

VYATTA, INC.

| Vyatta System

# Security

## REFERENCE GUIDE

Intrusion Prevention System  
Web Filtering



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

## **COPYRIGHT**

Copyright © 2005–2012 Vyatta, Inc. All rights reserved.

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](http://vyatta.com).

## **PROPRIETARY NOTICES**

Vyatta is a registered trademark of Vyatta, Inc.

VMware, VMware ESX, and VMware server are trademarks of VMware, Inc.

XenServer, and XenCenter are trademarks of Citrix Systems, Inc.

All other trademarks are the property of their respective owners.

RELEASE DATE: March 2012

DOCUMENT REVISION: R6.4 v01

RELEASED WITH: R6.4.0

PART NO. A0-0219-10-0015

# Contents

<b>Quick List of Commands</b> .....	<b>vii</b>
<b>List of Examples</b> .....	<b>x</b>
<b>Preface</b> .....	<b>xii</b>
Intended Audience .....	xiii
Organization of This Guide .....	xiii
Document Conventions .....	xiii
Vyatta Publications .....	xiv
<b>Chapter 1 Intrusion Prevention System</b> .....	<b>1</b>
IPS Configuration .....	2
IPS Overview .....	2
Vyatta PLUS Snort VRT Service .....	2
Vyatta Core IPS .....	3
Initial Setup .....	3
Initial Setup: Vyatta PLUS Snort VRT Service .....	3
Initial Setup: Vyatta Core IPS .....	4
Specifying Traffic to Inspect .....	4
Traffic Location and Direction .....	5
Traffic Type .....	6
Specifying Which IPS Rules to Apply .....	7
Apply a Policy .....	7
Include or Exclude Rules .....	7
Specifying How to Handle Malicious Traffic .....	8
Ongoing Monitoring and Maintenance .....	10
Planning Considerations .....	10
IPS Configuration Examples .....	10
Initial Setup Examples .....	11
Inspecting Specific Traffic Examples .....	13
Changing from Registration-based to Subscription-Based IPS .....	20
IPS Commands .....	22
content-inspection inspect-all enable .....	24
content-inspection ips actions priority-1 <action> .....	25
content-inspection ips actions priority-2 <action> .....	27
content-inspection ips actions priority-3 <action> .....	29
content-inspection ips actions other <action> .....	31

content-inspection ips auto-update oink-code <code> . . . . .	33
content-inspection ips auto-update snortvrt-subscription . . . . .	35
content-inspection ips auto-update update-hour <hour> . . . . .	37
content-inspection ips log local . . . . .	39
content-inspection ips log remote-db . . . . .	40
content-inspection ips log syslog . . . . .	42
content-inspection ips modify-rules disable-sid <rule> . . . . .	45
content-inspection ips modify-rules enable-sid <rule> . . . . .	47
content-inspection ips modify-rules exclude-category <category> . . . . .	49
content-inspection ips modify-rules include-category <category> . . . . .	51
content-inspection ips modify-rules internal-network <ipv4net>. . . . .	53
content-inspection ips policy . . . . .	55
content-inspection traffic-filter <filter>. . . . .	57
interfaces <interface> content-inspection <direction> enable . . . . .	59
show ips log . . . . .	64
show ips rules summary . . . . .	66
show ips summary . . . . .	69
show ips update-log. . . . .	71
update ips rules . . . . .	72
zone-policy zone <to-zone> from <from-zone> content-inspection enable. . . . .	74
<b>Chapter 2 Web Filtering . . . . .</b>	<b>76</b>
Web Filtering Configuration . . . . .	77
Introduction . . . . .	77
Web Filtering Overview. . . . .	78
Transparent Mode . . . . .	78
Non-Transparent Mode . . . . .	79
Transparent Mode for HTTPS Traffic . . . . .	79
Order of Evaluation . . . . .	81
Web Filtering Configuration Examples. . . . .	82
Blocking Specific URLs . . . . .	83
Verifying Filtering . . . . .	83
Filtering by Content Category . . . . .	84
Filtering by Keyword . . . . .	85
Allowing Specific Sites . . . . .	86
Redirecting Users . . . . .	86
Handling Different Groups of Users . . . . .	88
Handling Different Time Periods . . . . .	90
Transparent Mode HTTPS Filtering . . . . .	93
Creating a Whitelist . . . . .	95
Web Filtering Commands . . . . .	97

restart webproxy	101
service webproxy append-domain <domain>	102
service webproxy default-port <port>	103
service webproxy domain-block <domain>	104
service webproxy domain-noncache <domain>	105
service webproxy listen-address <ipv4>	106
service webproxy listen-address <ipv4> disable-transparent	108
service webproxy listen-address <ipv4> port <port>	110
service webproxy proxy-bypass <address>	112
.....	114
service webproxy reply-block-mime <mime-type>	115
service webproxy reply-body-max-size <size>	117
service webproxy url-filtering disable	118
service webproxy url-filtering squidguard	119
service webproxy url-filtering squidguard allow-category <category>	120
service webproxy url-filtering squidguard allow-ipaddr-url	122
service webproxy url-filtering squidguard auto-update update-hour <hour>	124
service webproxy url-filtering squidguard block-category <category>	126
service webproxy url-filtering squidguard default-action <action>	128
service webproxy url-filtering squidguard enable-safe-search	130
service webproxy url-filtering squidguard local-block <address>	132
service webproxy url-filtering squidguard local-block-keyword <keyword>	134
service webproxy url-filtering squidguard local-block-url <url>	136
service webproxy url-filtering squidguard local-ok <address>	138
service webproxy url-filtering squidguard local-ok-url <url>	140
service webproxy url-filtering squidguard log <category>	142
service webproxy url-filtering squidguard redirect-url <url>	144
service webproxy url-filtering squidguard rule <rule-num>	146
service webproxy url-filtering squidguard rule <rule-num> allow-category <category>	148
service webproxy url-filtering squidguard rule <rule-num> allow-ipaddr-url	150
service webproxy url-filtering squidguard rule <rule-num> block-category <category>	152
service webproxy url-filtering squidguard rule <rule-num> default-action <action>	154
service webproxy url-filtering squidguard rule <rule-num> description <desc>	156
service webproxy url-filtering squidguard rule <rule-num> enable-safe-search	158
service webproxy url-filtering squidguard rule <rule-num> local-block <address>	160
service webproxy url-filtering squidguard rule <rule-num> local-block-keyword <keyword>	162
service webproxy url-filtering squidguard rule <rule-num> local-block-url <url>	164
service webproxy url-filtering squidguard rule <rule-num> local-ok <address>	166
service webproxy url-filtering squidguard rule <rule-num> local-ok-url <url>	168
service webproxy url-filtering squidguard rule <rule-num> log <category>	170
service webproxy url-filtering squidguard rule <rule-num> redirect-url <url>	172

---

service webproxy url-filtering squidguard rule <rule-num> source-group <group-name> . . . . .	174
service webproxy url-filtering squidguard rule <rule-num> time-period <period-name> . . . . .	176
service webproxy url-filtering squidguard source-group <group-name> . . . . .	178
service webproxy url-filtering squidguard source-group <group-name> address <addr> . . . . .	180
service webproxy url-filtering squidguard source-group <group-name> description <desc>. . . . .	182
service webproxy url-filtering squidguard source-group <group-name> domain <domain> . . . . .	184
service webproxy url-filtering squidguard time-period <period-name>. . . . .	186
service webproxy url-filtering squidguard time-period <period-name> days <day> time <time> . . . . .	188
service webproxy url-filtering squidguard time-period <period-name> description <desc> . . . . .	190
service webproxy url-filtering squidguard vyattaguard https-filter . . . . .	192
service webproxy url-filtering squidguard vyattaguard mode . . . . .	194
show webproxy blacklist categories. . . . .	196
show webproxy blacklist domains . . . . .	197
show webproxy blacklist log . . . . .	198
show webproxy blacklist search <filter> . . . . .	200
show webproxy blacklist urls . . . . .	202
show webproxy log . . . . .	204
show webproxy vyattaguard categories . . . . .	205
show webproxy vyattaguard search <filter> . . . . .	206
update webproxy blacklists. . . . .	207
update webproxy vyattaguard . . . . .	208
<b>Glossary of Acronyms . . . . .</b>	<b>209</b>

# Quick List of Commands

Use this list to help you quickly locate commands.

.....	114
content-inspection inspect-all enable .....	24
content-inspection ips actions other <action> .....	31
content-inspection ips actions priority-1 <action>.....	25
content-inspection ips actions priority-2 <action>.....	27
content-inspection ips actions priority-3 <action>.....	29
content-inspection ips auto-update oink-code <code> .....	33
content-inspection ips auto-update snortvrt-subscription .....	35
content-inspection ips auto-update update-hour <hour>.....	37
content-inspection ips log local .....	39
content-inspection ips log remote-db .....	40
content-inspection ips log syslog .....	42
content-inspection ips modify-rules disable-sid <rule> .....	45
content-inspection ips modify-rules enable-sid <rule> .....	47
content-inspection ips modify-rules exclude-category <category> .....	49
content-inspection ips modify-rules include-category <category>.....	51
content-inspection ips modify-rules internal-network <ipv4net> .....	53
content-inspection ips policy .....	55
content-inspection traffic-filter <filter> .....	57
interfaces <interface> content-inspection <direction> enable .....	59
restart webproxy .....	101
service webproxy append-domain <domain>.....	102
service webproxy default-port <port>.....	103
service webproxy domain-block <domain> .....	104
service webproxy domain-noncache <domain> .....	105
service webproxy listen-address <ipv4> disable-transparent .....	108
service webproxy listen-address <ipv4> port <port>.....	110
service webproxy listen-address <ipv4> .....	106
service webproxy proxy-bypass <address> .....	112
service webproxy reply-block-mime <mime-type> .....	115
service webproxy reply-body-max-size <size> .....	117
service webproxy url-filtering disable .....	118

---

service webproxy url-filtering squidguard allow-category <category> . . . . .	120
service webproxy url-filtering squidguard allow-ipaddr-url . . . . .	122
service webproxy url-filtering squidguard auto-update update-hour <hour> . . . . .	124
service webproxy url-filtering squidguard block-category <category> . . . . .	126
service webproxy url-filtering squidguard default-action <action> . . . . .	128
service webproxy url-filtering squidguard enable-safe-search . . . . .	130
service webproxy url-filtering squidguard local-block <address> . . . . .	132
service webproxy url-filtering squidguard local-block-keyword <keyword> . . . . .	134
service webproxy url-filtering squidguard local-block-url <url> . . . . .	136
service webproxy url-filtering squidguard local-ok <address> . . . . .	138
service webproxy url-filtering squidguard local-ok-url <url> . . . . .	140
service webproxy url-filtering squidguard log <category> . . . . .	142
service webproxy url-filtering squidguard redirect-url <url> . . . . .	144
service webproxy url-filtering squidguard rule <rule-num> allow-category <category> . . . . .	148
service webproxy url-filtering squidguard rule <rule-num> allow-ipaddr-url . . . . .	150
service webproxy url-filtering squidguard rule <rule-num> block-category <category> . . . . .	152
service webproxy url-filtering squidguard rule <rule-num> default-action <action> . . . . .	154
service webproxy url-filtering squidguard rule <rule-num> description <desc> . . . . .	156
service webproxy url-filtering squidguard rule <rule-num> enable-safe-search . . . . .	158
service webproxy url-filtering squidguard rule <rule-num> local-block <address> . . . . .	160
service webproxy url-filtering squidguard rule <rule-num> local-block-keyword <keyword> . . . . .	162
service webproxy url-filtering squidguard rule <rule-num> local-block-url <url> . . . . .	164
service webproxy url-filtering squidguard rule <rule-num> local-ok <address> . . . . .	166
service webproxy url-filtering squidguard rule <rule-num> local-ok-url <url> . . . . .	168
service webproxy url-filtering squidguard rule <rule-num> log <category> . . . . .	170
service webproxy url-filtering squidguard rule <rule-num> redirect-url <url> . . . . .	172
service webproxy url-filtering squidguard rule <rule-num> source-group <group-name> . . . . .	174
service webproxy url-filtering squidguard rule <rule-num> time-period <period-name> . . . . .	176
service webproxy url-filtering squidguard rule <rule-num> . . . . .	146
service webproxy url-filtering squidguard source-group <group-name> address <addr> . . . . .	180
service webproxy url-filtering squidguard source-group <group-name> description <desc> . . . . .	182
service webproxy url-filtering squidguard source-group <group-name> domain <domain> . . . . .	184
service webproxy url-filtering squidguard source-group <group-name> . . . . .	178
service webproxy url-filtering squidguard time-period <period-name> days <day> time <time> . . . . .	188
service webproxy url-filtering squidguard time-period <period-name> description <desc> . . . . .	190
service webproxy url-filtering squidguard time-period <period-name> . . . . .	186



---

service webproxy url-filtering squidguard vyattaguard https-filter .....	192
service webproxy url-filtering squidguard vyattaguard mode .....	194
service webproxy url-filtering squidguard .....	119
show ips log .....	64
show ips rules summary .....	66
show ips summary .....	69
show ips update-log .....	71
show webproxy blacklist categories .....	196
show webproxy blacklist domains .....	197
show webproxy blacklist log .....	198
show webproxy blacklist search <filter> .....	200
show webproxy blacklist urls .....	202
show webproxy log .....	204
show webproxy vyattaguard categories .....	205
show webproxy vyattaguard search <filter> .....	206
update ips rules .....	72
update webproxy blacklists .....	207
update webproxy vyattaguard .....	208
zone-policy zone <to-zone> from <from-zone> content-inspection enable .....	74

# List of Examples

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

Example 1-1 IPS using the subscription-based service. ....	11
Example 1-2 Initial IPS setup using the registration-based service. ....	12
Example 1-3 Inspecting all traffic. ....	14
Example 1-4 Inspecting all traffic from an HTTP source. ....	15
Example 1-5 Inspecting HTTP traffic on eth0 ....	16
Example 1-6 Inspecting HTTP traffic passing between “public” and “lan” zones ....	18
Example 1-7 Converting from the registration-based service to the subscription-based service. ....	20
Example 1-8 “show ips log”: Displaying ips events ....	64
Example 1-9 “show ips rules summary”: Displaying information regarding IPS rules state. ....	66
Example 1-10 “show ips summary”: Displaying a summary of IPS alerts ....	69
Example 1-11 “show ips update-log”: Displaying ips rules update history ....	71
Example 1-12 “update ips rules”: Displaying output from a manual rules update ....	73
Example 2-1 Blocking specific URLs ....	83
Example 2-2 Verifying filtering ....	84
Example 2-3 Filtering by content category ....	84
Example 2-4 Filtering by keywords ....	85
Example 2-5 Allowing specific sites ....	86
Example 2-6 Redirecting users ....	87
Example 2-7 Handling different groups of users. ....	88
Example 2-8 Handling different time periods. ....	91
Example 2-9 Handling HTTPS filtering in transparent mode. ....	93
Example 2-10 Creating a whitelist. ....	95
Example 2-11 Restarting the webproxy service ....	101
Example 2-12 Displaying database categories ....	196
Example 2-13 Displaying database domains ....	197
Example 2-14 Displaying the blacklist log ....	198
Example 2-15 Displaying the blacklist log summary ....	199

---

Example 2-16	Searching for an IP address or URL in a database .....	200
Example 2-17	Displaying blacklisted URLs .....	202
Example 2-18	Viewing the web proxy log .....	204
Example 2-19	Downloading a squidGuard database .....	207

# Preface

This document describes the various deployment, installation, and upgrade options for Vyatta software.

