **firewall** command (see page 18). You then apply the firewall instance to interfaces and/or vifs using a statement like this one. Once applied, the instance acts as a packet filter.

The firewall instance will filter packets in one of the following ways, depending on what you specify when you apply it:

- **in.** If you apply the rule set as **in**, the firewall will filter packets entering the interface.
- **out.** If you apply the rule set as **out**, the firewall will filter packets leaving the interface.

- **local.** If you apply the rule set as **local**, the firewall will filter packets destined for the system itself.

For each interface, you can apply up to three firewall instances: one firewall **in** instance, one firewall **out** instance, and one firewall **local** instance.

Make sure the firewall instance you apply to an interface is already defined, or you may experience unintended results. If you apply a firewall instance that does not exist to an interface, the implicit firewall rule of **allow all** will be applied.

Use the **set** form of this command to apply a firewall instance to a PVC with Clasical IPOA encapsulation on an ADSL interface.

Use the **delete** form of this command to remove a firewall instance from a PVC with Clasical IPOA encapsulation on an ADSL interface.

Use the **show** form of this command to view a firewall configuration for a PVC with Clasical IPOA encapsulation on an ADSL interface.

interfaces adsl <adslx> pvc <pvc-id> pppoa <num> firewall

Applies a firewall instance to an ADSL PVC with PPPoA encapsulation.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **firewall** {**in name** *fw-name* | **local name** *fw-name* | **out name** *fw-name*}

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **firewall** [**in** | **local** | **out**]

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoa** *num* **firewall** [**in** | **local** | **out**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535 | auto] {
         pppoa 0-15 {
            firewall {
               in {
                  name text
               }
               local {
                  name text
               }
               out {
                  name text
               }
            }
         }
      }
   }
}
```

Parameters

adslx	Mandatory. The name of the interface. This can be the name of a PPPoA-, PPPoE-, Classical IPOA-, or Bridged Ethernet-encapsulated DSL interface; that is the interface name can be **pppoa***x*, **pppoe***x*, or **adsl***x*.
pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
num	Mandatory. The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
in name *fw-name*	Applies the specified firewall instance to inbound traffic on the specified interface.
local name *fw-name*	Applies the specified firewall instance to traffic arriving on the specified interface and bound for the local system.
out name *fw-name*	Applies the specified firewall instance to outbound traffic on the specified interface.

Default

None.

Usage Guidelines

Use this command to apply a firewall instance, or rule set, to a PVC with Point-to-Point Protocol over Asynchronous Transfer Mode (PPPoA) encapsulation on an ADSL interface.

A firewall has no effect on traffic traversing the system or destined to the system until a firewall rule set has been applied to an interface or a vif using this command.

To use the firewall feature, you define a firewall rule set as a named firewall instance, using the **firewall** command (see page 18). You then apply the firewall instance to interfaces and/or vifs using a statement like this one. Once applied, the instance acts as a packet filter.

The firewall instance will filter packets in one of the following ways, depending on what you specify when you apply it:

- **in.** If you apply the rule set as **in**, the firewall will filter packets entering the interface.
- **out.** If you apply the rule set as **out**, the firewall will filter packets leaving the interface.
- **local.** If you apply the rule set as **local**, the firewall will filter packets destined for the system itself.

For each interface, you can apply up to three firewall instances: one firewall **in** instance, one firewall **out** instance, and one firewall **local** instance.

Make sure the firewall instance you apply to an interface is already defined, or you may experience unintended results. If you apply a firewall instance that does not exist to an interface, the implicit firewall rule of **allow all** will be applied.

Use the **set** form of this command to apply a firewall instance to a PVC with PPPoA encapsulation on an ADSL interface.

Use the **delete** form of this command to remove a firewall instance from a PVC with PPPoA encapsulation on an ADSL interface.

Use the **show** form of this command to view a firewall configuration for a PVC with PPPoA encapsulation on an ADSL interface.

interfaces adsl <adslx> pvc <pvc-id> pppoe <num> firewall

Applies a firewall instance to an ADSL PVC with PPPoE encapsulation.

Syntax

set interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **firewall** {**in name** *fw-name* | **local name** *fw-name* | **out name** *fw-name*}

delete interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **firewall** [**in** | **local** | **out**]

show interfaces adsl *adslx* **pvc** *pvc-id* **pppoe** *num* **firewall** [**in** | **local** | **out**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   adsl adslx {
      pvc [0-255/0-65535 | auto] {
         pppoe 0-15 {
            firewall {
            in {
               name text
            local {
               name text
            out {
               name text
            }
         }
      }
   }
}
```

Parameters

adslx	Mandatory. The name of the interface. This can be the name of a PPPoA-, PPPoE-, Classical IPOA-, or Bridged Ethernet-encapsulated DSL interface; that is the interface name can be **pppoa***x*, **pppoe***x*, or **adsl***x*.

pvc-id	Mandatory. The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
num	Mandatory. The name of a defined PPPoE unit. The range is 0 to 15.
in name *fw-name*	Applies the specified firewall instance to inbound traffic on the specified interface.
local name *fw-name*	Applies the specified firewall instance to traffic arriving on the specified interface and bound for the local system.
out name *fw-name*	Applies the specified firewall instance to outbound traffic on the specified interface.

Default

None.

Usage Guidelines

Use this command to apply a firewall instance, or rule set, to a Point-to-Point over Ethernet (PPPoE) interface.

A firewall has no effect on traffic traversing the system or destined to the system until a firewall rule set has been applied to an interface or a vif using this command.

To use the firewall feature, you define a firewall rule set as a named firewall instance, using the **firewall** command (see page 18). You then apply the firewall instance to interfaces and/or vifs using a statement like this one. Once applied, the instance acts as a packet filter.

The firewall instance will filter packets in one of the following ways, depending on what you specify when you apply it:

- **in.** If you apply the rule set as **in**, the firewall will filter packets entering the interface.
- **out.** If you apply the rule set as **out**, the firewall will filter packets leaving the interface.
- **local.** If you apply the rule set as **local**, the firewall will filter packets destined for the system itself.

For each interface, you can apply up to three firewall instances: one firewall **in** instance, one firewall **out** instance, and one firewall **local** instance.

Make sure the firewall instance you apply to an interface is already defined, or you may experience unintended results. If you apply a firewall instance that does not exist to an interface, the implicit firewall rule of **allow all** will be applied.

Use the **set** form of this command to apply a firewall instance to a PPPoE interface.

Use the **delete** form of this command to remove a firewall instance from a PPPoE interface.

Use the **show** form of this command to view a firewall configuration for a PPPoE interface.

interfaces bonding <bondx> firewall

Applies a firewall instance to an Ethernet link bonding interface.

Syntax

set interfaces bonding *bondx* **firewall** {**in name** *fw-name* | **local name** *fw-name* | **out name** *fw-name*}

delete interfaces bonding *bondx* **firewall** [**in** | **local** | **out**]

show interfaces bonding *bondx* **firewall** [**in** | **local** | **out**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   bonding bond0..bond99 {
      firewall {
         in {
            name text
         }
         local {
            name text
         }
         out {
            name text
         }
      }
   }
}
```

Parameters

bondx	The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
in name *fw-name*	Applies the specified firewall instance to inbound traffic on the specified interface.
local name *fw-name*	Applies the specified firewall instance to traffic arriving on the specified interface and bound for the local system.

out name *fw-name*	Applies the specified firewall instance to outbound traffic on the specified interface.

Default

None.

Usage Guidelines

Use this command to apply a firewall instance, or rule set, to an Ethernet link bonding interface.

A firewall has no effect on traffic traversing the system or destined to the system until a firewall rule set has been applied to an interface using this command.

To use the firewall feature, you define a firewall rule set as a named firewall instance, using the **firewall** command (see page 18). You then apply the firewall instance to interfaces using a statement like this one. Once applied, the instance acts as a packet filter.

The firewall instance will filter packets in one of the following ways, depending on what you specify when you apply it:

- **in.** If you apply the rule set as **in**, the firewall will filter packets entering the interface.
- **out.** If you apply the rule set as **out**, the firewall will filter packets leaving the interface.
- **local.** If you apply the rule set as **local**, the firewall will filter packets destined for the system itself.

For each interface, you can apply up to three firewall instances: one firewall **in** instance, one firewall **out** instance, and one firewall **local** instance.

Make sure the firewall instance you apply to an interface is already defined, or you may experience unintended results. If you apply a firewall instance that does not exist to an interface, the implicit firewall rule of **allow all** will be applied.

Use the **set** form of this command to apply a firewall instance to an Ethernet link bonding interface.

Use the **delete** form of this command to remove a firewall instance from an Ethernet link bonding interface.

Use the **show** form of this command to view a firewall configuration for an Ethernet link bonding interface.

interfaces bonding <bondx> vif <vlan-id> firewall

Applies a firewall instance to an Ethernet link bonding interface vif.

Syntax

set interfaces bonding *bondx* **vif** *vlan-id* **firewall** {**in name** *fw-name* | **local name** *fw-name* | **out name** *fw-name*}

delete interfaces bonding *bondx* **vif** *vlan-id* **firewall** [**in** | **local** | **out**]

show interfaces bonding *bondx* **vif** *vlan-id* **firewall** [**in** | **local** | **out**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   bonding bond0..bond99 {
      vif 0-4094 {
         firewall {
         in {
            name text
         local {
            name text
         out {
            name text
         }
      }
   }
}
```

Parameters

bondx	Mandatory. The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
vlan-id	Mandatory. The VLAN ID for the vif. The range is 0 to 4094.
in name *fw-name*	Applies the specified firewall instance to inbound traffic on the specified vif.
local name *fw-name*	Applies the specified firewall instance to traffic arriving on the specified vif and bound for the local system.

out name *fw-name*	Applies the specified firewall instance to outbound traffic on the specified vif.