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

- [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
- IP services

## Organization of This Guide

---

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

- [Quick List of Commands](#)  
Use this list to help you quickly locate commands.
- [List of Examples](#)  
Use this list to help you locate examples you'd like to try or look at.

This guide has the following chapters:

Chapter	Description	Page
<a href="#">Chapter 1: Intrusion Prevention System</a>	This chapter lists the commands for setting up intrusion detection and prevention on the Vyatta System.	1
<a href="#">Chapter 2: Web Filtering</a>	This chapter explains how to set up web filtering on the Vyatta system.	76
<a href="#">Glossary of Acronyms</a>		209

## Document Conventions

---

This guide uses the following advisory paragraphs, as follows.



**WARNING** Warnings alert you to situations that may pose a threat to personal safety.



**CAUTION** Cautions alert you to situations that might cause harm to your system or damage to equipment, or that may affect service.

**NOTE** Notes provide information you might need to avoid problems or configuration errors.

This document uses the following typographic conventions.

Monospace	Examples, command-line output, and representations of configuration nodes.
<b>bold Monospace</b>	Your input: something you type at a command line.
<b>bold</b>	Commands, keywords, and file names, when mentioned inline.  Objects in the user interface, such as tabs, buttons, screens, and panes.
<i>italics</i>	An argument or variable where you supply a value.
<key>	A key on your keyboard, such as <Enter>. Combinations of keys are joined by plus signs (“+”), as in <Ctrl>+c.
[ key1   key2]	Enumerated options for completing a syntax. An example is [enable   disable].
<i>num1–numN</i>	A inclusive range of numbers. An example is 1–65535, which means 1 through 65535, inclusive.
<i>arg1..argN</i>	A range of enumerated values. An example is eth0..eth3, which means eth0, eth1, eth2, or eth3.
<i>arg[ arg...]</i> <i>arg[,arg...]</i>	A value that can optionally represent a list of elements (a space-separated list and a comma-separated list, respectively).

## Vyatta Publications

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 the information you need.

Additional information is available on [www.vyatta.com](http://www.vyatta.com) and [www.vyatta.org](http://www.vyatta.org).



# Chapter 1: Intrusion Prevention System

This chapter lists the commands for setting up intrusion detection and prevention on the Vyatta System.

**NOTE** *Support for IPS and Snort is deprecated. In this release, Intrusion Prevention System (IPS) and support for Snort is deprecated. Note that documentation is not yet updated in this regard, so Vyatta documentation incorrectly indicates that Snort and IPS are supported. Please disregard any mention of IPS or Snort in Vyatta documentation.*

This chapter presents the following topics:

- [IPS Configuration](#)
- [IPS Commands](#)



# IPS Configuration

---

This section presents the following topics:

- [IPS Overview](#)
- [Initial Setup](#)
- [Specifying Traffic to Inspect](#)
- [Specifying How to Handle Malicious Traffic](#)
- [Ongoing Monitoring and Maintenance](#)
- [Planning Considerations](#)
- [IPS Configuration Examples](#)

## IPS Overview

An Intrusion Prevention System (IPS) monitors network and/or system activities looking for malicious or unwanted behavior and can react, in real time, to block or prevent those activities. Network-based IPS, for example, may operate in-line to monitor all network traffic for malicious code or attacks. When an attack is detected, the system can drop the malicious packets while allowing all other traffic to pass.

The Vyatta system uses the Snort engine ([www.snort.org](http://www.snort.org)) for intrusion detection and prevention. Snort can perform protocol analysis and content searching/matching. It can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more. Snort uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plug-in architecture.

**NOTE** Only IPv4 content inspection is currently supported.

Vyatta offers the subscription-based Vyatta PLUS Snort VRT Service and an open version of the IPS for the Vyatta Core. Only one of these two services can be configured at one time.

## Vyatta PLUS Snort VRT Service



Customers who purchase a Vyatta PLUS Snort VRT Service rules subscription can access all updates available for the paid Snort subscription service (Sourcefire VRT Certified Rules). These subscribers receive Snort rule base updates as new vulnerabilities are found, 30 days before users registered for the free Snort service receive the same updates. To use this feature, you must obtain an entitlement license from Vyatta.

## Vyatta Core IPS



Vyatta Core users can set the system to download rule updates according to the Snort registered service. To use this feature, you must have registered for the free Snort service with Snort.org.

Configuring the IPS system involves three main steps:

- 1 Initial setup - downloading IPS rules and configuring the system to retrieve rules updates automatically.
- 2 Specifying traffic to inspect - defining which traffic is to be inspected for malicious content.
- 3 Specifying how to handle malicious traffic - defining what to do with various severities of malicious traffic.

The details of these steps are outlined below.

## Initial Setup

Prior to using the IPS system, whether it is the Vyatta Core IPS or the Vyatta PLUS Snort VRT Service, there are some initial setup steps that must be performed. These steps are dependent on which of these services you are using.

### Initial Setup: Vyatta PLUS Snort VRT Service

Prior to using the Vyatta PLUS Snort VRT Service you must first:

- 1 Obtain an entitlement key for Vyatta PLUS as well as a service key for the Vyatta PLUS Snort VRT Service from Vyatta.
- 2 Set up entitlement and service keys on the Vyatta system according to the *Guide to Vyatta Software Licensing*.
- 3 Download the snort rules. For example, to download snort rules you would enter the following command at the operational mode command line on your Vyatta system:

```
vyatta@R1:~$ update ips rules snortvrt-subscription
```

The snort rules will then be downloaded to your system.

- 4 In configuration mode, set up the system to retrieve rules updates automatically (in this case at 2:00 am each morning):

```
vyatta@R1# set content-inspection ips auto-update snortvrt-subscription
vyatta@R1# set content-inspection ips auto-update update-hour 2
```

- 5 Finally, define your internal network using the `content-inspection ips modify-rules internal-network <ipv4net>` command. For example, if your internal subnet is 192.168.1.0/24 you would specify the following:

```
vyatta@R1# set content-inspection ips modify-rules internal-network
192.168.1.0/24
```

**NOTE** The configuration cannot be committed at this point as the traffic to inspect must first be specified.

## Initial Setup: Vyatta Core IPS

Prior to using the Vyatta Core IPS you must first do the following:

- 1 Register to obtain snort rules updates at [www.snort.org](http://www.snort.org).
- 2 Obtain an “oinkcode” from [www.snort.org](http://www.snort.org).
- 3 Download the snort rules using your oinkcode. For example, to download snort rules using an oinkcode of “112233” you would enter the following command at the operational mode command line on your Vyatta system:
- 4 `vyatta@R1:~$ update ips rules oinkcode 112233`
- 5 The snort rules will then be downloaded to your system.
- 6 In configuration mode, set up the system to retrieve rules updates automatically (in this case at 2:00 am each morning):

```
vyatta@R1# set content-inspection ips auto-update oinkcode 112233
vyatta@R1# set content-inspection ips auto-update update-hour 2
```

- 7 Finally, define your internal network using the `content-inspection ips modify-rules internal-network <ipv4net>` command. For example, if your internal subnet is 192.168.1.0/24 you would specify the following:

```
vyatta@R1# set content-inspection ips modify-rules internal-network
192.168.1.0/24
```

**NOTE** The configuration cannot be committed at this point as the traffic to inspect must first be specified.

## Specifying Traffic to Inspect

Once the IPS rules have been downloaded and updated rules are set to download automatically, the next step is to specify which traffic is to be inspected. There are two components to this:

- 1 Traffic location and direction - specifying the interface(s) or zone(s) the traffic to be inspected is on and in which direction it is flowing.
- 2 Traffic type - specifying the type of traffic to inspect (e.g., HTTP traffic).

## Traffic Location and Direction

The traffic location and direction can be specified in one of three ways:

- 1 All traffic
- 2 Traffic on specific interfaces
- 3 Traffic from specific zones

**NOTE** *The first method and the last two methods are mutually exclusive and should not be configured at the same time.*

### ALL TRAFFIC

One option is to specify that all traffic is to be inspected. This can be done using the following configuration command:

```
vyatta@R1# set content-inspection inspect-all enable
```

The main disadvantage of this method is that it is resource intensive and can slow down system performance. In most cases, limiting the traffic to be inspected to specific traffic flows is a better alternative.

### TRAFFIC ON SPECIFIC INTERFACES

Another option is to inspect traffic flowing on a specific interface. Options include “in” (for inbound, through traffic), “out” (for outbound, through traffic), and “local” (for inbound, locally destined traffic). One or more of these directional parameters can be defined for a given interface. For example, to inspect all through traffic on interface eth0 the following configuration commands would be used:

```
vyatta@R1# set interfaces ethernet eth0 content-inspection in enable
vyatta@R1# set interfaces ethernet eth0 content-inspection out enable
```

**NOTE** *In general, it is best to specify both directions for a traffic flow because many of the IPS rules rely on seeing both sides of a connection.*

### TRAFFIC FROM SPECIFIC ZONES

The final option is to inspect traffic flowing from one zone to another. For example, to inspect all traffic flowing between the “public” and “lan” zones (where these zones are previously defined), the following configuration commands would be used:

```
vyatta@R1# set zone-policy zone lan from public content-inspection enable
vyatta@R1# set zone-policy zone public from lan content-inspection enable
```

**NOTE** In general, it is best to specify both directions for a traffic flow because many of the IPS rules rely on seeing both sides of a connection.

## Traffic Type

Once the traffic flows to inspect have been identified it is necessary to identify the traffic within those flows that is to be inspected by IPS. This traffic can be specified in one of two ways:

- 1 All traffic
- 2 Traffic identified by a filter (firewall rule set)

### ALL TRAFFIC

To specify that all traffic within a given flow is to be inspected, the following configuration command would be used:

```
vyatta@R1# set content-inspection traffic-filter preset all
```

### TRAFFIC IDENTIFIED BY A FILTER

It is also possible to narrow the scope of the content inspection by specifying the type of traffic to inspect. For example, to inspect only traffic to and from an HTTP source, the following configuration command would be used:

```
vyatta@R1# set content-inspection traffic-filter custom inspect_http
```

where “inspect\_http” is a previously defined firewall rule set that looks as follows:

```
vyatta@R1# show firewall name inspect_http
default-action accept
rule 1 {
    action inspect
    destination {
        port http
    }
    protocol tcp
}
rule 2{
    action inspect
    source {
        port http
    }
    protocol tcp
}
```

Note that the default action here is to accept traffic that will not pass through content inspection. If the default action was not set to **accept** (e.g. it was set to **drop** or **reject**) then all non-inspected traffic would be dropped. Filtered traffic is passed to content inspection to determine whether or not it is malicious and, if so, what to do with it.

Also note that the action is set to “inspect”. This is necessary for content inspection to occur.

**NOTE** *Filtering by traffic type is available with any of the traffic direction and location filters described above.*

## Specifying Which IPS Rules to Apply

Once you have filtered out the traffic that you wish to inspect, you need to specify which subset of the IPS rules database you would like to apply to that traffic. The first step is to apply an IPS policy and then to include additional rules or to exclude rules if necessary.

### Apply a Policy

IPS policies apply a pre-defined set of rules and are specified using [content-inspection ips policy](#). Options include **connectivity**, **security**, and **balanced**.

The **connectivity** setting provides a more limited set of rules where only the highest priority threats are scanned for. It is typically used in environments that are latency sensitive (for example, VOIP, financial transactions, etc.).

The **security** setting provides a very high level of security with an associated decrease in system performance. It is used in environments that are extremely security sensitive.

The **balanced** setting provides a balance of enabled rules - between **connectivity** and **security** in terms of its security level. This is the default and is appropriate for most cases.

### Include or Exclude Rules

IPS policies provide specific subsets of rules for different generalized use cases. Based on a review of the IPS log, a given network may be better served with changes to the selected subset. Four commands are provided to enable you to adjust the active subset of rules:

- [content-inspection ips modify-rules include-category <category>](#) - provides the ability to include a Snort rule category.
- [content-inspection ips modify-rules exclude-category <category>](#) - provides the ability to exclude a Snort rule category.
- [content-inspection ips modify-rules enable-sid <rule>](#) - provides the ability to enable a specific Snort rule.

- `content-inspection ips modify-rules disable-sid <rule>` - provides the ability to disable a specific Snort rule.

**NOTE** If a category is included, it overrides any disabled rules within that category. If a category is excluded, any rules that are explicitly enabled override the category exclusion. So, inclusion always takes precedence over exclusion.

## Specifying How to Handle Malicious Traffic

Once the malicious traffic has been identified, you need to specify what to do with it. By default, Snort rules classify malicious traffic according to the classifications shown in [Table 1-1](#). They are currently ordered with four default priorities where Priority 1 attacks are the most severe and Priority 4 attacks are the least severe.

Table 1-1 Default classifications

Classtype	Description	Priority
attempted-admin	Attempted Administrator Privilege Gain	Priority 1
attempted-user	Attempted User Privilege Gain	Priority 1
inappropriate-content	Inappropriate Content was Detected	Priority 1
policy-violation	Potential Corporate Privacy Violation	Priority 1
shellcode-detect	Executable code was detected	Priority 1
successful-admin	Successful Administrator Privilege Gain	Priority 1
successful-user	Successful User Privilege Gain	Priority 1
trojan-activity	A Network Trojan was detected	Priority 1
unsuccessful-user	Unsuccessful User Privilege Gain	Priority 1
web-application-attack	Web Application Attack	Priority 1
attempted-dos	Attempted Denial of Service	Priority 2
attempted-recon	Attempted Information Leak	Priority 2
bad-unknown	Potentially Bad Traffic	Priority 2
default-login-attempt	Attempt to login by a default username and password	Priority 2
denial-of-service	Detection of a Denial of Service Attack	Priority 2
misc-attack	Misc Attack	Priority 2

Table 1-1 Default classifications

Classtype	Description	Priority
non-standard-protocol	Detection of a non-standard protocol or event	Priority 2
rpc-portmap-decode	Decode of an RPC Query	Priority 2
successful-dos	Denial of Service	Priority 2
successful-recon-largescale	Large Scale Information Leak	Priority 2
successful-recon-limited	Information Leak	Priority 2
suspicious-filename-detect	A suspicious filename was detected	Priority 2
suspicious-login	An attempted login using a suspicious username was detected	Priority 2
system-call-detect	A system call was detected	Priority 2
unusual-client-port-connection	A client was using an unusual port	Priority 2
web-application-activity	Access to a potentially vulnerable web application	Priority 2
icmp-event	Generic ICMP event	Priority 3
misc-activity	Misc activity	Priority 3
network-scan	Detection of a Network Scan	Priority 3
not-suspicious	Not Suspicious Traffic	Priority 3
protocol-command-decode	Generic Protocol Command Decode	Priority 3
string-detect	A suspicious string was detected	Priority 3
unknown	Unknown Traffic	Priority 3
tcp-connection	A TCP connection was detected	Priority 4

You specify an action to take with each class of malicious traffic using the “**content-inspection ips actions**” commands. The actions available are: **alert**, **drop**, **pass**, and **sdrop**. These are defined 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).



For example, to drop priority 1 attacks, alert on priority 2 and 3 attacks, and pass all other traffic you would specify the following configuration commands:

```
vyatta@R1# set content-inspection ips actions priority-1 drop
vyatta@R1# set content-inspection ips actions priority-2 alert
vyatta@R1# set content-inspection ips actions priority-3 alert
vyatta@R1# set content-inspection ips actions other pass
```

## Ongoing Monitoring and Maintenance

Once the IPS system has been configured and is running it requires continued effort to monitor and maintain. For the most part, this involves reviewing the IPS log periodically and potentially adjusting the active Snort rules based on the findings in the logs.

## Planning Considerations

The following points should be taken into consideration when planning an IPS configuration:

- **Content inspection occurs after firewall** - Content inspection occurs within the Vyatta system after the firewall. This means that only traffic that is allowed past the firewall can be inspected by IPS.
- **Configure for flow-based rule matches** - For IPS to work for flow-based rule matches, it must be either configured to inspect all traffic or configured such that it sees both sides of the connection (for example, traffic is inspected in both the “in” and “out” direction on an interface). In addition, if a specific traffic type is to be filtered (for example, HTTP traffic), firewall rules must be set to inspect for both source and destination of that traffic type (for example, “source port http” and “destination port http”).
- **Approaches to level of security** - One way to approach configuring a network for IPS is to be very restrictive in the types of traffic that are allowed and then to adjust the configuration to allow for additional traffic flows as required. Another approach is to be less restrictive to start with and then restrict traffic types as required. In general, the first approach (more restrictive to begin with) works best when deploying a new network and the second approach (less restrictive to begin with) works best when deploying IPS into an existing network.
- **Concurrent use of URL filtering** - For IPS and URL filtering to work simultaneously, the IPS configuration must be set to **inspect-all enable** and no **internal-networks** can be defined.

## IPS Configuration Examples

This section presents the following topics:

- [Initial Setup Examples](#)
- [Inspecting Specific Traffic Examples](#)
- [Changing from Registration-based to Subscription-Based IPS](#)

## Initial Setup Examples

Initial setup for a Vyatta Core IPS is slightly different than that of the Vyatta PLUS Snort VRT Service. Once the initial setup is complete the rest of the configuration is the same for both system types.

### INITIAL SETUP FOR VYATTA PLUS SNORT VRT SERVICE

[Example 1-1](#) enables subscription-based intrusion prevention on the Vyatta system and sets it to automatically update the rule base. In this example:

- We assume that a subscription for the Vyatta Plus Snort VRT Service is in effect, that entitlement and service keys have been configured, and that Snort rules have been downloaded using the [update ips rules](#) command.
- A check for Snort rule base updates is to occur each morning at 2 AM.
- All traffic not sourced from the internal network (192.168.1.0/24) is to be inspected.

To complete the initial configuration of the IPS using the registration-based service on the Vyatta system, perform the following steps in configuration mode:

Example 1-1 IPS using the subscription-based service.

Step	Command
Specify the snortvrt subscription service is to be used for Snort rule base updates.	<code>vyatta@R1# set content-inspection ips auto-update snortvrt-subscription</code>
Update Snort rule base at 2:00 am.	<code>vyatta@R1# set content-inspection ips auto-update update-hour 2</code>
Specify the internal network.	<code>vyatta@R1# set content-inspection ips modify-rules internal-network 192.168.1.0/24</code>

---

**Example 1-1** IPS using the subscription-based service.

Show the IPS configuration. Note that the configuration cannot be committed at this point because the minimal configuration for content-inspection has not yet been created.	<pre>vyatta@R1# show content-inspection +ips { +  auto-update { +    snortvrt-subscription +    update-hour 2 +  } +  modify-rules { +    internal-network 192.168.1.0/24 +  } +}</pre>
--	---

---

### INITIAL SETUP FOR VYATTA CORE IPS

[Example 1-2](#) enables registration-based intrusion prevention on the Vyatta system and sets it to automatically update the rule base. In this example:

- We assume that a Snort “oink-code” of 112233 has been obtained from snort.org and Snort rules were downloaded using the [update ips rules](#) command.
- A check for Snort rule base updates is to occur each morning at 2 AM.
- All traffic not sourced from the internal network (192.168.1.0/24) is to be inspected.

To complete the initial configuration of the IPS using the registration-based service on the Vyatta system, perform the following steps in configuration mode:

**Example 1-2** Initial IPS setup using the registration-based service.

Step	Command
Specify the Snort registration service “oink” code obtained from snort.org.	<pre>vyatta@R1# set content-inspection ips auto-update oink-code 112233</pre>
Update Snort rule base at 2:00 AM.	<pre>vyatta@R1# set content-inspection ips auto-update update-hour 2</pre>
Specify the internal network.	<pre>vyatta@R1# set content-inspection ips modify-rules internal-network 192.168.1.0/24</pre>

---

Example 1-2 Initial IPS setup using the registration-based service.

---

Show the IPS configuration. Note that the configuration cannot be committed at this point because the minimal configuration for content-inspection has not yet been created.	<pre>vyatta@R1# show content-inspection +ips { +  auto-update { +    oink-code 112233 +    update-hour 2 +  } +  modify-rules { +    internal-network 192.168.1.0/24 +  } +}</pre>
--	--

---

## Inspecting Specific Traffic Examples

The following examples assume a Vyatta PLUS Snort VRT Service has been initialized as in the previous example, though the same configuration steps would be performed for a Vyatta Core IPS system. In addition, these examples handle any malicious traffic in the same way - drop priority 1 attacks, alert on priority 2 or 3 attacks, and allow all other traffic.

### INSPECTING ALL TRAFFIC

To inspect all traffic on a system (except that from the internal network) the configuration looks as follows:

## Example 1-3 Inspecting all traffic

---

Step	Command
Show the IPS configuration.	<pre>vyatta@R1# show content-inspection inspect-all {   enable } ips {   actions {     other pass     priority-1 drop     priority-2 alert     priority-3 alert   }   auto-update {     snortvrt-subscription     update-hour 2   }   modify-rules {     internal-network 192.168.1.0/24   } } traffic-filter {   preset all }</pre>

---

The “**inspect-all enable**” configuration turns on inspection of all forwarded, locally destined, and locally originated traffic. The “**traffic-filter preset all**” configuration does not filter this traffic any further.

**NOTE** Care should be taken when making the choice to inspect all system traffic as it is resource intensive and can impact system performance.

### INSPECTING ALL TRAFFIC FROM AN HTTP SOURCE

To inspect all traffic from an HTTP source (except that from the internal network) the configuration looks as follows:

## Example 1-4 Inspecting all traffic from an HTTP source

---

Step	Command
Show the IPS configuration.	<pre>vyatta@R1# show content-inspection inspect-all {   enable } ips {   actions {     other pass     priority-1 drop     priority-2 alert     priority-3 alert   }   auto-update {     snortvrt-subscription     update-hour 2   }   modify-rules {     internal-network 192.168.1.0/24   } } traffic-filter {   custom inspect_http }</pre>
Show the firewall configuration.	<pre>vyatta@R1# show firewall name inspect_http default-action accept rule 1 {   action inspect   destination {     port http   }   protocol tcp } rule 2{   action inspect   source {     port http   }   protocol tcp }</pre>

---

The “**inspect-all enable**” configuration turns on inspection of all forwarded, locally destined, and locally originated traffic. The “**traffic-filter custom inspect\_http**” configuration points to a filter that is defined in the firewall section of the configuration tree. This filter specifies that only traffic to and from an HTTP source should be examined.

**NOTE** Care should be taken when making the choice to inspect all system traffic as it is resource intensive and can impact system performance.

## INSPECTING INBOUND AND OUTBOUND TRAFFIC ON AN INTERFACE FROM AN HTTP SOURCE

Rather than inspecting traffic on all interfaces, the following example specifies that only HTTP traffic passing through the device on a specific interface (eth0) is to be examined. The configuration looks as follows:

Example 1-5 Inspecting HTTP traffic on eth0

Step	Command
Show the IPS configuration.	<pre>vyatta@R1# show content-inspection ips {     actions {         other pass         priority-1 drop         priority-2 alert         priority-3 alert     }     auto-update {         snortvrt-subscription         update-hour 2     }     modify-rules {         internal-network 192.168.1.0/24     } } traffic-filter {     custom inspect_http }</pre>

---

**Example 1-5 Inspecting HTTP traffic on eth0**

---

Show the firewall configuration.    `vyatta@R1# show firewall name inspect_http`

```
default-action accept
rule 1 {
  action inspect
  destination {
    port http
  }
  protocol tcp
}
rule 2{
  action inspect
  source {
    port http
  }
  protocol tcp
}
```

---

Show the eth0 configuration.    `vyatta@R1# show interfaces ethernet eth0`

```
address 12.34.56.78/24
content-inspection {
  in {
    enable
  }
  out {
    enable
  }
}
```

---

Note that the “**inspect-all enable**” configuration has been removed. Instead, the “**content-inspection in enable**” and “**content-inspection out enable**” configuration has been added to the configuration for eth0. The rest of the configuration is the same.

### INSPECTING INTER-ZONE TRAFFIC FROM AN HTTP SOURCE

The following example specifies that only HTTP traffic passing between the “**public**” and “**lan**” zones is to be inspected. The configuration looks as follows:



---

**Example 1-6** Inspecting HTTP traffic passing between “public” and “lan” zones

---

<b>Step</b>	<b>Command</b>
Show the IPS configuration.	<pre>vyatta@R1# show content-inspection ips {     actions {         other pass         priority-1 drop         priority-2 alert         priority-3 alert     }     auto-update {         snortvrt-subscription         update-hour 2     }     modify-rules {         internal-network 192.168.1.0/24     } } traffic-filter {     custom inspect_http }</pre>
Show the firewall configuration.	<pre>vyatta@R1# show firewall name inspect_http default-action accept rule 1 {     action inspect     destination {         port http     }     protocol tcp } rule 2 {     action inspect     source {         port http     }     protocol tcp }</pre>

---

---

**Example 1-6** Inspecting HTTP traffic passing between “public” and “lan” zones

---

Show the “lan” zone configuration.	<pre>vyatta@R1# show zone-policy zone lan default-action drop from public {   content-inspection {     enable   }   firewall {     name allow-all   } } interface eth1</pre>
Show the related firewall configuration. Allow all incoming traffic.	<pre>vyatta@R1# show firewall name allow-all default-action accept</pre>
Show the “public” zone configuration.	<pre>vyatta@R1# show zone-policy zone public default-action drop from lan {   content-inspection {     enable   }   firewall {     name allow-state-established-related   } } interface eth0</pre>
Show the related firewall configuration. Allow only established and related traffic.	<pre>vyatta@R1# show firewall name allow-state-established-related default-action drop rule 1 {   action accept   state {     established enable     related enable   } }</pre>

---

The difference from the previous example is that instead of enabling content-inspection on a specific interface, it is enabled on traffic from one zone to another. Specifically, the “**content-inspection enable**” configuration has been added to the configuration for traffic from the “public” zone to the “lan” zone and for traffic from the “lan” zone to the “public” zone.