Default

None.

Usage Guidelines

Use this command to apply a firewall instance, or rule set, to an Ethernet link bonding interface vif.

A firewall has no effect on traffic traversing the system or destined to the system until a firewall rule set has been applied to a vif using this command.

To use the firewall feature, you define a firewall rule set as a named firewall instance, using the **firewall** command (see page 18). You then apply the firewall instance to vifs using a statement like this one. Once applied, the instance acts as a packet filter.

The firewall instance will filter packets in one of the following ways, depending on what you specify when you apply it:

- **in.** If you apply the rule set as **in**, the firewall will filter packets entering the interface.
- **out.** If you apply the rule set as **out**, the firewall will filter packets leaving the interface.
- **local.** If you apply the rule set as **local**, the firewall will filter packets destined for the system itself.

For each interface, you can apply up to three firewall instances: one firewall **in** instance, one firewall **out** instance, and one firewall **local** instance.

Make sure the firewall instance you apply to an interface is already defined, or you may experience unintended results. If you apply a firewall instance that does not exist to an interface, the implicit firewall rule of **allow all** will be applied.

Use the **set** form of this command to apply a firewall instance to a vif.

Use the **delete** form of this command to remove a firewall instance from a vif.

Use the **show** form of this command to view a firewall configuration for a vif.

interfaces ethernet <ethx> firewall

Applies a firewall instance to an Ethernet interface.

Syntax

set interfaces ethernet *ethx* **firewall** {**in name** *fw-name* | **local name** *fw-name* | **out name** *fw-name*}

delete interfaces ethernet *ethx* **firewall** [**in** | **local** | **out**]

show interfaces ethernet *ethx* **firewall** [**in** | **local** | **out**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet eth0..eth23 {
      firewall {
         in {
            name text
         }
         local {
            name text
         }
         out {
            name text
         }
      }
   }
}
```

Parameters

ethx	The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
in name *fw-name*	Applies the specified firewall instance to inbound traffic on the specified interface.
local name *fw-name*	Applies the specified firewall instance to traffic arriving on the specified interface and bound for the local system.

out name *fw-name*	Applies the specified firewall instance to outbound traffic on the specified interface.

Default

None.

Usage Guidelines

Use this command to apply a firewall instance, or rule set, to an Ethernet interface.

A firewall has no effect on traffic traversing the system or destined to the system until a firewall rule set has been applied to an interface or a vif using this command.

To use the firewall feature, you define a firewall rule set as a named firewall instance, using the **firewall** command (see page 18). You then apply the firewall instance to interfaces and/or vifs using a statement like this one. Once applied, the instance acts as a packet filter.

The firewall instance will filter packets in one of the following ways, depending on what you specify when you apply it:

- **in.** If you apply the rule set as **in**, the firewall will filter packets entering the interface.
- **out.** If you apply the rule set as **out**, the firewall will filter packets leaving the interface.
- **local.** If you apply the rule set as **local**, the firewall will filter packets destined for the system itself.

For each interface, you can apply up to three firewall instances: one firewall **in** instance, one firewall **out** instance, and one firewall **local** instance.

Make sure the firewall instance you apply to an interface is already defined, or you may experience unintended results. If you apply a firewall instance that does not exist to an interface, the implicit firewall rule of **allow all** will be applied.

Use the **set** form of this command to apply a firewall instance to an Ethernet interface.

Use the **delete** form of this command to remove a firewall instance from an Ethernet interface.

Use the **show** form of this command to view a firewall configuration for an Ethernet interface.

interfaces ethernet <ethx> pppoe <num> firewall

Applies a firewall instance to a PPPoE interface.

Syntax

set interfaces ethernet *ethx* **pppoe** *num* **firewall** {**in name** *fw-name* | **local name** *fw-name* | **out name** *fw-name*}

delete interfaces ethernet *ethx* **pppoe** *num* **firewall** [**in** | **local** | **out**]

show interfaces ethernet *ethx* **pppoe** *num* **firewall** [**in** | **local** | **out**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet eth0..eth23 {
      pppoe 0-15 {
         firewall {
         in {
            name text
         local {
            name text
         out {
            name text
         }
      }
   }
}
```

Parameters

ethx	The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
num	Mandatory. The name of a defined PPPoE unit. The range is 0 to 15.
in name *fw-name*	Applies the specified firewall instance to inbound traffic on the specified interface.

local name *fw-name*	Applies the specified firewall instance to traffic arriving on the specified interface and bound for the local system.
out name *fw-name*	Applies the specified firewall instance to outbound traffic on the specified interface.

Default

None.

Usage Guidelines

Use this command to apply a firewall instance, or rule set, to a Point-to-Point over Ethernet (PPPoE) interface.

A firewall has no effect on traffic traversing the system or destined to the system until a firewall rule set has been applied to an interface or a vif using this command.

To use the firewall feature, you define a firewall rule set as a named firewall instance, using the **firewall** command (see page 18). You then apply the firewall instance to interfaces and/or vifs using a statement like this one. Once applied, the instance acts as a packet filter.

The firewall instance will filter packets in one of the following ways, depending on what you specify when you apply it:

- **in.** If you apply the rule set as **in**, the firewall will filter packets entering the interface.
- **out.** If you apply the rule set as **out**, the firewall will filter packets leaving the interface.
- **local.** If you apply the rule set as **local**, the firewall will filter packets destined for the system itself.

For each interface, you can apply up to three firewall instances: one firewall **in** instance, one firewall **out** instance, and one firewall **local** instance.

Make sure the firewall instance you apply to an interface is already defined, or you may experience unintended results. If you apply a firewall instance that does not exist to an interface, the implicit firewall rule of **allow all** will be applied.

Use the **set** form of this command to apply a firewall instance to a PPPoE interface.

Use the **delete** form of this command to remove a firewall instance from a PPPoE interface.

Use the **show** form of this command to view a firewall configuration for a PPPoE interface.

interfaces ethernet <ethx> vif <vlan-id> firewall

Applies a firewall instance to an Ethernet vif.

Syntax

set interfaces ethernet *ethx* **vif** *vlan-id* **firewall** {**in name** *fw-name* | **local name** *fw-name* | **out name** *fw-name*}

delete interfaces ethernet *ethx* **vif** *vlan-id* **firewall** [**in** | **local** | **out**]