## Changing from Registration-based to Subscription-Based IPS

**Example 1-7** converts a system that is using registration-based IPS (Vyatta Core IPS) to one using subscription-based IPS (Vyatta PLUS Snort VRT Service IPS). It assumes that an Entitlement key, username, password, and Service key for **snortvrtsub** have been obtained from Vyatta and that, other than changing from a registration-based service to a subscription-based service, the IPS configuration remains the same. In this example:

- The “oink-code” for the registration-based service is removed
- The IPS system is configured to use the subscription-based service
- The entitlement information is configured
- The rules base is updated

To convert from a registration-based service to a subscription-based service on the Vyatta system, perform the following steps in configuration mode:

**Example 1-7** Converting from the registration-based service to the subscription-based service.

Step	Command
Remove the “oink-code” used by the registration-based service.	<code>vyatta@R1# delete content-inspection ips auto-update oink-code</code>
Configure IPS for the subscription-based service.	<code>vyatta@R1# set content-inspection ips auto-update snortvrt-subscription</code>
Set the entitlement key.	<code>vyatta@R1# set entitlement entitlement-key 5300-4000-0200-0050-0001</code>
Set the entitlement username.	<code>vyatta@R1# set entitlement username testuser</code>
Set the entitlement password.	<code>vyatta@R1# set entitlement password 9testpassword9</code>
Add the service key for <b>snortvrtsub</b> to the entitlement key.	<code>vyatta@R1# add entitlement service-assignment 5300-4000-0200-0050-0001 5407-4000-1546-0221-8600</code>
Commit the change	<code>vyatta@R1# commit</code>
Exit to operational mode.	<code>vyatta@R1# exit</code> <code>exit</code> <code>vyatta@R1:~\$</code>

---

**Example 1-7** Converting from the registration-based service to the subscription-based service.

---

```

Confirm the entitlement configuration.      vyatta@R1:~$ show entitlement

=====
Current entitlement information
=====

Available entitlement keys:
  Entitlement key: 5300-4000-0200-0050-0001
    Start date: 2009/07/01 00:00
    Expire date: 2011/07/01 00:00
  Services assigned to this entitlement key:
    [snortvrtsub] service:
      Service key: 5407-4000-1546-0221-8600
      Start date: 2010/07/01 00:00
      Expire date: 2011/07/01 00:00

Available and unassigned service keys:
  None

vyatta@R1:~$

```

---

```

Update IPS rules base.      vyatta@R1:~$ update ips rules
Starting Snort VRT subscription update [manual]
19035189/19035189 - 100%
New subscription rules downloaded.

```

---

# IPS Commands

This chapter contains the following commands.

Configuration Commands	
<code>content-inspection inspect-all enable</code>	Turns on inspection of all traffic in all directions.
<code>content-inspection ips actions priority-1 &lt;action&gt;</code>	Specifies the action to take for packets matching Priority 1 IPS rules.
<code>content-inspection ips actions priority-2 &lt;action&gt;</code>	Specifies the action to take for packets matching Priority 2 IPS rules.
<code>content-inspection ips actions priority-3 &lt;action&gt;</code>	Specifies the action to take for packets matching Priority 3 IPS rules.
<code>content-inspection ips actions other &lt;action&gt;</code>	Specifies what to do with packets matching IPS rules with priority other than 1, 2, or 3.
<code>content-inspection ips auto-update oink-code &lt;code&gt;</code>	Specifies the registration code for automatic Snort rule base updates.
<code>content-inspection ips auto-update snortvrt-subscription</code>	Retrieves subscription-based Snort rule base updates.
<code>content-inspection ips auto-update update-hour &lt;hour&gt;</code>	Specifies the hour of the day for daily Snort rule base updates.
<code>content-inspection ips log local</code>	Specifies that alerts are to be stored in a local file.
<code>content-inspection ips log remote-db</code>	Specifies the database that IPS alerts are to be stored in.
<code>content-inspection ips log syslog</code>	Specifies that alerts are to be stored in the system logfile.
<code>content-inspection ips modify-rules disable-sid &lt;rule&gt;</code>	Allows you to selectively disable a Snort rule.
<code>content-inspection ips modify-rules enable-sid &lt;rule&gt;</code>	Allows you to selectively enable a Snort rule.
<code>content-inspection ips modify-rules exclude-category &lt;category&gt;</code>	Allows you to selectively exclude an entire Snort rule category.
<code>content-inspection ips modify-rules include-category &lt;category&gt;</code>	Allows you to selectively include an entire Snort rule category.
<code>content-inspection ips modify-rules internal-network &lt;ipv4net&gt;</code>	Allows you to specify an internal network.

---

<code>content-inspection ips policy</code>	Allows you to specify a subset of rules based on various policy types.
<code>content-inspection traffic-filter &lt;filter&gt;</code>	Specifies which traffic is to be processed by Vyatta IPS functions.
<code>interfaces &lt;interface&gt; content-inspection &lt;direction&gt; enable</code>	Applies content inspection to a traffic stream on an interface.
<code>zone-policy zone &lt;to-zone&gt; from &lt;from-zone&gt; content-inspection enable</code>	Applies content inspection to traffic arriving at the “to” zone from the “from” zone.

---

**Operational Commands**

---

<code>show ips log</code>	Displays alerts logged by the IPS.
<code>show ips rules summary</code>	Displays a summary of the enabled and disabled IPS rules.
<code>show ips summary</code>	Displays a summary of all IPS alerts.
<code>show ips update-log</code>	Displays the history of automatic IPS rules updates.
<code>update ips rules</code>	Updates the IPS rules base.

---

## content-inspection inspect-all enable

Turns on inspection of all traffic in all directions.

---

### Syntax

```
set content-inspection inspect-all enable
delete content-inspection inspect-all
show content-inspection inspect-all
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
content-inspection{
  inspect-all {
    enable
  }
}
```

---

### Parameters

None.

---

### Default

Traffic is not inspected on all interfaces in all directions.

---

### Usage Guidelines

Use this command to specify that all system traffic is to be inspected. The scope of this command can be limited using the [content-inspection traffic-filter <filter>](#) command.

Inspecting all system traffic is resource intensive. A less resource intensive alternative is to inspect selected traffic on specific interfaces using [interfaces <interface>](#) [content-inspection <direction> enable](#).

Use the **set** form of this command to specify that all traffic on all system traffic is to be inspected (based on **traffic-filter** settings).

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

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

## 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 action
    }
  }
}
```

---

### Parameters

---

<i>action</i>	The action to take when a packet matches a priority 1 rule. Supported values are as follows: <b>alert:</b> Allows the packet and log an alert. <b>drop:</b> Drops the packet and log an alert. <b>pass:</b> Allows the packet and take no further action. <b>sdrop:</b> 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 1 Intrusion Prevention System (IPS) rules.

Rule categories and the priorities for each category are found in `/etc/snort/classifications.config`.

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 action
    }
  }
}
```

---

### Parameters

---

<i>action</i>	The action to take when a packet matches a priority 2 rule. Supported values are as follows: <b>alert:</b> Allows the packet and log an alert. <b>drop:</b> Drops the packet and log an alert. <b>pass:</b> Allows the packet and take no further action. <b>sdrop:</b> 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 Prevention System (IPS) rules.

Rule categories and the priorities for each category are found in `/etc/snort/classifications.config`.

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 action
    }
  }
}
```

---

### Parameters

---

<i>action</i>	The action to take when a packet matches a priority 3 rule. Supported values are as follows: <b>alert:</b> Allows the packet and log an alert. <b>drop:</b> Drops the packet and log an alert. <b>pass:</b> Allows the packet and take no further action. <b>sdrop:</b> 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 Prevention System (IPS) rules.

Rule categories and the priorities for each category are found in `/etc/snort/classifications.config`.

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 action
    }
  }
}
```

---

### Parameters

---

<i>action</i>	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:  <b>alert:</b> Allows the packet and log an alert. <b>drop:</b> Drops the packet and log an alert. <b>pass:</b> Allows the packet and take no further action. <b>sdrop:</b> 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 Prevention System (IPS) rules other than rules with priority 1, 2, or 3.

Rule categories and the priorities for each category are found in `/etc/snort/classifications.config`.

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>

Specifies the registration 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 code
    }
  }
}
```

---

### Parameters

<i>code</i>	Mandatory if updates are to be received. The “oink” code generated at <a href="http://www.snort.org">www.snort.org</a> . 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 registration-based Snort rule updates.

The Vyatta system uses the Snort ([www.snort.org](http://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. These updates are free and are available 30 days after they are made available to Snort subscription users (see [“content-inspection ips auto-update snortvrt-subscription”](#) on page 35 for information on the subscription-based version).

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.

**NOTE** *If you configure the system to retrieve the registration-based Snort rule base, do not also configure it to retrieve the subscription-based rule base using [content-inspection ips auto-update snortvrt-subscription](#) command.*

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 snortvrt-subscription

Retrieves subscription-based Snort rule base updates.

---

## Availability

Vyatta Plus

---

## Syntax

```
set content-inspection ips auto-update snortvrt-subscription
delete content-inspection ips auto-update snortvrt-subscription
show content-inspection ips auto-update
```

---

## Command Mode

Configuration mode.

---

## Configuration Statement

```
content-inspection{
  ips {
    auto-update {
      snortvrt-subscription
    }
  }
}
```

---

## Parameters

None.

---

## Default

None.

---

## Usage Guidelines

Use this command to specify that subscription-based Snort rule base updates are to be downloaded from Vyatta. To use this feature:

- A subscription for the Vyatta Plus service **snortvrtsub** must be in effect

- The corresponding Vyatta entitlement key and service key must have been obtained from Vyatta and configured on the system (see the Guide to Vyatta Software Licensing for further information about entitlement and service keys).

The Vyatta system uses the Snort ([www.snort.org](http://www.snort.org)) engine for intrusion detection. The Snort rule base is available directly from Snort and can be automatically downloaded (after registering with Snort) using `content-inspection ips auto-update oink-code <code> command`. These updates are free and are available 30 days after they are made available to Snort subscription customers. Snort subscription customers, on the other hand, receive updates when new vulnerabilities are found.

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.

**NOTE** *If you configure the system to retrieve the subscription-based Snort rule base, do not also configure it to retrieve the registration-based rule base using `content-inspection ips auto-update oink-code <code> command`.*

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 hour
    }
  }
}
```

---

### Parameters

<i>hour</i>	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.

## content-inspection ips log local

Specifies that alerts are to be stored in a local file.

---

### Syntax

```
set content-inspection ips log local
delete content-inspection ips log local
show content-inspection ips log
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
content-inspection{
  ips {
    log {
      local
    }
  }
}
```

---

### Parameters

None.

---

### Default

IPS alerts are stored in a local file.

---

### Usage Guidelines

Use this command to specify that IPS alerts are to be stored in a local file.

Use the **set** form of this command to specify that IPS alerts are to be stored in a local file.

Use the **delete** form of this command to restore the default behavior for IPS alert logging.

Use the **show** form of this command to display the IPS alert logging setting.

## content-inspection ips log remote-db

Specifies the database that IPS alerts are to be stored in.

---

### Syntax

```
set content-inspection ips log remote-db db-name db-name [db-type db-type] host
hostname password password username username
```

```
delete content-inspection ips log remote-db [db-name] [db-type] [host] [password]
[username]
```

```
show content-inspection ips log remote-db [db-name] [db-type] [host] [password]
[username]
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
content-inspection{
  ips {
    log {
      remote-db {
        db-name db-name
        db-type db-type
        host hostname
        password password
        username username
      }
    }
  }
}
```

---

### Parameters

---

<i>db-name</i>	Mandatory. The name of the database that IPS alerts are to be stored in.
----------------	--

---

<i>db-type</i>	The type of database that IPS alerts are to be stored in. Supported values are as follows:
----------------	---

**mysql**: Store IPS alerts in a MySQL database.

The default is **mysql**.

---

---

<i>hostname</i>	Mandatory. The hostname or IPv4 address of the database server that IPS alerts are to be stored on.
<i>password</i>	Mandatory. The password associated with the username used to access the database that IPS alerts are to be stored in.
<i>username</i>	Mandatory. The username used to access the database that IPS alerts are to be stored in.

---

---

### Default

None.

---

### Usage Guidelines

Use this command to specify the database that IPS alerts are to be stored in.

**NOTE** *Instructions for creating a MySQL database to accept IPS alerts are beyond the scope of the Vyatta documentation but can be found in the documentation at [www.snort.org](http://www.snort.org).*

Use the **set** form of this command to specify the database that IPS alerts are to be stored in.

Use the **delete** form of this command to remove the database settings.

Use the **show** form of this command to display the database settings.



## content-inspection ips log syslog

Specifies that alerts are to be stored in the system logfile.

---

### Syntax

```
set content-inspection ips log syslog facility facility level level
delete content-inspection ips log syslog [facility] [level]
show content-inspection ips log syslog [facility] [level]
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
content-inspection{
  ips {
    log {
      syslog {
        facility facility
        level level
      }
    }
  }
}
```

---

### Parameters

<i>facility</i>	The kinds of messages that will be sent to the system logfile. Please see the Usage Guidelines below for supported facilities.
<i>level</i>	The minimum severity of messages that will be sent to the system logfile. Please see the Usage Guidelines below for meanings of these severity levels.

---

### Default

IPS alerts are stored in a local file.

---

### Usage Guidelines

Use this command to specify that IPS alerts are to be stored in the system logfile.

The Vyatta system supports standard syslog facilities. These are as follows:

Table 1-2 Syslog facilities

<b>Facility</b>	<b>Description</b>
auth	Authentication and authorization
authpriv	Non-system authorization
cron	Cron daemon
daemon	System daemons
kern	Kernel
lpr	Line printer spooler
mail	Mail subsystem
mark	Timestamp
news	USENET subsystem
security	Security subsystem
syslog	System logging
user	Application processes
uucp	UUCP subsystem
local0	Local facility 0
local1	Local facility 1
local2	Local facility 2
local3	Local facility 3
local4	Local facility 4
local5	Local facility 5
local6	Local facility 6
local7	Local facility 7
all	All facilities excluding "mark"

Log messages generated by the Vyatta system will be associated with one of the following levels of severity.

Table 1-3 Syslog message severities

Severity	Meaning
emerg	Emergency. A general system failure or other serious failure has occurred, such that the system is unusable.
alert	Alert. Immediate action is required to prevent the system from becoming unusable—for example, because a network link has failed, or the database has become compromised.
crit	Critical. A critical condition exists, such as resource exhaustion—for example, the system is out of memory, CPU processing thresholds are being exceeded, or a hardware failure has occurred.
err	Error. An error condition has occurred, such as a failed system call. However, the system is still functioning.
warning	Warning. An event has occurred that has the potential to cause an error, such as invalid parameters being passed to a function. This situation should be monitored.
notice	Notice. A normal but significant event has occurred, such as an unexpected event. It is not an error, but could potentially require attention.
info	Informational. Normal events of interest are being reported as they occur.
debug	Debug level. Trace-level information is being provided.

Use the **set** form of this command to specify that IPS alerts are to be stored in the system logfile.

Use the **delete** form of this command to delete IPS alert logging settings.

Use the **show** form of this command to display the IPS alert logging settings.

## content-inspection ips modify-rules disable-sid <rule>

Allows you to selectively disable a Snort rule.

---

### Syntax

```
set content-inspection ips modify-rules disable-sid rule
delete content-inspection ips modify-rules disable-sid rule
show content-inspection ips modify-rules disable-sid
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
content-inspection{
  ips {
    modify-rules {
      disable-sid rule
    }
  }
}
```

---

### Parameters

<i>rule</i>	<p>Multi-node. A Snort rule to be selectively disabled. The format is [<i>group-id</i>:]<i>sig-id</i>, where <i>group-id</i> is the group ID of a group of Snort rules and <i>sig-id</i> is the signature ID of the rule.</p> <p>You can disable multiple Snort rules by creating multiple <b>disable-sid</b> configuration nodes.</p> <p>Rules in the <b>rules</b> directory have a group ID of 1. Rules in other Snort rule directories have other group IDs; see Snort documentation for this information. If no group ID is specified, a group ID of 1 is assumed.</p>
-------------	--

---

### Default

None.

---

### Usage Guidelines

Use this command to selectively disable a Snort rule in the **rules** directory (`/etc/snort/rules`) or the **preproc\_rules** directory (`/etc/snort/preproc_rules`). The typical use for this command is to disable Snort rules that generate too many false positives. These can be seen in the Snort log files (located in `/var/log/snort`). Refer to [snort.org](http://snort.org) to determine if a rule can safely be disabled in your environment.

Use the **set** form of this command to specify a Snort rule to be selectively disabled.

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

Use the **show** form of this command to display which rules have been selectively disabled.

## content-inspection ips modify-rules enable-sid <rule>

Allows you to selectively enable a Snort rule.

---

### Syntax

```
set content-inspection ips modify-rules enable-sid rule
delete content-inspection ips modify-rules enable-sid rule
show content-inspection ips modify-rules enable-sid
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
content-inspection{
  ips {
    modify-rules {
      enable-sid rule
    }
  }
}
```

---

### Parameters

<i>rule</i>	<p>Multi-node. A Snort rule to be selectively enabled. The format is [<i>group-id</i>]:<i>sig-id</i>, where <i>group-id</i> is the group ID of a group of Snort rules and <i>sig-id</i> is the signature ID of the rule.</p> <p>You can enable multiple Snort rules by creating multiple <b>enable-sid</b> configuration nodes.</p> <p>Rules in the <b>rules</b> directory have a group ID of 1. Rules in other Snort rule directories have other group IDs; see Snort documentation for this information. If no group ID is specified, a group ID of 1 is assumed.</p>
-------------	---

---

### Default

None.

---

## Usage Guidelines

Use this command to selectively enable a Snort rule that is disabled by default in the in the `rules` directory (`/etc/snort/rules`) or the `preproc_rules` directory (`/etc/snort/preproc_rules`).

By default, the Snort rule base includes many rules that are disabled. You can use this command to selectively re-enable these rules. (It is not necessary to use this command to enable rules already enabled in the Snort rule base.)

Snort logs are found in `/var/log/snort`.

Use the `set` form of this command to specify a Snort rule to be selectively enabled.

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

Use the `show` form of this command to display which rules have been selectively enabled.

## content-inspection ips modify-rules exclude-category <category>

Allows you to selectively exclude an entire Snort rule category.

---

### Syntax

```
set content-inspection ips modify-rules exclude-category category
delete content-inspection ips modify-rules exclude-category category
show content-inspection ips modify-rules exclude-category
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
content-inspection{
  ips {
    modify-rules {
      exclude-category category
    }
  }
}
```

---

### Parameters

<i>category</i>	Multi-node. The Snort rule category to be excluded from processing—for example, <b>chat</b> , <b>ddos</b> , <b>backdoor</b> , or <b>scan</b> . You can exclude multiple Snort categories by creating multiple <b>exclude-category</b> configuration nodes.
-----------------	---

---

### Default

None.



---

## Usage Guidelines

Use this command to selectively exclude an entire category of Snort rules. If all the rules in a category are to be disabled, you can exclude the entire category rather than disabling individual rules using their signature IDs.

Snort rules are found in the **rules** directory (`/etc/snort/rules`) or the **preproc\_rules** directory (`/etc/snort/preproc_rules`).

Snort logs are found in `/var/log/snort`.

**NOTE** If a category is both included using `content-inspection ips modify-rules include-category <category>` and excluded using `content-inspection ips modify-rules exclude-category <category>`, the end result is that the category will be included.

Use the **set** form of this command to specify a Snort category to be excluded from processing.

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

Use the **show** form of this command to display which categories have been selectively excluded.

## content-inspection ips modify-rules include-category <category>

Allows you to selectively include an entire Snort rule category.

---

### Syntax

```
set content-inspection ips modify-rules include-category category
delete content-inspection ips modify-rules include-category category
show content-inspection ips modify-rules include-category
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
content-inspection{
  ips {
    modify-rules {
      include-category category
    }
  }
}
```

---

### Parameters

<i>category</i>	Multi-node. The Snort rule category to be included in processing—for example, <b>chat</b> , <b>ddos</b> , <b>backdoor</b> , or <b>scan</b> . You can include multiple Snort categories by creating multiple <b>include-category</b> configuration nodes.
-----------------	---

---

### Default

None.

---

## Usage Guidelines

Use this command to selectively include an entire category of Snort rules. If all the rules in a category are to be enabled, you can include the entire category rather than enabling individual rules using their signature IDs.

Snort rules are found in the **rules** directory (`/etc/snort/rules`) or the **preproc\_rules** directory (`/etc/snort/preproc_rules`).

Snort logs are found in `/var/log/snort`.

**NOTE** If a category is both included using `content-inspection ips modify-rules include-category <category>` and excluded using `content-inspection ips modify-rules exclude-category <category>`, the end result is that the category will be included.

Use the **set** form of this command to specify a Snort category to be included from processing.

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

Use the **show** form of this command to display which categories have been selectively included.

## content-inspection ips modify-rules internal-network <ipv4net>

Allows you to specify an internal network.

---

### Syntax

```
set content-inspection ips modify-rules internal-network ipv4net
delete content-inspection ips modify-rules internal-network ipv4net
show content-inspection ips modify-rules internal-network
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
content-inspection{
  ips {
    modify-rules {
      internal-network ipv4net
    }
  }
}
```

---

### Parameters

<i>ipv4net</i>	Multinode. An IPv4 network representing the internal network. You can define multiple values by creating multiple <b>internal-network</b> configuration nodes.
----------------	--

---

### Default

None.

---

### Usage Guidelines

Use this command to define an internal network so that the IPS engine can distinguish between local and external traffic sources.

Use the **set** form of this command to specify and internal network.

Use the **delete** form of this command to delete an internal network specification.

Use the **show** form of this command to display configured values for internal networks.

## content-inspection ips policy

Allows you to specify a subset of rules based on various policy types.

---

### Syntax

```
set content-inspection ips policy [balanced | connectivity | security]
delete content-inspection ips policy
show content-inspection ips policy
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
content-inspection{
  ips {
    policy [balanced | connectivity | security]
  }
}
```

---

### Parameters

<b>balanced</b>	Provides a balance of enabled rules - between <b>connectivity</b> and <b>security</b> in terms of its security level. This is the default.
<b>connectivity</b>	Provides a more limited set of rules where only the highest priority threats are scanned for. This is used in environments that are latency sensitive (for example, VOIP, financial transactions, etc.).
<b>security</b>	Provides a very high level of security with an associated decrease in system performance. This is used in environments that are extremely security sensitive.

---

### Default

A **balanced** policy is used.

---

### Usage Guidelines

Use this command to define, at a very high level, the subset of rules used to establish a base security policy.

Use the **set** form of this command to specify a subset of rules based on a policy type.

Use the **delete** form of this command to delete the policy type and revert to the default set of rules.

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

## 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 rule
  }
}
```

---

### Parameters

---

<b>preset all</b>	All IPv4 traffic is processed by the IPS.
<b>custom <i>rule</i></b>	Specifies the name of an IPv4 firewall rule set (defined under “firewall name”) 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 Prevention System (IPS) functions.

Traffic is processed by the IPS only when the traffic-filter has been configured and the **ips** configuration node is defined.



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

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

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

## interfaces <interface> content-inspection <direction> enable

Applies content inspection to a traffic stream on an interface.

---

### Syntax

```
set interfaces interface content-inspection {in enable | local enable | out enable}
delete interfaces interface content-inspection [in | local | out]
show interfaces interface content-inspection [in | local | out]
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
interfaces interface {
  content-inspection {
    in {
      enable
    }
    local {
      enable
    }
    out {
      enable
    }
  }
}
```

---

### Parameters

<i>interface</i>	Mandatory. Multi-node. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
<b>in enable</b>	Applies content inspection to inbound through traffic on the specified interface.
<b>local enable</b>	Applies content inspection to traffic arriving on the specified interface and bound for the local system.

---

<b>out enable</b>	Applies content inspection to outbound through traffic on the specified interface.
-------------------	--

---

### Default

None.

### Usage Guidelines

Use this command to apply content inspection to a specific traffic stream on an interface.

The system will apply content inspection to a traffic stream in one of the following ways, depending on what you specify when you apply it:

- **in.** If you **enable** content inspection for traffic coming **in**, packets entering the interface and passing through the system will be inspected.
- **out.** If you **enable** content inspection for traffic going **out**, packets leaving the interface will be inspected.
- **local.** If you **enable** content inspection for **local** traffic, packets entering the interface but destined for the system itself will be inspected.

For each interface, you can apply content inspection to one or more traffic streams (**in**, **out**, and/or **local**).

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	<code>adsl <i>adslx</i> pvc <i>pvc-id</i> bridged-ethernet</code>	<p><i>adslx</i> The name of a Bridged Ethernet- encapsulated DSL interface.</p> <p><i>pvc-id</i> The identifier for the PVC. It can either be the <i>vpi/vci</i> pair or the keyword <b>auto</b>, where <i>vpi</i> is a Virtual Path Index from 0 to 255, <i>vci</i> is a Virtual Circuit Index from from 0 to 65535, and <b>auto</b> directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.</p>
ADSL Classical IPOA	<code>adsl <i>adslx</i> pvc <i>pvc-id</i> classical-ipoa</code>	<p><i>adslx</i> The name of a Classical IPOA- encapsulated DSL interface.</p> <p><i>pvc-id</i> The identifier for the PVC. It can either be the <i>vpi/vci</i> pair or the keyword <b>auto</b>, where <i>vpi</i> is a Virtual Path Index from 0 to 255, <i>vci</i> is a Virtual Circuit Index from from 0 to 65535, and <b>auto</b> directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.</p>

Interface Type	Syntax	Parameters
ADSL PPPoA	<code>adsl <i>adslx</i> pvc <i>pvc-id</i> pppoa <i>num</i></code>	<p><i>adslx</i> The name of a Classical IpoA- encapsulated DSL interface.</p> <p><i>pvc-id</i> The identifier for the PVC. It can either be the <i>vpi/vci</i> pair or the keyword <b>auto</b>, where <i>vpi</i> is a Virtual Path Index from 0 to 255, <i>vci</i> is a Virtual Circuit Index from from 0 to 65535, and <b>auto</b> directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.</p> <p><i>num</i> 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.</p>
ADSL PPPoE	<code>adsl <i>adslx</i> pvc <i>pvc-id</i> pppoe <i>num</i></code>	<p><i>adslx</i> The name of a Classical IpoA- encapsulated DSL interface.</p> <p><i>pvc-id</i> The identifier for the PVC. It can either be the <i>vpi/vci</i> pair or the keyword <b>auto</b>, where <i>vpi</i> is a Virtual Path Index from 0 to 255, <i>vci</i> is a Virtual Circuit Index from from 0 to 65535, and <b>auto</b> directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.</p> <p><i>num</i> The name of a defined PPPoE unit. The range is 0 to 15.</p>
Bonding	<code>bonding <i>bondx</i></code>	<i>bondx</i> The identifier for the bonding interface. Supported values are <b>bond0</b> through <b>bond99</b> .
Bonding Vif	<code>bonding <i>bondx</i> vif <i>vlan-id</i></code>	<p><i>bondx</i> The identifier for the bonding interface. Supported values are <b>bond0</b> through <b>bond99</b>.</p> <p><i>vlan-id</i> The VLAN ID for the vif. The range is 0 to 4094.</p>
Bridge	<code>bridge <i>brx</i></code>	<i>brx</i> The name of a Bridge group. The range is <b>br0</b> through <b>br999</b> .
Ethernet	<code>ethernet <i>ethx</i></code>	<i>ethx</i> The name of an Ethernet interface. The range is <b>eth0</b> through <b>eth23</b> , depending on the physical interfaces available on your system.
Ethernet PPPoE	<code>ethernet <i>ethx</i> pppoe <i>num</i></code>	<p><i>ethx</i> The name of an Ethernet interface. The range is <b>eth0</b> through <b>eth23</b>, depending on the physical interfaces available on your system.</p> <p><i>num</i> The name of a defined PPPoE unit. The range is 0 to 15.</p>
Ethernet Vif	<code>ethernet <i>ethx</i> vif <i>vlan-id</i></code>	<p><i>ethx</i> The name of an Ethernet interface. The range is <b>eth0</b> through <b>eth23</b>, depending on the physical interfaces available on your system.</p> <p><i>vlan-id</i> The VLAN ID for the vif. The range is 0 to 4094.</p>

Interface Type	Syntax	Parameters
Ethernet Vif PPPoE	ethernet <i>ethx</i> vif <i>vlan-id</i> pppoe <i>num</i>	<i>ethx</i> The name of an Ethernet interface. The range is <b>eth0</b> through <b>eth23</b> , depending on the physical interfaces available on your system. <i>vlan-id</i> The VLAN ID for the vif. The range is 0 to 4094. <i>num</i> The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback <i>lo</i>	<i>lo</i> The name of the loopback interface.
Multilink	multilink <i>mlx</i> vif <i>1</i>	<i>mlx</i> The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are <b>ml0</b> (“em ell zero”) through <b>ml23</b> (“em ell twenty-three”). <i>1</i> The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn <i>vtunx</i>	<i>vtunx</i> The identifier for the OpenVPN interface. This may be <b>vtun0</b> to <b>vtunx</b> , where <i>x</i> is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet <i>pethx</i>	<i>pethx</i> The name of a pseudo-Ethernet interface. The range is <b>peth0</b> through <b>peth999</b> .
Serial Cisco HDLC	serial <i>wanx</i> cisco-hdlc vif <i>1</i>	<i>wanx</i> The serial interface you are configuring: one of <b>wan0</b> through <b>wan23</b> . The interface must already have been defined. <i>1</i> The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.
Serial Frame Relay	serial <i>wanx</i> frame-relay vif <i>dlci</i>	<i>wanx</i> The serial interface you are configuring: one of <b>wan0</b> through <b>wan23</b> . The interface must already have been defined. <i>dlci</i> 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.
Serial PPP	serial <i>wanx</i> ppp vif <i>1</i>	<i>wanx</i> The serial interface you are configuring: one of <b>wan0</b> through <b>wan23</b> . The interface must already have been defined. <i>1</i> 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.
Tunnel	tunnel <i>tunx</i>	<i>tunx</i> An identifier for the tunnel interface you are defining. The range is <b>tun0</b> to <b>tun23</b> .

Interface Type	Syntax	Parameters
VRRP	interface <i>parent-if</i> vrrp vrrp-group <i>group</i> interface	<i>parent-if</i> The type and identifier of the parent interface; for example, <b>ethernet eth0</b> or <b>bonding bond0</b> . <i>group</i> The VRRP group identifier. The name of the VRRP interface is not specified. The system internally constructs the interface name from the parent interface identifier plus the VRRP group number—for example, <b>eth0v99</b> , <b>eth0.15v99</b> , <b>bond0v99</b> , or <b>bond0.15v99</b> . Note that VRRP interfaces support the same feature set as the parent interface does.
Wireless	wireless <i>wlanx</i>	<i>wlanx</i> The identifier for the wireless interface you are using. This may be <b>wlan0</b> to <b>wlan999</b> .
Wireless Modem	wirelessmodem <i>wlmx</i>	<i>wlmx</i> The identifier for the wireless modem interface you are using. This may be <b>wlm0</b> to <b>wlm999</b> .

Use the **set** form of this command to apply content inspection to a traffic stream on an interface.

Use the **delete** form of this command to remove content inspection from a traffic stream on an interface.

Use the **show** form of this command to view content inspection configuration for a traffic stream on an interface.

## 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 Prevention System (IPS).

---

### Examples

[Example 1-8](#) shows the first screen of output for `show ips log`.

Example 1-8 “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 rules summary

Displays a summary of the enabled and disabled IPS rules.

---

### Syntax