show interfaces ethernet *ethx* **vif** *vlan-id* **firewall** [**in** | **local** | **out**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet eth0..eth23 {
      vif 0-4094 {
         firewall {
         in {
            name text
         local {
            name text
         out {
            name text
         }
      }
   }
}
```

Parameters

ethx	The Ethernet interface you are configuring: one of **eth0** through **eth23**. The interface must already have been defined.
vlan-id	The VLAN ID for the vif. The range is 0 to 4094.
in name *fw-name*	Applies the specified firewall instance to inbound traffic on the specified vif.
local name *fw-name*	Applies the specified firewall instance to traffic arriving on the specified vif and bound for the local system.

out name *fw-name*	Applies the specified firewall instance to outbound traffic on the specified vif.

Default

None.

Usage Guidelines

Use this command to apply a firewall instance, or rule set, to an Ethernet vif.

A firewall has no effect on traffic traversing the system or destined to the system until a firewall rule set has been applied to an interface or a vif using this command.

To use the firewall feature, you define a firewall rule set as a named firewall instance, using the **firewall** command (see page 18). You then apply the firewall instance to interfaces and/or vifs using a statement like this one. Once applied, the instance acts as a packet filter.

The firewall instance will filter packets in one of the following ways, depending on what you specify when you apply it:

- **in.** If you apply the rule set as **in**, the firewall will filter packets entering the interface.
- **out.** If you apply the rule set as **out**, the firewall will filter packets leaving the interface.
- **local.** If you apply the rule set as **local**, the firewall will filter packets destined for the system itself.

For each interface, you can apply up to three firewall instances: one firewall **in** instance, one firewall **out** instance, and one firewall **local** instance.

Make sure the firewall instance you apply to an interface is already defined, or you may experience unintended results. If you apply a firewall instance that does not exist to an interface, the implicit firewall rule of **allow all** will be applied.

Use the **set** form of this command to apply a firewall instance to an Ethernet vif.

Use the **delete** form of this command to remove a firewall instance from an Ethernet vif.

Use the **show** form of this command to view a firewall configuration for an Ethernet vif.

interfaces openvpn <vtunx> firewall

Applies a firewall instance to an OpenVPN interface.

Syntax

set interfaces openvpn *vtunx* **firewall** {**in name** *fw-name* | **local name** *fw-name* | **out name** *fw-name*}

delete interfaces openvpn *vtunx* **firewall** [**in** | **local** | **out**]

show interfaces openvpn *vtunx* **firewall** [**in** | **local** | **out**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    openvpn vtun0..vtunx {
        firewall {
            in {
                name text
            }
            local {
                name text
            }
            out {
                name text
            }
        }
    }
}
```

Parameters

vtunx	Mandatory. Multi-node. The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
in name *fw-name*	Applies the specified firewall instance to inbound traffic on the specified interface.
local name *fw-name*	Applies the specified firewall instance to traffic arriving on the specified interface and bound for the local system.

out name *fw-name*	Applies the specified firewall instance to outbound traffic on the specified interface.

Default

None.

Usage Guidelines

Use this command to apply a firewall instance, or rule set, to an OpenVPN interface.

A firewall has no effect on traffic traversing the system or destined to the system until a firewall rule set has been applied to an interface using this command.

To use the firewall feature, you define a firewall rule set as a named firewall instance, using the **firewall** command (see page 18). You then apply the firewall instance to interfaces using a statement like this one. Once applied, the instance acts as a packet filter.

The firewall instance will filter packets in one of the following ways, depending on what you specify when you apply it:

- **in.** If you apply the rule set as **in**, the firewall will filter packets entering the interface.
- **out.** If you apply the rule set as **out**, the firewall will filter packets leaving the interface.
- **local.** If you apply the rule set as **local**, the firewall will filter packets destined for the system itself.

For each interface, you can apply up to three firewall instances: one firewall **in** instance, one firewall **out** instance, and one firewall **local** instance.

Make sure the firewall instance you apply to an interface is already defined, or you may experience unintended results. If you apply a firewall instance that does not exist to an interface, the implicit firewall rule of **allow all** will be applied.

Use the **set** form of this command to apply a firewall instance to an OpenVPN interface.

Use the **delete** form of this command to remove a firewall instance from an OpenVPN interface.

Use the **show** form of this command to view a firewall configuration for an OpenVPN interface.

interfaces serial <wanx> cisco-hdlc vif 1 firewall

Applies a firewall instance to a Cisco HDLC–encapsulated serial interface.

Syntax

set interfaces serial *wanx* **cisco-hdlc vif 1 firewall** {**in name** *fw-name* | **local name** *fw-name* | **out name** *fw-name*}

delete interfaces serial *wanx* **cisco-hdlc vif 1 firewall** [**in** | **local** | **out**]

show interfaces serial *wanx* **cisco-hdlc vif 1 firewall** [**in** | **local** | **out**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   serial wan0..wan23 {
      cisco-hdlc {
         vif 1 {
            firewall {
               in {
                  name text
               local {
                  name text
               out {
                  name text
               }
            }
         }
      }
   }
}
```

Parameters

wanx	The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined.
1	The identifier for the vif you are configuring. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.

in name *fw-name*	Applies the specified firewall instance to inbound traffic on the specified interface.
local name *fw-name*	Applies the specified firewall instance to traffic arriving on the specified interface and bound for the local system.
out name *fw-name*	Applies the specified firewall instance to outbound traffic on the specified interface.

Default

None.

Usage Guidelines

Use this command to apply a firewall instance, or rule set, to the vif of a Cisco HDLC–encapsulated serial interface.

A firewall has no effect on traffic traversing the system or destined to the system until a firewall rule set has been applied to an interface or a vif using this command.

To use the firewall feature, you define a firewall rule set as a named firewall instance, using the **firewall** command (see page 18). You then apply the firewall instance to interfaces and/or vifs using a statement like this one. Once applied, the instance acts as a packet filter.

The firewall instance will filter packets in one of the following ways, depending on what you specify when you apply it:

- **in.** If you apply the rule set as **in**, the firewall will filter packets entering the interface.
- **out.** If you apply the rule set as **out**, the firewall will filter packets leaving the interface.
- **local.** If you apply the rule set as **local**, the firewall will filter packets destined for the system itself.

For each interface, you can apply up to three firewall instances: one firewall **in** instance, one firewall **out** instance, and one firewall **local** instance.

Make sure the firewall instance you apply to an interface is already defined, or you may experience unintended results. If you apply a firewall instance that does not exist to an interface, the implicit firewall rule of **allow all** will be applied.

Use the **set** form of this command to apply a firewall instance to the vif of a Cisco HDLC–encapsulated serial interface.

Use the **delete** form of this command to remove a firewall instance from the vif of a Cisco HDLC–encapsulated serial interface.

Use the **show** form of this command to view a firewall instance on the vif of a Cisco HDLC–encapsulated serial interface.

interfaces serial <wanx> frame-relay vif <dlci> firewall

Applies a firewall instance to a Frame Relay–encapsulated serial interface.

Syntax

set interfaces serial *wanx* **frame-relay vif** *dlci* **firewall** {**in name** *fw-name* | **local name** *fw-name* | **out name** *fw-name*}

delete interfaces serial *wanx* **frame-relay vif** *dlci* **firewall** [**in** | **local** | **out**]

show interfaces serial *wanx* **frame-relay vif** *dlci* **firewall** [**in** | **local** | **out**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   serial wan0..wan23 {
      frame-relay {
         vif 16–991 {
            firewall {
               in {
                  name text
               local {
                  name text
               out {
                  name text
               }
            }
         }
      }
   }
}
```

Parameters

wanx	The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined.
dlci	The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. The range is 16 to 991. The vif must already have been defined.

in name *fw-name*	Applies the specified firewall instance to inbound traffic on the specified interface.
local name *fw-name*	Applies the specified firewall instance to traffic arriving on the specified interface and bound for the local system.
out name *fw-name*	Applies the specified firewall instance to outbound traffic on the specified interface.

Default

None.

Usage Guidelines

Use this command to apply a firewall instance, or rule set, to the vif of a Frame Relay–encapsulated serial interface.

A firewall has no effect on traffic traversing the system or destined to the system until a firewall rule set has been applied to an interface or a vif using this command.

To use the firewall feature, you define a firewall rule set as a named firewall instance, using the **firewall** command (see page 18). You then apply the firewall instance to interfaces and/or vifs using a statement like this one. Once applied, the instance acts as a packet filter.

The firewall instance will filter packets in one of the following ways, depending on what you specify when you apply it:

- **in.** If you apply the rule set as **in**, the firewall will filter packets entering the interface.
- **out.** If you apply the rule set as **out**, the firewall will filter packets leaving the interface.
- **local.** If you apply the rule set as **local**, the firewall will filter packets destined for the system itself.

For each interface, you can apply up to three firewall instances: one firewall **in** instance, one firewall **out** instance, and one firewall **local** instance.

Make sure the firewall instance you apply to an interface is already defined, or you may experience unintended results. If you apply a firewall instance that does not exist to an interface, the implicit firewall rule of **allow all** will be applied.

Use the **set** form of this command to apply a firewall instance to the vif of a Frame Relay–encapsulated serial interface.

Use the **delete** form of this command to remove a firewall instance from the vif of a Frame Relay–encapsulated serial interface.

Use the **show** form of this command to view a firewall instance on the vif of a Frame Relay–encapsulated serial interface.

interfaces serial <wanx> ppp vif 1 firewall

Applies a firewall instance to a PPP-encapsulated serial interface.