```
show ips rules summary
```

---

### Command Mode

Operational mode.

---

### Parameters

None.

---

### Default

None.

---

### Usage Guidelines

Use this command to see the number of enabled and disabled rules for each IPS rule classification.

---

### Examples

[Example 1-9](#) shows the output for `show ips rules summary`.

Example 1-9 “show ips rules summary”: Displaying information regarding IPS rules state.

---

```
vyatta@R1:~$ show ips rules summary

Snort Classifications                                     Enabled Disabled

Priority 1
=====
Executable Code was Detected                             37          28
Unsuccessful User Privilege Gain                         7           5
Inappropriate Content was Detected                       0           0
Attempted User Privilege Gain                           1136       2455
Successful User Privilege Gain                           1           0
Attempted Administrator Privilege Gain                   321       3299
A Network Trojan was Detected                            647       474
Web Application Attack                                   23        397
```

Successful Administrator Privilege Gain	1	8
Potential Corporate Privacy Violation	105	196
	-----	-----
	2278	6862
 Priority 2 =====		
Detection of a Denial of Service Attack	15	17
Attempt to Login By a Default Username and Password	0	6
An Attempted Login Using a Suspicious Username was Detected	4	15
Information Leak	287	21
Sensitive Data was Transmitted Across the Network	1	0
Detection of a Non-Standard Protocol or Event	5	8
Attempted Denial of Service	98	175
Denial of Service	0	0
Access to a Potentially Vulnerable Web Application	6	514
Misc Attack	74	115
A Suspicious Filename was Detected	12	28
A System Call was Detected	4	1
Potentially Bad Traffic	109	309
Attempted Information Leak	99	297
A Client was Using an Unusual Port	0	3
Decode of an RPC Query	108	25
Large Scale Information Leak	0	0
	-----	-----
	822	1534
 Priority 3 =====		
Generic Protocol Command Decode	319	3302
Generic ICMP event	0	0
Misc activity	924	328
Unknown Traffic	13	3
A Suspicious String was Detected	4	0
Detection of a Network Scan	3	2
Not Suspicious Traffic	11	14
	-----	-----
	1274	3649
 Priority 4 =====		
A TCP Connection was Detected	3	0
	-----	-----
	3	0
 Total	=====	=====
	4377	12045

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

---

## 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 Prevention System (IPS) alerts.

---

### Examples

[Example 1-10](#) shows the output for `show ips summary`.

Example 1-10 “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 Prevention System (IPS) rules updates.

Note that the first time an update is run the IPS system takes a few minutes to update the snort rules and the log file is not created until the first update is complete - so running this command prior to the first update completing could produce a “log not found” error.

---

### Examples

[Example 1-11](#) shows the output for `show ips update-log`.

Example 1-11 “show ips update-log”: Displaying ips rules update history

---

```
vyatta@R1:~$ show ips update-log
2010-06-21-144916: Starting Snort VRT subscription update [manual]
2010-06-21-144943: New subscription rules downloaded.
2010-06-21-144943: Starting unpack & processing new rules.
2010-06-21-144943: IPS rules update completed successfully
2010-06-21-144943: Original rules are moved to
/opt/vyatta/etc/ips/backup/snort.2010-06-21-144943
vyatta@R1:~$
```

---

## update ips rules

Updates the IPS rules base.

---

### Syntax

```
update ips rules [oinkcode oinkcode | snortvrt-subscription]
```

---

### Command Mode

Operational mode.

---

### Parameters

---

<b>oinkcode</b> <i>oinkcode</i>	The “oinkcode” from snort.org. Indicates that the user is a registered snort user (registered for the free service).
<b>snortvrt-subscription</b>	Indicates that the user is a Vyatta PLUS Snort VRT Service subscriber.

---

---

### Default

The command will use information configured via either [content-inspection ips auto-update oink-code <code>](#) or [content-inspection ips auto-update snortvrt-subscription](#) to determine which IPS rules to download.

---

### Usage Guidelines

Use this command to manually update the IPS rules base rather than waiting for the automatic update hour to arrive. Note that this process will run in the background and can take quite a long time to complete. The [show ips update-log](#) command can be used to view the progress/results of the update.

No parameters are required if the system has been configured to automatically update IPS rules via either [content-inspection ips auto-update oink-code <code>](#) (snort.org registered users) or [content-inspection ips auto-update snortvrt-subscription](#) (Vyatta PLUS Snort VRT Service users).

If the system has not been configured to automatically update IPS rules then parameters are required. Snort.org registered users must specify the “oink code” they received when they registered with snort.org. Vyatta PLUS Snort VRT Service users must include the *snortvrt-subscription* parameter and must have already configured the system with an entitlement key and a service key for this service.

---

## Examples

[Example 1-12](#) shows the output for **update ips rules** where the user has configured their system to use the Vyatta PLUS Snort VRT Service.

Example 1-12 “update ips rules”: Displaying output from a manual rules update

---

```
vyatta@R1:~$ update ips rules
Starting Snort VRT subscription update [manual]
19035189/19035189 - 100%
New subscription rules downloaded.

Starting unpack & processing of new rules.
vyatta@R1:~$
```

---



## zone-policy zone <to-zone> from <from-zone> content-inspection enable

Applies content inspection to traffic arriving at the “to” zone from the “from” zone.

---

### Syntax

```
set zone-policy zone to-zone from from-zone content-inspection enable
delete zone-policy zone to-zone from from-zone content-inspection
show zone-policy zone to-zone from from-zone content-inspection
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
zone-policy zone to-zone {
    from from-zone {
        content-inspection {
            enable
        }
    }
}
```

---

### Parameters

<i>to-zone</i>	The zone the traffic is destined for.
<i>from-zone</i>	The zone the traffic is coming from.

---

### Default

None.

---

### Usage Guidelines

Use this command to apply content inspection to traffic arriving at the “to” zone from the “from” zone.

Use the set form of this command to specify that content inspection is to be applied to all packets arriving at the “to” zone from the “from” zone.

Use the **delete** form of this command to remove content inspection from traffic destined for the “to” zone from the “from” zone.

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

## Chapter 2: Web Filtering

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

This chapter presents the following topics:

- [Web Filtering Configuration](#)
- [Web Filtering Commands](#)

# Web Filtering Configuration

---

This section presents the following topics:

- [Introduction](#)
- [Web Filtering Overview](#)
- [Order of Evaluation](#)
- [Web Filtering Configuration Examples](#)

## Introduction

The Vyatta system can be configured to act as a web proxy server providing both web caching and web filtering functionality. To learn more about using the Vyatta system for web caching, please see the *Vyatta Services Reference Guide*.

Web filtering (also called URL filtering) is an important tool for managing web access to reduce exposure to web-based threats, to limit legal liabilities by blocking objectionable content, to increase productivity, and to manage bandwidth usage. The Vyatta system provides basic web filtering services as part of the Vyatta Core. VyattaGuard enhanced web filtering is available as a Vyatta Plus service.



Web filtering is available as part of the Vyatta Core system, providing access to a list of filtering categories in a community-updated “blacklist.”



VyattaGuard advanced web filtering is available as a subscription-based Vyatta Plus service offering which includes expanded content categorization (50+) and a continually updated database containing 350 million+ categorized URLs, as well as the ability to create customized whitelists and blacklists with time and date controls.

Key features of VyattaGuard include the following:

- 50+ content categories
- 350+ million classified URLs
- Classification of HTTPS traffic in transparent mode using IP address classification
- Dynamic classification of new URLs
- Best feed of compromised, malicious, and phishing URLs
- Near real-time blocking of newly classified and malicious sites
- Broad international coverage
- International domain name support includes domains with special characters
- Blend of automatic and human classification
- Fast response time in classifying new sites

- Granular URL classifications support specific pages, paths, subdomains, and parent domains
- Multiple categories per URL
- Reputation-based filtering services

**NOTE** The VyattaGuard database is considerably larger than the standard database. For this reason systems, that use VyattaGuard require an additional 2 GB of disk space over the standard disk space requirements.

## Web Filtering Overview

This section presents the following topics:

- [Transparent Mode](#)
- [Non-Transparent Mode](#)
- [Transparent Mode for HTTPS Traffic](#)

The Vyatta system's web proxy provides web filtering functionality. It receives client requests to access remote web sites (URLs) and passes these requests to the web filtering process. The web filtering process in turn queries a categorization database; this can be either the standard database (provided with the Vyatta Core system) or the enhanced VyattaGuard database (available as a Vyatta Plus service) to determine into which category the URL falls. Based on the way you have configured the web filtering service, the request is either allowed to continue to its destination or is redirected to a redirection site, which signals the user that the request has been blocked.

The web proxy can operate in one of two modes:

- Transparent mode
- Non-transparent mode.

## Transparent Mode

In transparent mode, the web clients making the web requests (that is, the browser on the user's machine) is not configured in any special way.

For HTTP traffic, the web proxy process intercepts port 80 traffic and either blocks or allows the requests based on web filtering configuration within the Vyatta system. In transparent mode, the Vyatta system listens for web requests on a specific "listen address," which is configured using the **service webproxy listen-address** command; in the example below, the listen address is 172.16.139.200.

---

```
set service webproxy listen-address 172.16.139.200
```

---

The web proxy's capabilities are different for HTTPS traffic than for HTTP traffic. The proxy can intercept the HTTPS packets, but because they are encrypted (and the web proxy does not have the decryption key, since it is not the intended destination for the packets) it cannot "see" anything inside the packets other than the destination address. For HTTPS traffic, you must either use non-transparent mode or use transparent mode with limitations (as described in ["Transparent Mode for HTTPS Traffic" on page 79](#)).

## Non-Transparent Mode

In non-transparent mode, web clients (browsers) that make the web requests are configured explicitly to forward requests to the web proxy (e.g. send port 80 traffic to 172.16.139.200:3128, or, for HTTPS, send port 443 traffic to 172.16.139.200:3128). In this case the web proxy does not have to intercept the traffic as it is sent to it directly. The web proxy process then either blocks or allows the requests based on the web filtering configuration within the Vyatta system. The Vyatta system is configured to listen for web requests on a specific address in non-transparent mode as follows:

```
set service webproxy listen-address 172.16.139.200 disable-transparent
```

where 172.16.139.200 is the address that the web proxy service will listen for web requests on.

Because it is not necessary to intercept traffic in non-transparent mode, both HTTP and HTTPS traffic can be sent to the web proxy. For HTTPS traffic, the client establishes a TLS (encrypted) session directly with the web proxy and (given the request is allowed by web filtering) the web proxy establishes a separate TLS session with the destination web server on behalf of the client. The web proxy receives the encrypted request from the client, decrypts it, uses web filtering to determine that it is allowed, re-encrypts it (using different keys), then forwards it to the destination. Responses from the destination are decrypted by the web proxy then re-encrypted (using different keys) before being passed to the client.

## Transparent Mode for HTTPS Traffic



This capability is available with VyattaGuard advanced web filtering.

The three cases that we have seen thus far are as follows:

- 1 Transparent mode and HTTP (port 80) traffic
- 2 Non-transparent mode and HTTP (port 80) traffic
- 3 Non-transparent mode and HTTPS (port 443) traffic

In the first case the web proxy intercepts the client requests and, based on the web filtering configuration, determines whether to pass the requests along or redirect them to a redirection URL. In the second two cases, the client establishes a connection directly with the web proxy so there is no need for the web proxy to intercept the traffic. Both HTTP and HTTPS requests can be categorized by web filtering and then a determination can be made to either allow or block access to the requested URL.

The remaining case is transparent mode and HTTPS (port 443) traffic. This case is an issue because the client is establishing an encrypted session with a remote destination, not with the web proxy. Because of this the web proxy cannot simply intercept the port 443 traffic, extract the URL and pass it to web filtering because the URL is encrypted and the web proxy does not have the key to decrypt it.

This problem can be circumvented using the IP address based categorization feature available only as part of VyattaGuard advanced web filtering. With this capability, the IP address of the destination is passed to web filtering and VyattaGuard's IP address based categorization determines how the request should be categorized. Based on the categorization, web filtering determines whether or not to allow the request according to the web filtering configuration.

It should be noted that the HTTPS connection is not proxied using web proxy as it is in non-transparent mode; it is simply filtered using web filter. The result of this is that requests can only be allowed or dropped, they cannot be redirected.

Also, because the web filtering process does not have access to the URL (as it is encrypted), web filtering commands that require the URL (e.g., **local-block-keyword**) will not work for HTTPS filtering in transparent mode. Finally, be aware that with commands like **local-block** and **local-ok**, if an IP address is specified explicitly, the filter will fail to work as expected if the IP address of the destination web site changes.

[Table 2-1](#) shows commands that may not work as expected for HTTPS filtering in transparent mode.

Table 2-1 Commands Operating with Limitations for HTTPS in Transparent Mode

<b>service webproxy</b>	<b>Reason</b>
domain block	Cannot filter: requires the URL.
<b>service webproxy url-filtering squiguard</b>	<b>Reason</b>
enable-safe-search	Does not work as intended: requires a rewrite of the search URL
local-block-keyword	Cannot filter: requires the URL.
local-block-url	Cannot filter: requires the URL.
local-ok-url	Cannot filter: requires the URL.

Table 2-1 Commands Operating with Limitations for HTTPS in Transparent Mode

redirect-url	Does not work: no web proxy is involved in sending a redirect URL
<b>service webproxy url-filtering squidguard rule &lt;rule&gt;</b>	<b>Reason</b>
enable-safe-search	Does not work as intended: requires a rewrite of the search URL
local-block-keyword	Cannot filter: requires the URL.
local-block-url	Cannot filter: requires the URL.
local-ok-url	Cannot filter: requires the URL.
redirect-url	Does not work: no web proxy is involved in sending a redirect URL

**NOTE** Commands under **webproxy** that are not specified are not affected by HTTPS filtering in transparent mode. They work the same for both HTTP and HTTPS filtering.

In order to configure HTTPS filtering in transparent mode the following configurations must be made:

- 1 Configure the **allow-ipaddr-url** option globally. For example:  

```
set service webproxy url-filtering squidguard allow-ipaddr-url
```
- 2 Configure the **allow-ipaddr-url** option within each rule specified. For example:  

```
set service webproxy url-filtering squidguard rule 1 allow-ipaddr-url
```
- 3 Configure the **vyattaguard** option. For example:  

```
set service webproxy url-filtering squidguard vyattaguard mode normal
```
- 4 Configure the **vyattaguard https-filter** option. For example:  

```
set service webproxy url-filtering squidguard vyattaguard https-filter
```

## Order of Evaluation

To understand the results of web filtering for both HTTP and HTTPS requests, it is important to keep in mind that the various web filtering configuration commands are evaluated in a specific order. All filters contained within rules are evaluated first (in rule number order) followed by all global filters (that is, those not contained within a rule). The filters, either within a rule or global, are evaluated in the following order:

- 1) local-ok
- 2) local-block



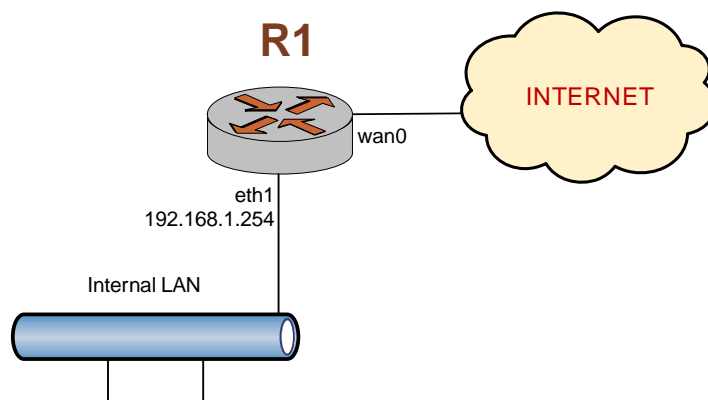
- 3) allow-ipaddr-url
- 4) block-category
- 5) allow-category
- 6) local-block-keyword
- 7) default-action

## Web Filtering Configuration Examples

Figure 2-1 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 (in transparent mode) to provide web filtering functionality to employees accessing the Internet.

Figure 2-1 Web proxy



This section presents the following examples:

- Example 2-1 Blocking specific URLs
- Example 2-2 Verifying filtering
- Example 2-3 Filtering by content category
- Example 2-4 Filtering by keywords
- Example 2-5 Allowing specific sites
- Example 2-6 Redirecting users
- Example 2-7 Handling different groups of users
- Example 2-8 Handling different time periods.

- Example 2-9 Handling HTTPS filtering in transparent mode.
- Example 2-10 Creating a whitelist.

## Blocking Specific URLs

[Example 2-1](#) blocks specific URLs by explicitly specifying them using the **local-block** option, rather than by downloading and setting up a filter list. To block specific URLs on the Vyatta system, perform the following steps:

Example 2-1 Blocking specific URLs

Step	Command
Set the address to listen for requests on.	<code>vyatta@R1# set service webproxy listen-address 192.168.1.254</code>
Deny requests for the YouTube web site.	<code>vyatta@R1# set service webproxy url-filtering squidguard local-block youtube.com</code>
Deny requests for the Facebook web site.	<code>vyatta@R1# set service webproxy url-filtering squidguard local-block facebook.com</code>
Commit the change	<code>vyatta@R1# commit</code>
Show the updated web proxy-related configuration.	<code>vyatta@R1# show service webproxy listen-address 192.168.1.254 { } url-filtering {   squidguard {     local-block youtube.com     local-block facebook.com   } }</code>

## 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 2-2](#) enables logging for locally blocked URLs. To log web proxy functions in this way, perform the following steps:

## Example 2-2 Verifying filtering

Step	Command
Set the web proxy to log everything filtered by the “local-block” option.	<code>vyatta@R1# set service webproxy url-filtering squidguard log local-block</code>
Commit the change	<code>vyatta@R1# commit</code>
Show the updated web proxy-related configuration.	<code>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   } }</code>

## Filtering by Content Category

[Example 2-3](#) uses a downloaded squidGuard database (downloaded using [update webproxy blacklists](#)) to filter web contents by content category. Customers using the Vyatta Plus advanced web filtering service, VyattaGuard, would download a separate database (using [update webproxy vyattaguard](#)). 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 2-3 Filtering by content category

Step	Command
Block the ads category	<code>vyatta@R1# set service webproxy url-filtering squidguard block-category ads</code>
Block the spyware category	<code>vyatta@R1# set service webproxy url-filtering squidguard block-category spyware</code>
Block the gambling category	<code>vyatta@R1# set service webproxy url-filtering squidguard block-category gambling</code>
Commit the change	<code>vyatta@R1# commit</code>

**Example 2-3** Filtering by content category

---

Show the updated web proxy-related configuration.	<pre> 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   } } </pre>
---	--

---

## Filtering by Keyword

[Example 2-4](#) uses keyword filtering to block access to sites that match a specific string of characters. In this example, access to all Chinese sites are blocked. To configure the web proxy in this way, perform the following steps:

**Example 2-4** Filtering by keywords

Step	Command
Block access to Chinese sites.	<pre>vyatta@R1# set service webproxy url-filtering squidguard local-block-keyword ".cn"</pre>
Commit the change	<pre>vyatta@R1# commit</pre>
Show the updated web proxy-related configuration.	<pre> 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-block-keyword .cn     log local-block   } } </pre>

---

## Allowing Specific Sites

[Example 2-5](#) 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 2-5 Allowing specific sites

Step	Command
Allow users to access www.company-ads.com	<code>vyatta@R1# set service webproxy url-filtering squidguard local-ok www.company-ads.com</code>
Commit the change	<code>vyatta@R1# commit</code>
Show the updated web proxy-related configuration.	<code>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-block-keyword .cn     local-ok www.company-ads.com     log local-block   } }</code>

## Redirecting Users

By default, a user who tries to access a blocked site is sent to a pre-defined redirect site. The redirect site can be changed using the **redirect-url** command; another option is to display the reason (category) the requested URL was blocked.

[Example 2-6](#) directs the system to display the category and URL of a blocked site when an access attempt is made by a user. To configure the web proxy in this way, perform the following steps:

## Example 2-6 Redirecting users

Step	Command
<p>Specify an HTTP query. The query shown in the example retrieves a squidGuard script that displays a blocked URL and the reason for blocking it.</p> <p>(Note the case in the query; HTTP queries are case-sensitive.)</p> <p>(Note that in order to enter a “?” in the URL string it must be preceded by a &lt;Ctrl&gt;+v or it will be interpreted as a command completion character. Another option is to use the “set terminal key query-help disable” command to disable the CLI from interpreting the “?” as a command completion character.)</p>	<pre>vyatta@R1# set service webproxy url-filtering squidguard redirect-url "http://192.168.1.254/cgi-bin/squidGuard-simple.cgi?targetclass=%t&amp;url=%u"</pre>
Commit the change.	<pre>vyatta@R1# commit</pre>
Show the updated webproxy-related configuration.	<pre>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-block-keyword .cn     local-ok www.company-ads.com     log local-block     redirect-url     "http://192.168.1.254/cgi-bin/squidGuard-simple.cgi?targetclass=%t&amp;url=%u"   } }</pre>

## Handling Different Groups of Users

To this point the examples assumed that all users would be treated equally. In the real world, some users need to be handled differently than others. The **source-group** command provides the ability to segregate users based on their specific IP address or the subnet that they are on.

[Example 2-7](#) assumes the same network diagram as above (Example 2-1 Web proxy) but this time it is configured to address the needs of a school where system administrators, teachers, and students are each treated differently.

Example 2-7 Handling different groups of users

Step	Command
Remove the previous configuration.	<code>vyatta@R1# delete service webproxy url-filtering</code>
Commit the change.	<code>vyatta@R1# commit</code>
Redirect blocked requests to google.com	<code>vyatta@R1# set service webproxy url-filtering squidguard redirect-url "http://google.com"</code>
Create the administrators group (a single address).	<code>vyatta@R1# set service webproxy url-filtering squidguard source-group ADMIN address 10.0.5.15</code>
Create the teachers group (a single subnet).	<code>vyatta@R1# set service webproxy url-filtering squidguard source-group TEACHERS address 10.0.5.0/24</code>
Create the students group (the first of two subnets).	<code>vyatta@R1# set service webproxy url-filtering squidguard source-group STUDENTS address 10.0.1.0/24</code>
Create the students group (the second of two subnets).	<code>vyatta@R1# set service webproxy url-filtering squidguard source-group STUDENTS address 10.0.2.0/24</code>
Create the rule to filter requests from the ADMIN group. In this case nothing gets filtered.	<code>vyatta@R1# set service webproxy url-filtering squidguard rule 10 source-group ADMIN</code>
Create the rule to filter requests from the TEACHERS group.	<code>vyatta@R1# set service webproxy url-filtering squidguard rule 20 source-group TEACHERS</code>
Block the "porn" category.	<code>vyatta@R1# set service webproxy url-filtering squidguard rule 20 block-category porn</code>
Block the "shopping" category.	<code>vyatta@R1# set service webproxy url-filtering squidguard rule 20 block-category shopping</code>
Create the rule to filter requests from the STUDENTS group.	<code>vyatta@R1# set service webproxy url-filtering squidguard rule 30 source-group STUDENTS</code>

---

**Example 2-7 Handling different groups of users**

---

Block the “adult” category.	<code>vyatta@R1# set service webproxy url-filtering squidguard rule 30 block-category adult</code>
Block the “warez” category.	<code>vyatta@R1# set service webproxy url-filtering squidguard rule 30 block-category warez</code>
Block the “drugs” category.	<code>vyatta@R1# set service webproxy url-filtering squidguard rule 30 block-category drugs</code>
Block the “filehosting” category.	<code>vyatta@R1# set service webproxy url-filtering squidguard rule 30 block-category filehosting</code>
Block the “audio-video” category.	<code>vyatta@R1# set service webproxy url-filtering squidguard rule 30 block-category audio-video</code>
Commit the change.	<code>vyatta@R1# commit</code>

---



---

**Example 2-7** Handling different groups of users

---