Syntax

set interfaces serial *wanx* **ppp vif 1 firewall** {**in name** *fw-name* | **local name** *fw-name* | **out name** *fw-name*}

delete interfaces serial *wanx* **ppp vif 1 firewall** [**in** | **local** | **out**]

show interfaces serial *wanx* **ppp vif 1 firewall** [**in** | **local** | **out**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    serial wan0..wan23 {
        ppp {
            vif 1 {
                firewall {
                    in {
                        name text
                    local {
                        name text
                    out {
                        name text
                    }
                }
            }
        }
    }
}
```

Parameters

wanx	The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined.
1	The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.

in name *fw-name*	Applies the specified firewall instance to inbound traffic on the specified interface.
local name *fw-name*	Applies the specified firewall instance to traffic arriving on the specified interface and bound for the local system.
out name *fw-name*	Applies the specified firewall instance to outbound traffic on the specified interface.

Default

None.

Usage Guidelines

Use this command to apply a firewall instance, or rule set, to the vif of a Point-to-Point Protocol (PPP)–encapsulated serial interface.

A firewall has no effect on traffic traversing the system or destined to the system until a firewall rule set has been applied to an interface or a vif using this command.

To use the firewall feature, you define a firewall rule set as a named firewall instance, using the **firewall** command (see page 18). You then apply the firewall instance to interfaces and/or vifs using a statement like this one. Once applied, the instance acts as a packet filter.

The firewall instance will filter packets in one of the following ways, depending on what you specify when you apply it:

- **in.** If you apply the rule set as **in**, the firewall will filter packets entering the interface.
- **out.** If you apply the rule set as **out**, the firewall will filter packets leaving the interface.
- **local.** If you apply the rule set as **local**, the firewall will filter packets destined for the system itself.

For each interface, you can apply up to three firewall instances: one firewall **in** instance, one firewall **out** instance, and one firewall **local** instance.

Make sure the firewall instance you apply to an interface is already defined, or you may experience unintended results. If you apply a firewall instance that does not exist to an interface, the implicit firewall rule of **allow all** will be applied.

Use the **set** form of this command to apply a firewall instance to the vif of a PPP–encapsulated serial interface.

Use the **delete** form of this command to remove a firewall instance from the vif of a PPP–encapsulated serial interface.

Use the **show** form of this command to view a firewall instance on the vif of a PPP–encapsulated serial interface.

interfaces tunnel <tunx> firewall

Applies named firewall instances (packet-filtering rule sets) to a tunnel interface.

Syntax

set interfaces tunnel *tunx* **firewall** {**in name** *fw-name* | **local name** *fw-name* | **out name** *fw-name*}

delete interfaces tunnel *tunx* **firewall** [**in** | **local** | **out**]

show interfaces tunnel *tunx* **firewall** [**in** | **local** | **out**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   tunnel tun0..tun23 {
      firewall {
         in {
            name text
         local {
            name text
         out {
            name text
         }
      }
   }
}
```

Parameters

tunx	Mandatory. Multi-node. An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**. You can define multiple tunnel interfaces by creating multiple **tunnel** configuration nodes.
in name *fw-name*	Applies the specified firewall instance to inbound traffic on the specified interface.
local name *fw-name*	Applies the specified firewall instance to traffic arriving on the specified interface and bound for the local system.

out name *fw-name*	Applies the specified firewall instance to outbound traffic on the specified interface.

Default

None.

Usage Guidelines

Use this command to apply a firewall instance, or rule set, to the vif of a Point-to-Point Protocol (PPP)–encapsulated serial interface.

A firewall has no effect on traffic traversing the system or destined to the system until a firewall rule set has been applied to an interface or a vif using this command.

To use the firewall feature, you define a firewall rule set as a named firewall instance, using the **firewall** command (see page 18). You then apply the firewall instance to interfaces and/or vifs using a statement like this one. Once applied, the instance acts as a packet filter.

The firewall instance will filter packets in one of the following ways, depending on what you specify when you apply it:

- **in.** If you apply the rule set as **in**, the firewall will filter packets entering the interface.
- **out.** If you apply the rule set as **out**, the firewall will filter packets leaving the interface.
- **local.** If you apply the rule set as **local**, the firewall will filter packets destined for the system itself.

For each interface, you can apply up to three firewall instances: one firewall **in** instance, one firewall **out** instance, and one firewall **local** instance.

Make sure the firewall instance you apply to an interface is already defined, or you may experience unintended results. If you apply a firewall instance that does not exist to an interface, the implicit firewall rule of **allow all** will be applied.

Use the **set** form of this command to apply a firewall instance to a tunnel interface.

Use the **delete** form of this command to remove a firewall instance from a tunnel interface.

Use the **show** form of this command to view a firewall instance on a tunnel interface.

interfaces wirelessmodem <wlmx> firewall

Applies named firewall instances (packet-filtering rule sets) to a wirelessmodem interface.

Syntax

set interfaces wirelessmodem *wlmx* **firewall** {**in name** *fw-name* | **local name** *fw-name* | **out name** *fw-name*}

delete interfaces wirelessmodem *wlmx* **firewall** [**in** | **local** | **out**]

show interfaces wirelessmodem *wlmx* **firewall** [**in** | **local** | **out**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   wirelessmodem wlm0..wlm999 {
      firewall {
         in {
            name text
         local {
            name text
         out {
            name text
         }
      }
   }
}
```

Parameters

wlmx	Mandatory. Multi-node. The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.
in name *fw-name*	Applies the specified firewall instance to inbound traffic on the specified interface.
local name *fw-name*	Applies the specified firewall instance to traffic arriving on the specified interface and bound for the local system.
out name *fw-name*	Applies the specified firewall instance to outbound traffic on the specified interface.

Default

None.

Usage Guidelines

Use this command to apply a firewall instance, or rule set, to a wirelessmodem interface.

A firewall has no effect on traffic traversing the system or destined to the system until a firewall rule set has been applied to an interface using this command.

To use the firewall feature, you define a firewall rule set as a named firewall instance, using the **firewall** command (see page 18). You then apply the firewall instance to interfaces using a statement like this one. Once applied, the instance acts as a packet filter.

The firewall instance will filter packets in one of the following ways, depending on what you specify when you apply it:

- **in.** If you apply the rule set as **in**, the firewall will filter packets entering the interface.
- **out.** If you apply the rule set as **out**, the firewall will filter packets leaving the interface.
- **local.** If you apply the rule set as **local**, the firewall will filter packets destined for the system itself.

For each interface, you can apply up to three firewall instances: one firewall **in** instance, one firewall **out** instance, and one firewall **local** instance.

Make sure the firewall instance you apply to an interface is already defined, or you may experience unintended results. If you apply a firewall instance that does not exist to an interface, the implicit firewall rule of **allow all** will be applied.

Use the **set** form of this command to apply a firewall instance to a wirelessmodem interface.

Use the **delete** form of this command to remove a firewall instance from a wirelessmodem interface.

Use the **show** form of this command to view a firewall instance on a wirelessmodem interface.

show firewall

Displays rules associated with a firewall instance.

Syntax

show firewall [*name* [**rule** *rule-num* |**detail** [**rule** *rule-num*] |

Command Mode

Operational mode.

Parameters

name	Optional. Displays information for the specified firewall instance.
rule *rule-num*	Optional. Displays the specified firewall rule.
detail	Optional. Displays detailed information for the specified firewall instance.
rule *rule-num*	Optional. Displays detailed information for the specified firewall rule.

Default

By default all rules for all firewall instances are displayed.

Usage Guidelines

Use this command to show information with a firewall instance or firewall rule.

Examples

Example 1-10 shows the rules associated with a firewall instance on R1.

Example 1-10 “show firewall TEST”: Displaying a firewall instance

```
vyatta@R1:~$ show firewall TEST
State Codes: E - Established, I - Invalid, N - New, R - Related

rule  action  source            destination        proto  state
----  ------  ------            -----------        -----  -----
10    ACCEPT  192.168.0.0/24      0.0.0.0/0                 any
1025  DROP    0.0.0.0/0           0.0.0.0/0          all    any

vyatta@R1:~$
```

Example 1-11 shows rule 10 from firewall instance TEST on R1.

Example 1-11 “show firewall TEST detail rule 10”: Displaying rule information

```
vyatta@R1:~$ show firewall TEST detail rule 10

Rule: 10
Packets: 0          Bytes: 0
Action: ACCEPT
Protocol:
State: any
Source
  Address: 192.168.0.0/24
  Ports: all
Destination
  Address: 0.0.0.0/0
  Ports: all
ICMP Code: -
ICMP Type: -
Logging:
------------------------

vyatta@R1:~$
```

show firewall <name> statistics

Displays statistics information for a firewall instance.

Syntax

show firewall *name* **statistics**

Command Mode

Operational mode.

Configuration Statement

None.

Parameters

name	Mandatory. The name of a specific firewall instance.

Default

None.

Usage Guidelines

Use this command to display statistics information for the specified firewall instance.

Examples

Example 1-12 shows the statistics for firewall instance TEST on R1.

Example 1-12 “show firewall TEST statistics”: Displaying the statistics associated with the TEST instance.

```
vyatta@R1:~$ show firewall TEST statistics

rule  packets   bytes action  source              destination
----  -------   ----- ------  ------              -----------
10    245       14232 ACCEPT  192.168.0.0/24      0.0.0.0/0
1025  0         0     DROP    0.0.0.0/0           0.0.0.0/0
```

```
vyatta@R1:~$
```

Chapter 2: Intrusion Protection System

This chapter lists the commands for setting up intrustion detection and prevention, and traffic filtering on the Vyatta system.

This chapter presents the following topics:

- IPS Commands

IPS Commands

This chapter contains the following commands.

Configuration Commands	
content-inspection ips actions priority-1 <action>	Specifies the action to take for packets matching priority 1 IPS rules.
content-inspection ips actions priority-2 <action>	Specifies the action to take for packets matching priority 2 IPS rules.
content-inspection ips actions priority-3 <action>	Specifies the action to take for packets matching priority 3 IPS rules.
content-inspection ips actions other <action>	Specifies what to do with packets matching IPS rules with priority other than 1, 2, or 3.
content-inspection ips auto-update oink-code <code>	Records a Snort “oink code” for automatic Snort rule base updates.
content-inspection ips auto-update update-hour <hour>	Specifies the hour of the day for daily Snort rule base updates.
Operational Commands	
show ips log	Displays alerts logged by the IPS.
show ips summary	Displays a summary of all IPS alerts.
show ips update-log	Displays the history of automatic IPS rules updates.

content-inspection ips actions priority-1 <action>

Specifies the action to take for packets matching priority 1 IPS rules.

Syntax

set content-inspection ips actions priority-1 *action*

delete content-inspection ips actions priority-1

show content-inspection ips actions priority-1

Command Mode

Configuration mode.

Configuration Statement

```
content-inspection{
   ips {
      actions {
         priority-1 [alert|drop|pass|sdrop]
      }
   }
}
```

Parameters

action	The action to take when a packet matches a priority 1 rule. Supported values are as follows: **alert**: Allows the packet and log an alert. **drop**: Drops the packet and log an alert. **pass**: Allows the packet and take no further action. **sdrop**: Drops packet but does not log an alert (that is, drops the packet silently).

Default

The default action is **drop**.

Usage Guidelines

Use this command to specify the action to take for packets matching priority 1 Intrusion Protection System (IPS) rules.

Use the **set** form of this command to specify the action.

Use the **delete** form of this command to restore the default action.

Use the **show** form of this command to display IPS priority 1 action configuration.

content-inspection ips actions priority-2 <action>

Specifies the action to take for packets matching priority 2 IPS rules.

Syntax

set content-inspection ips actions priority-2 *action*

delete content-inspection ips actions priority-2

show content-inspection ips actions priority-2

Command Mode

Configuration mode.

Configuration Statement

```
content-inspection{
   ips {
      actions {
         priority-2 [alert|drop|pass|sdrop]
      }
   }
}
```

Parameters

action	The action to take when a packet matches a priority 2 rule. Supported values are as follows: **alert**: Allows the packet and log an alert. **drop**: Drops the packet and log an alert. **pass**: Allows the packet and take no further action. **sdrop**: Drops packet but does not log an alert (that is, drops the packet silently).

Default

The default action is **alert**.

Usage Guidelines

Use this command to specify the action to take for packets matching priority 2 Intrusion Protection System (IPS) rules.

Use the **set** form of this command to specify the action.

Use the **delete** form of this command to restore the default action.

Use the **show** form of this command to display IPS priority 2 action configuration.

content-inspection ips actions priority-3 <action>

Specifies the action to take for packets matching priority 3 IPS rules.

Syntax

set content-inspection ips actions priority-3 *action*

delete content-inspection ips actions priority-3

show content-inspection ips actions priority-3

Command Mode

Configuration mode.

Configuration Statement

```
content-inspection{
   ips {
      actions {
         priority-3 [alert|drop|pass|sdrop]
      }
   }
}
```

Parameters

action The action to take when a packet matches a priority 3 rule. Supported values are as follows:

alert: Allows the packet and log an alert.

drop: Drops the packet and log an alert.

pass: Allows the packet and take no further action.

sdrop: Drops packet but does not log an alert (that is, drops the packet silently).

Default

The default action is **alert**.

Usage Guidelines

Use this command to specify the action to take for packets matching priority 3 Intrusion Protection System (IPS) rules.

Use the **set** form of this command to specify the action.

Use the **delete** form of this command to restore the default action.

Use the **show** form of this command to display IPS priority 3 action configuration.

content-inspection ips actions other <action>

Specifies what to do with packets matching IPS rules with priority other than 1, 2, or 3.

Syntax

set content-inspection ips actions other *action*

delete content-inspection ips actions other

show content-inspection ips actions other

Command Mode

Configuration mode.

Configuration Statement

```
content-inspection{
   ips {
      actions {
         other [alert|drop|pass|sdrop]
      }
   }
}
```

Parameters

action	The action to take when a packet matches a rule other than those having a priority of 1, 2, or 3. Supported values are as follows: **alert**: Allows the packet and log an alert. **drop**: Drops the packet and log an alert. **pass**: Allows the packet and take no further action. **sdrop**: Drops packet but does not log an alert (that is, drops the packet silently).

Default

The default action is **pass**.

Usage Guidelines

Use this command to specify what to do with packets matching Intrusion Protection System (IPS) rules other than rules with priority 1, 2, or 3.

Use the **set** form of this command to specify the action.

Use the **delete** form of this command to restore the default action.

Use the **show** form of this command to display IPS rule action configuration.

content-inspection ips auto-update oink-code <code>

Records a Snort “oink code” for automatic Snort rule base updates.

Syntax

set content-inspection ips auto-update oink-code *code*

delete content-inspection ips auto-update oink-code

show content-inspection ips auto-update oink-code

Command Mode

Configuration mode.

Configuration Statement

```
content-inspection{
   ips {
      auto-update {
         oink-code text
      }
   }
}
```

Parameters

code	Mandatory if updates are to be received. The “oink” code generated at www.snort.org. This code is required in order to receive automatic IPS rule base updates from snort.org.

Default

None.

Usage Guidelines

Use this command to specify the “oink code” for downloading Snort rule updates.

The Vyatta system uses the Snort (www.snort.org) engine for intrusion detection. The Snort rule base can be automatically downloaded; however, in order to access Snort rule updates, you must register with the Snort organization and generate an “oink” code, which is used to authenticate the system.

Specify your oink code using this command. The Vyatta system uses this code when seeking rule base updates from the Snort organization.

A successful rule base update requires a restart of the Snort daemon. This restart can take five to ten seconds during which time the IPS will not be in effect.

Use the **set** form of this command to specify your Snort oink code.

Use the **delete** form of this command to remove Snort oink code configuration.

Use the **show** form of this command to display the configured Snort oink code.

content-inspection ips auto-update update-hour <hour>

Specifies the hour of the day for daily Snort rule base updates.

Syntax

set content-inspection ips auto-update update-hour *hour*

delete content-inspection ips auto-update update-hour

show content-inspection ips auto-update update-hour

Command Mode

Configuration mode.

Configuration Statement

```
content-inspection{
   ips {
      auto-update {
         update-hour u32
      }
   }
}
```

Parameters

hour	Mandatory if updates are to be received. The hour of the day at which to update the Snort rule base. The time is based on a 24-hour clock.

Default

None.

Usage Guidelines

Use this command to specify the hour of the day for Snort rule base updates.

A successful rule base update requires a restart of the Snort daemon. This restart can take five to ten seconds during which time the IPS will not be in effect.

Use the **set** form of this command to specify the hour of the day for rules updates.

Use the **delete** form of this command to remove the configuration.

Use the **show** form of this command to display the configuration.

show ips log

Displays alerts logged by the IPS.

Syntax

show ips log

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to see alerts logged by the Vyatta Intrusion Protection System (IPS).

Examples

Example 2-1 shows the first screen of output for **show ips log**.

Example 2-1 “show ips log”: Displaying ips events

```
vyatta@R1:~$ show ips log
================================================
IPS events logged since Fri Apr 18 23:08:33 2008
================================================
2008-04-19 01:04:36.972690 {ICMP} 76.75.95.195 -> 76.74.103.8
(misc-activity) Misc activity (priority 3)
[1:483:5] ICMP PING CyberKit 2.2 Windows
---------------------------------------------------------------
------------
2008-04-19 01:04:38.410018 {ICMP} 76.75.95.195 -> 76.74.103.64
(misc-activity) Misc activity (priority 3)
[1:483:5] ICMP PING CyberKit 2.2 Windows
---------------------------------------------------------------
------------
2008-04-19 01:04:38.410091 {ICMP} 76.75.95.195 -> 76.74.103.65
(misc-activity) Misc activity (priority 3)
```

```
[1:483:5] ICMP PING CyberKit 2.2 Windows
---------------------------------------------------------------
------------
2008-04-19 01:04:38.413503 {ICMP} 76.75.95.195 -> 76.74.103.66
(misc-activity) Misc activity (priority 3)
[1:483:5] ICMP PING CyberKit 2.2 Windows
---------------------------------------------------------------
------------
2008-04-19 01:04:38.417576 {ICMP} 76.75.95.195 -> 76.74.103.67
(misc-activity) Misc activity (priority 3)
[1:483:5] ICMP PING CyberKit 2.2 Windows
---------------------------------------------------------------
------------
```

show ips summary

Displays a summary of all IPS alerts.

Syntax

show ips summary

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to see a summary of all Intrusion Protection System (IPS) alerts.

Examples

Example 2-2 shows the output for **show ips summary**.

Example 2-2 “show ips summary”: Displaying a summary of IPS alerts

```
vyatta@R1:~$ show ips summary
Processing log files...
Done.

===========================================================
Summary of IPS events logged since Fri Apr 18 23:08:33 2008
===========================================================
  Total number of events: 22331

  Breakdown by priorities:
    Priority 2: 17120
    Priority 3: 5211

  Breakdown by classes:
    bad-unknown: 9983 (Potentially Bad Traffic)
    attempted-recon: 95 (Attempted Information Leak)
    misc-activity: 5211 (Misc activity)
    misc-attack: 7042 (Misc Attack)

  Breakdown by signatures:
    [1:469:3]: 93 (ICMP PING NMAP)
    [1:476:4]: 2 (ICMP webtrends scanner)
    [1:483:5]: 5189 (ICMP PING CyberKit 2.2 Windows)
    [1:486:4]: 10 (ICMP Destination Unreachable Communication
with Destination Host is Administratively Prohibited)
    [1:524:8]: 12 (BAD-TRAFFIC tcp port 0 traffic)
    [1:527:8]: 9983 (DELETED BAD-TRAFFIC same SRC/DST)
    [1:2003:8]: 3521 (MS-SQL Worm propagation attempt)
    [1:2004:7]: 3521 (MS-SQL Worm propagation attempt OUTBOUND)

  Breakdown by dates:
    2008-04-19: 510
    2008-04-20: 1132
    2008-04-21: 1101
    2008-04-22: 2363
    2008-04-23: 2788
    2008-04-24: 1200
    2008-04-25: 1119
    2008-04-26: 7190
    2008-04-27: 2653
    2008-04-28: 1219
    2008-04-29: 1056

vyatta@R1:~$
```

show ips update-log

Displays the history of automatic IPS rules updates.

Syntax

show ips update-log

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to see a history of automatic Intrusion Protection System (IPS) rules updates.

Examples

Example 2-3 shows the output for **show ips update-log**.

Example 2-3 “show ips update-log”: Displaying ips rules update history

```
vyatta@R1:~$ show ips update-log
2008-06-18-015801: Failed to get
http://www.snort.org/pub-bin/oinkmaster.cgi/foo/snortrules-snap
shot-2.7.tar.gz
2008-06-18-015801: Update aborted due to error. IPS rules not
updated.
vyatta@R1:~$
```

Chapter 3: Traffic Filtering

This chapter lists the commands for setting up traffic filtering on the Vyatta system.

This chapter presents the following topics:

- Traffic Filtering Commands

Traffic Filtering Commands

This chapter contains the following commands.

Configuration Commands	
content-inspection traffic-filter <filter>	Specifies which traffic is to be processed by Vyatta IPS functions.
Operational Commands	
None	

content-inspection traffic-filter <filter>

Specifies which traffic is to be processed by Vyatta IPS functions.

Syntax

set content-inspection traffic-filter {**preset all** | **custom** *rule*}

delete content-inspection traffic-filter

show content-inspection traffic-filter

Command Mode

Configuration mode.

Configuration Statement

```
content-inspection{
   traffic-filter {
      preset all
      custom text
   }
}
```

Parameters

preset all	All traffic is processed by the IPS.
custom *rule*	Specifies the name of a firewall rule set defining the type of traffic to be processed by the IPS.

Default

All traffic is processed when IPS is enabled.

Usage Guidelines

Use this command to specify the kind of traffic to be processed by Intrusion Protection System (IPS) functions.

Even if the traffic filter is specified, traffic is processed by the IPS only when the **ips** configuration node is defined.

Use the **set** form of this command to designate traffic for IPS processing.

Use the **delete** form of this command to restore default traffic filtering.

Use the **show** form of this command to display traffic filter configuration.

Chapter 4: URL Filtering

This chapter explains how to set up URL filtering on the Vyatta system.

This chapter presents the following topics:

- URL Filtering Configuration
- URL Filtering Commands

URL Filtering Configuration

This section presents the following topics:

- URL Filtering Overview
- URL Filtering Configuration Examples

URL Filtering Overview

The Vyatta system can be configured to act as a web proxy server for web caching and URL filtering. A client can request a web page from the Vyatta system, which connects to the web server and requests the page on the client's behalf. The Vyatta system caches the response; if the page is requested again it can be served directly from the cache, saving the time and bandwidth required for transacting with the web server.

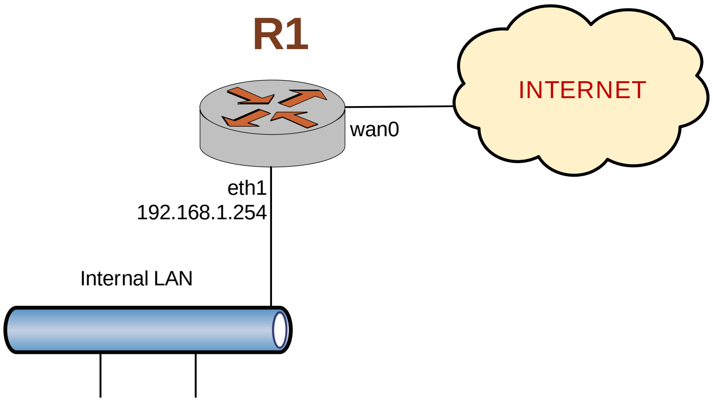
When acting as a web proxy, the Vyatta system can also provide URL filtering. Access to URLs can be denied by specifying them on a "blacklist."

URL Filtering Configuration Examples

Figure 4-1 shows the web proxy deployment used in the examples in this section. In this scenario:

- Devices on the company's internal LAN are accessing the Internet through the Vyatta system (R1).
- The web proxy is deployed on R1 to provide caching and URL filtering functionality to employees accessing the Internet.

Figure 4-1 Web proxy



This section presents the following examples:

- Example 4-1 Blocking specific web sites
- Example 4-2 Logging URL filtering
- Example 4-3 Filtering on blacklist categories
- Example 4-4 Allowing access to a specific site

Blocking Specific URLs

Example 4-1 blocks specific URLs by explicitly specifying them using the **local-block** option, rather than by downloading and setting up a blacklist. To block specific URLs on the Vyatta system, perform the following steps:

Example 4-1 Blocking specific web sites

Step	Command
Set the address to listen for requests on.	vyatta@R1# **set service webproxy listen-address 192.168.1.254** [edit]
Deny requests for the YouTube web site.	vyatta@R1# **set service webproxy url-filtering squidguard local-block youtube.com** [edit]
Deny requests for the Facebook web site.	vyatta@R1# **set service webproxy url-filtering squidguard local-block facebook.com** [edit]
Commit the change	vyatta@R1# **commit** [edit]

Show the updated web proxy–related configuration.	vyatta@R1# **show service webproxy**
	listen-address 192.168.1.254 {
	}
	url-filtering {
	    squidguard {
	        local-block youtube.com
	        local-block facebook.com
	    }
	}
	[edit]

Verifying Filtering

You can verify that filtering is working for the previous example by enabling logging for the **local-block** category (“**log all”** would also work.). To view the results, use the **show webproxy blacklist log** command.

Example 4-2 enables logging for locally blocked URLs. To log web proxy functions in this way, perform the following steps:

Example 4-2 Logging URL filtering

Step	Command
Set the web proxy to log everything filtered by the “local-block” option.	vyatta@R1# **set service webproxy url-filtering squidguard log local-block** [edit]
Commit the change	vyatta@R1# **commit** [edit]

Show the updated web proxy–related configuration.	vyatta@R1# **show service webproxy**
	listen-address 192.168.1.254 {
	}
	url-filtering {
	    squidguard {
	        local-block youtube.com
	        local-block facebook.com
	        log local-block
	    }
	}
	[edit]

Filtering by Content Category

Example 4-3 uses a downloaded squidGuard database (downloaded using **update webproxy blacklists**) to filter web contents by content category. In this example, web content is filtered for URLs related to advertisements, spyware, and gambling. To configure the web proxy in this way, perform the following steps:

Example 4-3 Filtering on blacklist categories

Step	Command
Block the ads category	vyatta@R1# **set service webproxy url-filtering squidguard block-category ads** [edit]

Example 4-3 Filtering on blacklist categories

Block the spyware category	vyatta@R1# **set service webproxy url-filtering squidguard block-category spyware** [edit]
Block the gambling category	vyatta@R1# **set service webproxy url-filtering squidguard block-category gambling** [edit]
Commit the change	vyatta@R1# **commit** [edit]

Show the updated web proxy–related configuration.	vyatta@R1# **show service webproxy**
	listen-address 192.168.1.254 {
	}
	url-filtering {
	    squidguard {
	        block-category ads
	        block-category spyware
	        block-category gambling
	        local-block youtube.com
	        local-block facebook.com
	        log local-block
	    }
	}
	[edit]

Allowing Specific Sites

Example 4-3 enables sites that are blocked in virtue of being within a blocked category to be specifically allowed. In this example, the URL www.company-ads.com is specifically allowed, even though it falls within the blocked category of advertisements. To allow specific URLs, perform the following steps:

Example 4-4 Allowing access to a specific site

Step	Command
Allow users to access www.company-ads.com	vyatta@R1# **set service webproxy url-filtering squidguard local-ok www.company-ads.com** [edit]
Commit the change	vyatta@R1# **commit** [edit]

Example 4-4 Allowing access to a specific site

Show the updated web proxy–related configuration.

```
vyatta@R1# show service webproxy
listen-address 192.168.1.254 {
}
url-filtering {
   squidguard {
      block-category ads
      block-category spyware
      block-category gambling
      local-block youtube.com
      local-block facebook.com
      local-ok www.foobar.com
      log local-block
}
[edit]
```

URL Filtering Commands

This chapter contains the following commands.

Configuration Commands	
service webproxy url-filtering squidguard	Blocks URLs in all categories.
service webproxy url-filtering squidguard auto-update <interval>	Sets the interval at which to update squidGuard databases.
service webproxy url-filtering squidguard block-category <category>	Blocks web content by squidGuard database category.
service webproxy url-filtering squidguard local-block <address>	Defines a specific IP address or URL to be blocked.
service webproxy url-filtering squidguard local-ok <address>	Specifies an IP address or URL to allow.
service webproxy url-filtering squidguard log <category>	Enables logging for a squidGuard database category.
service webproxy url-filtering squidguard redirect-url <url>	Specifies a URL to redirect users to when a blacklisted URL is requested.
Operational Commands	
show webproxy blacklist categories	Displays all categories defined in the installed squidGuard database.
show webproxy blacklist domains	Displays all domains listed in the installed database.
show webproxy blacklist log	Displays the log for blacklisted URLs.
show webproxy blacklist search <filter>	Displays domains and/or URLs matching search text.
show webproxy blacklist urls	Displays all URLs in squidGuard database categories.
show webproxy log	Displays the web proxy log.
update webproxy blacklists	Updates the squidGuard database.

service webproxy url-filtering squidguard

Blocks URLs in all categories.

Syntax

set service webproxy url-filtering squidguard

delete service webproxy url-filtering squidguard

show service webproxy url-filtering squidguard

Command Mode

Configuration mode.

Configuration Statement

```
service {
   webproxy {
      url-filtering {
         squidguard {}
      }
   }
}
```

Parameters

None.

Default

None.

Usage Guidelines

Use this command with no additional configuration nodes to block URLs in all squidGuard categories. Specifying additional nodes in the configuration tree under **squidguard** refines the URLs to be blocked.

Use the **set** form of this command to apply URL filtering.

Use the **delete** form of this command to remove URL filtering.

Use the **show** form of this command to view URL filtering configuration.

service webproxy url-filtering squidguard auto-update <interval>

Sets the interval at which to update squidGuard databases.

Syntax

set service webproxy url-filtering squidguard auto-update *interval*

delete service webproxy url-filtering squidguard auto-update

show service webproxy url-filtering squidguard auto-update

Command Mode

Configuration mode.

Configuration Statement

```
service {
   webproxy {
      url-filtering {
         squidguard {
            auto-update u32
         }
      }
   }
}
```

Parameters

interval	The interval, in days, at which the web proxy service will check for database updates.

Default

None.

Usage Guidelines

Use this command to specify the interval at which the system should check for database updates.

Use the **set** form of this command to set the update interval.

Use the **delete** form of this command to restore the default database update interval.

Use the **show** form of this command to view database update interval configuration.

service webproxy url-filtering squidguard block-category <category>

Blocks web content by squidGuard database category.

Syntax

set service webproxy url-filtering squidguard block-category *category*

delete service webproxy url-filtering squidguard block-category *category*

show service webproxy url-filtering squidguard block-category

Command Mode

Configuration mode.

Configuration Statement

```
service {
   webproxy {
      url-filtering {
         squidguard {
            block-category text
         }
      }
   }
}
```

Parameters

category	Multi-node. The database category to block, or the keyword **all** to block all categories. You can block more than one category by creating multiple **block-category** configuration nodes.

Default

When the **squidguard** configuration node is defined with no block categories, all categories are blocked.

Usage Guidelines

Use this command to specify database categories to block.

The categories available will vary with the specific database. To view the categories defined in the installed database, issue the **show webproxy blacklist categories** command (see page 131).

Use the **set** form of this command to bock a database category.

Use the **delete** form of this command to stop a database category from being blocked.

Use the **show** form of this command to view the database categories blocking configuration.

service webproxy url-filtering squidguard local-block <address>

Defines a specific IP address or URL to be blocked.

Syntax

set service webproxy url-filtering squidguard local-block *address*

delete service webproxy url-filtering squidguard local-block *address*

show service webproxy url-filtering squidguard local-block

Command Mode

Configuration mode.

Configuration Statement

```
service {
   webproxy {
      url-filtering {
         squidguard {
            local-block text
         }
      }
   }
}
```

Parameters

address	Multi-node. An IP address or URL to be blocked. You can block a number of IP addresses and/or URLs by creating multiple **local-block** configuration nodes.

Default

None.

Usage Guidelines

Use this command to specify an IP address or URL to be blocked. This allows you to block sites not belonging to a database category.

Use the **set** form of this command to block a specific IP address or URL.

Use the **delete** form of this command to stop an IP address or URL from being blocked.

Use the **show** form of this command to view individual blocking configuration.

service webproxy url-filtering squidguard local-ok <address>

Specifies an IP address or URL to allow.

Syntax

set service webproxy url-filtering squidguard local-ok *address*

delete service webproxy url-filtering squidguard local-ok *address*

show service webproxy url-filtering squidguard local-ok

Command Mode

Configuration mode.

Configuration Statement

```
service {
   webproxy {
      url-filtering {
         squidguard {
            local-ok text
         }
      }
   }
}
```

Parameters

address	Multi-node. An IP address or URL to allow.

Default

None.

Usage Guidelines

Use this command to allow an IP address or URL that blocked because it belongs to a squidGuard database category.

Use the **set** form of this command to specify an IP address or URL to allow.

Use the **delete** form of this command to return an IP address or URL in a blocked category to being blocked.

Use the **show** form of this command to view IP addresses and URLs being specifically allowed.

service webproxy url-filtering squidguard log <category>

Enables logging for a squidGuard database category.

Syntax

set service webproxy url-filtering squidguard log *category*

delete service webproxy url-filtering squidguard log *category*

show service webproxy url-filtering squidguard log

Command Mode

Configuration mode.

Configuration Statement

```
service {
   webproxy {
      url-filtering {
         squidguard {
            log text
         }
      }
   }
}
```

Parameters

category	Multi-node. The squidGuard database category to log, or the keyword **all** to log all categories.

Default

Web proxy URL filtering is not logged.

Usage Guidelines

Use this command to direct the system to log filtering of squidGuard database categories.

Use the **set** form of this command to specify a database category to be logged.

Use the **delete** form of this command to stop the system from logging a database category.

Use the **show** form of this command to view database category logging configuration.

service webproxy url-filtering squidguard redirect-url <url>

Specifies a URL to redirect users to when a blacklisted URL is requested.

Syntax

set service webproxy url-filtering squidguard redirect-url *url*

delete service webproxy url-filtering squidguard redirect-url

show service webproxy url-filtering squidguard redirect-url

Command Mode

Configuration mode.

Configuration Statement

```
service {
   webproxy {
      url-filtering {
         squidguard {
            redirect-url text
         }
      }
   }
}
```

Parameters

url	The URL to which to redirect users when the user attempts to access a blacklisted URL. By default, users are redirected to **http://www.google.com**.

Default

Users attempting to access a blacklisted site are redirected to **http://www.google.com**.

Usage Guidelines

Use this command to specify a redirect URL for users attempting to access a filtered URL.

Use the **set** form of this command to specify a redirect URL.

Use the **delete** form of this command to restore the default redirect URL.

Use the **show** form of this command to view redirect URL configuration.

show webproxy blacklist categories

Displays all categories defined in the installed squidGuard database.

Syntax

show webproxy blacklist categories

Command Mode

Operational mode.

Parameters

None

Usage Guidelines

Use this command to display all database categories that are available in the squidGuard database that is currently installed.

Examples

Example 4-5 displays categories for a squidGuard database.

Example 4-5 Displaying database categories

```
vyatta@R1> show webproxy blacklist categories
ads
aggressive
audio-video
drugs
gambling
hacking
mail
porn
proxy
redirector
spyware
suspect
violence
warez
vyatta@R1>
```

show webproxy blacklist domains

Displays all domains listed in the installed database.

Syntax

show webproxy blacklist domains

Command Mode

Operational mode.

Parameters

None

Usage Guidelines

Use this command to display all the domains in the installed squidGuard database. Domains from all database categories are shown.

Examples

Example 4-6 shows the first few domains displayed from an installed database.

Example 4-6 Displaying database domains

```
vyatta@R1> show webproxy blacklist domains
101com.com
101order.com
103bees.com
1100i.com
123banners.com
123found.com
123pagerank.com
180searchassistant.com
180solutions.com
207.net
247media.com
247realmedia.com
24pm-affiliation.com
:
:
```

show webproxy blacklist log

Displays the log for blacklisted URLs.

Syntax

show webproxy blacklist log

Command Mode

Operational mode.

Parameters

None

Usage Guidelines

Use this command to display the system’s record of URLs that have been filtered.

Examples

Example 4-7 shows sample output of **show webproxy blacklist log**.

Example 4-7 Displaying the blacklist log

```
vyatta@R1> show webproxy blacklist log
2008-09-03 18:12:01 [12027] Request(default/gambling/-)
http://www.goldenpalacepoker.com 10.1.0.173/- - GET
2008-09-04 10:00:44 [12988] Request(default/spyware/-)
http://www.180solutions.com 10.1.0.173/- - GET
vyatta@R1>
```

show webproxy blacklist search <filter>

Displays domains and/or URLs matching search text.

Syntax

show webproxy blacklist search *filter*

Command Mode

Operational mode.

Parameters

filter	The filter text.

Usage Guidelines

Use this command to search for domains or URLs within the installed squidGuard database. All domains or URLs matching the filter string are shown.

Examples

Example 4-8 lists the IP addresses in the installed database that begin with “206.132.42”.

Example 4-8 Searching for an IP address or URL in a database

```
vyatta@R1> show webproxy blacklist search 206.132.42
porn/domains     206.132.42.195
porn/domains     206.132.42.197
porn/domains     206.132.42.200
porn/domains     206.132.42.201
porn/domains     206.132.42.206
porn/domains     206.132.42.212
porn/domains     206.132.42.213
porn/domains     206.132.42.215
porn/domains     206.132.42.218
porn/domains     206.132.42.219
porn/domains     206.132.42.231
porn/domains     206.132.42.250
porn/domains     206.132.42.251
```

```
porn/domains     206.132.42.253
warez/domains     206.132.42.196
warez/domains     206.132.42.208
vyatta@R1>
```

show webproxy blacklist urls

Displays all URLs in squidGuard database categories.

Syntax

show webproxy blacklist urls

Command Mode

Operational mode.

Parameters

None.

Usage Guidelines

Use this command to display all the URLs in squidGuard database categories.

Examples

Example 4-9 shows the first few entries of sample output of **show webproxy blacklist urls**.

Example 4-9 Displaying blacklisted URLs

```
vyatta@R1> show webproxy blacklist urls
thisisarandomentrythatdoesnotexist.com/foo
thisisarandomentrythatdoesnotexist.com/foo
134.121.0.99/~dcarp
165.21.101.33/~mp3mania
194.134.35.11/mp3forever
194.134.35.12/mp3forever
194.134.35.17/mp3forever
194.145.63.33/bg-mp3
195.141.34.45/mp3millennium
195.141.34.45/mp3sweden
195.66.60.36/mhs00160
195.96.96.198/~brouns
205.188.134.217/h0tp00lman
209.202.218.12/mb/honzicek
:
:
```

show webproxy log

Displays the web proxy log.

Syntax

show webproxy log

Command Mode

Operational mode.

Parameters

None.

Usage Guidelines

Use this command to display the web proxy log.

Examples

Example 4-10 displays a portion of the web proxy log.

Example 4-10 Viewing the web proxy log

```
vyatta@R1> show webproxy log
1220642370.525    708 172.16.117.25 TCP_REFRESH_MODIFIED/200
17825 GET
http://newsrss.bbc.co.uk/rss/newsonline_world_edition/front_pag
e/rss.xml - DIRECT/212.58.226.29 text/xml
1220642699.568    830 172.16.117.25 TCP_MISS/200 46448 GET
http://sb.google.com/safebrowsing/update? -
DIRECT/209.85.133.136 text/html
1220644499.691   1274 172.16.117.25 TCP_MISS/200 53832 GET
http://sb.google.com/safebrowsing/update? -
DIRECT/209.85.133.93 text/html
1220645984.836     34 172.16.117.25 TCP_MISS/302 694 GET
http://en-us.fxfeeds.mozilla.com/en-US/firefox/headlines.xml -
DIRECT/63.245.209.121 text/html
1220645984.881     31 172.16.117.25 TCP_MISS/302 736 GET
http://fxfeeds.mozilla.com/firefox/headlines.xml -
DIRECT/63.245.209.121 text/html
:
:
```

update webproxy blacklists

Updates the squidGuard database.

Syntax

update webproxy blacklists

Command Mode

Operational mode.

Parameters

None.

Usage Guidelines

Use this command to initiated an update the squidGuard database. If no databases have been installed, the system allows you to download and install one.

Examples

Example 4-11 shows the system interaction for downloading a first squidGuard database.

Example 4-11 Downloading a squidGuard database

```
vyatta@R1> update webproxy blacklists
No url-filtering blacklist installed
Would you like to download a blacklist? [confirm][y]
--2008-09-10 01:32:15--
http://squidguard.mesd.k12.or.us/blacklists.tgz
Resolving squidguard.mesd.k12.or.us... 198.236.66.41
Connecting to squidguard.mesd.k12.or.us|198.236.66.41|:80...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 5459348 (5.2M) [application/x-gzip]
Saving to: `/tmp/blacklists.gz'

100%[==========================================================
===============================================================
=======================>] 5,459,348    408K/s   in 13s
```

```
2008-09-10 01:32:29 (407 KB/s) - `/tmp/blacklists.gz' saved
[5459348/5459348]

Uncompressing blacklist...
```

Appendix A: ICMP Types

This appendix lists the ICMP 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 and standard literal strings onto ICMP types.Table A-1 lists the ICMP types defined by the IANA.

Table A-1 ICMP types

ICMP Type	Literal
0	echo-reply
3	unreachable
4	source-quench
5	redirect
6	alternate-address
8	echo
9	router-advertisement
10	router-solicitation
11	time-exceeded
12	parameter-problem
13	timestamp-reply
14	timestamp-request
15	information-request
16	information-reply
17	mask-request
18	mask-reply
31	conversion-error
32	mobile-redirect
33	where-are-you
34	i-am-here
35	mobile-regist-request
36	mobile-regist-response
37	domainname-request

Table A-1 ICMP types

ICMP Type	Literal
38	domainname-response
39	skip
40	photuris

Glossary of Acronyms

ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
AS	autonomous system
ARP	Address Resolution Protocol
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DLCI	data-link connection identifier
DMI	desktop management interface
DMZ	demilitarized zone
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
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/Ouput
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
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
ISP	Internet Service Provider
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
MAC	medium access control
MIB	Management Information Base
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit

NAT	Network Address Translation
ND	Neighbor Discovery
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
PCI	peripheral component interconnect
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
Rx	receive
SNMP	Simple Network Management Protocol
SONET	Synchronous Optical Network
SSH	Secure Shell
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus

TCP	Transmission Control Protocol
ToS	Type of Service
Tx	transmit
UDP	User Datagram Protocol
vif	virtual interface
VLAN	virtual LAN
VPN	Virtual Private Network
VRRP	Virtual Router Redundancy Protocol
WAN	wide area network