```
Show the new webproxy-related configuration.  vyatta@R1# show service webproxy
listen-address 192.168.1.254 {
}
url-filtering {
  squidguard {
    redirect-url http://google.com
    rule 10 {
      source-group ADMIN
    }
    rule 20 {
      block-category porn
      block-category shopping
      source-group TEACHERS
    }
    rule 30 {
      block-category adult
      block-category audio-video
      block-category drugs
      block-category filehosting
      block-category warez
      source-group STUDENTS
    }
    source-group ADMIN {
      address 10.0.5.15
    }
    source-group STUDENTS {
      address 10.0.1.0/24
      address 10.0.2.0/24
    }
    source-group TEACHERS {
      address 10.0.5.0/24
    }
  }
}
```

---

## Handling Different Time Periods

In the previous example the filtering rules applied at all times. In order to change the filtering for a group based on the day of the week and the time of day the **time-period** command is used.

[Example 2-8](#) shows how to change the filtering based on time period. In this case, the teachers will be allowed to access sites in the “shopping” category but only during non-school hours.

## Example 2-8 Handling different time periods.

<b>Step</b>	<b>Command</b>
Define the SCHOOLHOURS time period.	<code>vyatta@R1# set service webproxy url-filtering squidguard time-period SCHOOLHOURS days weekdays time "09:00-12:00, 13:00-16:00"</code>
Create a new rule to filter requests from the TEACHERS group.	<code>vyatta@R1# set service webproxy url-filtering squidguard rule 25 source-group TEACHERS</code>
Block only the "porn" category.	<code>vyatta@R1# set service webproxy url-filtering squidguard rule 25 block-category porn</code>
Apply the more restrictive rule (rule 20) to the teachers during school hours.	<code>vyatta@R1# set service webproxy url-filtering squidguard rule 20 time-period SCHOOLHOURS</code>
Apply the less restrictive rule (rule 25) to the teachers during non-school hours (using "!" to negate the time period).	<code>vyatta@R1# set service webproxy url-filtering squidguard rule 25 time-period !SCHOOLHOURS</code>
Commit the change.	<code>vyatta@R1# commit</code>

---

**Example 2-8** Handling different time periods.

---

```
Show the new webproxy-related configuration.  vyatta@R1# show service webproxy
listen-address 192.168.1.254 {
}
url-filtering {
  squidguard {
    redirect-url http://google.com
    rule 10 {
      source-group ADMIN
    }
    rule 20 {
      block-category porn
      block-category shopping
      source-group TEACHERS
      time-period SCHOOLHOURS
    }
    rule 25 {
      block-category porn
      source-group TEACHERS
      time-period !SCHOOLHOURS
    }
    rule 30 {
      block-category adult
      block-category audio-video
      block-category drugs
      block-category filehosting
      block-category warez
      source-group STUDENTS
    }
    source-group ADMIN {
      address 10.0.5.15
    }
    source-group STUDENTS {
      address 10.0.1.0/24
      address 10.0.2.0/24
    }
    source-group TEACHERS {
      address 10.0.5.0/24
    }
    time-period SCHOOLHOURS {
      days weekdays {
        time "09:00-12:00, 13:00-16:00"
      }
    }
  }
}
```

---

## Transparent Mode HTTPS Filtering

Filtering HTTPS traffic in transparent mode is a special case. The following example, the configuration required to provide HTTPS filtering in transparent mode are added.

[Example 2-9](#) shows a configuration using HTTPS filtering in transparent mode.

Example 2-9 Handling HTTPS filtering in transparent mode.

Step	Command
Set global IP address filtering.	<pre>vyatta@R1# set service webproxy url-filtering squidguard allow-ipaddr-url</pre>
Set IP address filtering for each of the rules.	<pre>vyatta@R1# set service webproxy url-filtering squidguard rule 10 allow-ipaddr-url vyatta@R1# set service webproxy url-filtering squidguard rule 20 allow-ipaddr-url vyatta@R1# set service webproxy url-filtering squidguard rule 25 allow-ipaddr-url vyatta@R1# set service webproxy url-filtering squidguard rule 30 allow-ipaddr-url</pre>
Specify that VyattaGuard web filtering is to be used.	<pre>vyatta@R1# set service webproxy url-filtering squidguard vyattaguard mode normal</pre>
Specify that HTTPS filtering is to be performed.	<pre>vyatta@R1# set service webproxy url-filtering squidguard vyattaguard https-filter</pre>
Commit the change.	<pre>vyatta@R1# commit</pre>

---

**Example 2-9** Handling HTTPS filtering in transparent mode.

---

```
Show the new webproxy-related configuration.  vyatta@R1# show service webproxy
listen-address 192.168.1.254 {
}
url-filtering {
  squidguard {
    allow-ipaddr-url
    redirect-url http://google.com
    rule 10 {
      allow-ipaddr-url
      source-group ADMIN
    }
    rule 20 {
      allow-ipaddr-url
      block-category porn
      block-category shopping
      source-group TEACHERS
      time-period SCHOOLHOURS
    }
    rule 25 {
      allow-ipaddr-url
      block-category porn
      source-group TEACHERS
      time-period !SCHOOLHOURS
    }
    rule 30 {
      allow-ipaddr-url
      block-category adult
      block-category audio-video
      block-category drugs
      block-category filehosting
      block-category warez
      source-group STUDENTS
    }
    source-group ADMIN {
      address 10.0.5.15
    }
    source-group STUDENTS {
      address 10.0.1.0/24
      address 10.0.2.0/24
    }
    source-group TEACHERS {
      address 10.0.5.0/24
    }
  }
}
```

---

Example 2-9 Handling HTTPS filtering in transparent mode.

```

        time-period SCHOOLHOURS {
            days weekdays {
                time "09:00-12:00, 13:00-16:00"
            }
        }
        vyattaguard {
            https-filter
            mode normal
        }
    }
}

```

## Creating a Whitelist

The typical usage of web filtering is to allow access to all sites except those that are blocked using the various blocking filters. There are instances where the general case is to block access to all sites except a chosen few - a “whitelist”.

[Example 2-10](#) shows how to create a whitelist.

Example 2-10 Creating a whitelist.

Step	Command
Remove the previous configuration.	<code>vyatta@R1# delete service webproxy url-filtering</code>
Commit the change.	<code>vyatta@R1# commit</code>
Redirect blocked requests to google.com.	<code>vyatta@R1# set service webproxy url-filtering squidguard redirect-url "http://google.com"</code>
Set the default action to block access to all sites.	<code>vyatta@R1# set service webproxy url-filtering squidguard default-action block</code>
Allow access to “vyatta.com”.	<code>vyatta@R1# set service webproxy url-filtering squidguard local-ok vyatta.com</code>
Allow access to “vyatta.org”.	<code>vyatta@R1# set service webproxy url-filtering squidguard local-ok vyatta.org</code>
Allow access to “google.com”.	<code>vyatta@R1# set service webproxy url-filtering squidguard local-ok google.com</code>
Commit the change.	<code>vyatta@R1# commit</code>

---

**Example 2-10** Creating a whitelist.

---

Show the new webproxy-related configuration.	<pre>vyatta@R1# show service webproxy listen-address 192.168.1.254 { } url-filtering {   squidguard {     default-action block     local-ok google.com     local-ok vyatta.com     local-ok vyatta.org     redirect-url http://google.com   } }</pre>
--	---

---

# Web Filtering Commands

This chapter contains the following commands.

Configuration Commands	
<code>service webproxy append-domain &lt;domain&gt;</code>	Appends a domain name to URLs not containing a dot.
<code>service webproxy default-port &lt;port&gt;</code>	Sets the default port for web proxy listen addresses.
<code>service webproxy domain-block &lt;domain&gt;</code>	Specifies a domain to block.
<code>service webproxy domain-noncache &lt;domain&gt;</code>	Specifies a domain that is not to be cached.
<code>service webproxy listen-address &lt;ipv4&gt;</code>	Specifies a web proxy listening address.
<code>service webproxy listen-address &lt;ipv4&gt; disable-transparent</code>	Disables web proxy transparent mode at a listening address.
<code>service webproxy listen-address &lt;ipv4&gt; port &lt;port&gt;</code>	Sets the listening port for a listening address.
<code>service webproxy proxy-bypass &lt;address&gt;</code>	Specifies an address or subnet to bypass the webproxy service.
<code>service webproxy reply-block-mime &lt;mime-type&gt;</code>	Specifies a mime type to block.
<code>service webproxy reply-body-max-size &lt;size&gt;</code>	Specifies the maximum size of a reply body.
<code>service webproxy url-filtering disable</code>	Disables web filtering without discarding configuration.
<code>service webproxy url-filtering squidguard</code>	Blocks URLs in all categories.
<code>service webproxy url-filtering squidguard allow-category &lt;category&gt;</code>	Allows web content by squidGuard database category.
<code>service webproxy url-filtering squidguard allow-ipaddr-url</code>	Specifies that direct IP address requests should be allowed.
<code>service webproxy url-filtering squidguard auto-update update-hour &lt;hour&gt;</code>	Sets the hour of the day at which to check for squidGuard database updates.
<code>service webproxy url-filtering squidguard block-category &lt;category&gt;</code>	Blocks web content by squidGuard database category.
<code>service webproxy url-filtering squidguard default-action &lt;action&gt;</code>	Specifies the default action to take for all traffic passing through the webproxy.
<code>service webproxy url-filtering squidguard enable-safe-search</code>	Enables Safe Search on many popular search engines.



<code>service webproxy url-filtering squidguard local-block &lt;address&gt;</code>	Specifies an IP address or domain to be blocked.
<code>service webproxy url-filtering squidguard local-block-keyword &lt;keyword&gt;</code>	Specifies a URL substring within a URL to be blocked.
<code>service webproxy url-filtering squidguard local-block-url &lt;url&gt;</code>	Specifies a URL to be blocked.
<code>service webproxy url-filtering squidguard local-ok &lt;address&gt;</code>	Specifies an IP address or domain to allow.
<code>service webproxy url-filtering squidguard local-ok-url &lt;url&gt;</code>	Specifies a URL to be allowed.
<code>service webproxy url-filtering squidguard log &lt;category&gt;</code>	Enables logging for a squidGuard database category.
<code>service webproxy url-filtering squidguard redirect-url &lt;url&gt;</code>	Specifies a URL to redirect users to when a blacklisted URL is requested.
<code>service webproxy url-filtering squidguard rule &lt;rule-num&gt;</code>	Specifies a web filtering rule.
<code>service webproxy url-filtering squidguard rule &lt;rule-num&gt; allow-category &lt;category&gt;</code>	Allows web content by squidGuard database category within the rule.
<code>service webproxy url-filtering squidguard rule &lt;rule-num&gt; allow-ipaddr-url</code>	Specifies that direct IP address requests should be allowed within the rule.
<code>service webproxy url-filtering squidguard rule &lt;rule-num&gt; block-category &lt;category&gt;</code>	Blocks web content by squidGuard database category within the rule.
<code>service webproxy url-filtering squidguard rule &lt;rule-num&gt; default-action &lt;action&gt;</code>	Specifies the default action to take for traffic within the rule.
<code>service webproxy url-filtering squidguard rule &lt;rule-num&gt; description &lt;desc&gt;</code>	Specifies a brief description for a web filtering policy rule.
<code>service webproxy url-filtering squidguard rule &lt;rule-num&gt; enable-safe-search</code>	Enables Safe Search on many popular search engines for a web filtering policy rule.
<code>service webproxy url-filtering squidguard rule &lt;rule-num&gt; local-block &lt;address&gt;</code>	Defines a specific IP address or URL to be blocked within the rule.
<code>service webproxy url-filtering squidguard rule &lt;rule-num&gt; local-block-keyword &lt;keyword&gt;</code>	Defines a URL substring within a URL to be blocked within the rule.
<code>service webproxy url-filtering squidguard rule &lt;rule-num&gt; local-block-url &lt;url&gt;</code>	Specifies a URL to be blocked within the rule.
<code>service webproxy url-filtering squidguard rule &lt;rule-num&gt; local-ok &lt;address&gt;</code>	Specifies an IP address or URL to allow within the rule.
<code>service webproxy url-filtering squidguard rule &lt;rule-num&gt; local-ok-url &lt;url&gt;</code>	Specifies a URL to be allowed within the rule.

<code>service webproxy url-filtering squidguard rule &lt;rule-num&gt; log &lt;category&gt;</code>	Enables logging for a squidGuard database category within the rule.
<code>service webproxy url-filtering squidguard rule &lt;rule-num&gt; redirect-url &lt;url&gt;</code>	Specifies a URL to redirect users to when a blacklisted URL is requested within the rule.
<code>service webproxy url-filtering squidguard rule &lt;rule-num&gt; source-group &lt;group-name&gt;</code>	Specifies the source group to be used for the web filtering rule.
<code>service webproxy url-filtering squidguard rule &lt;rule-num&gt; time-period &lt;period-name&gt;</code>	Specifies the time period to be used for the web filtering rule.
<code>service webproxy url-filtering squidguard source-group &lt;group-name&gt;</code>	Specifies a web filtering source group.
<code>service webproxy url-filtering squidguard source-group &lt;group-name&gt; address &lt;addr&gt;</code>	Specifies an IPv4 address or subnet to include in the web filtering source group.
<code>service webproxy url-filtering squidguard source-group &lt;group-name&gt; description &lt;desc&gt;</code>	Specifies a brief description for a web filtering source group.
<code>service webproxy url-filtering squidguard source-group &lt;group-name&gt; domain &lt;domain&gt;</code>	Specifies a domain to include in a web filtering source group.
<code>service webproxy url-filtering squidguard time-period &lt;period-name&gt;</code>	Specifies a time period to be used in a web filtering rule.
<code>service webproxy url-filtering squidguard time-period &lt;period-name&gt; days &lt;day&gt; time &lt;time&gt;</code>	Specifies a day and time included in the time period.
<code>service webproxy url-filtering squidguard time-period &lt;period-name&gt; description &lt;desc&gt;</code>	Specifies a brief description for the time period.
<code>service webproxy url-filtering squidguard vyattaguard https-filter</code>	Specifies that transparent mode HTTPS filtering is to be used.
<code>service webproxy url-filtering squidguard vyattaguard mode</code>	Specifies how the VyattaGuard system will operate.
<b>Operational Commands</b>	
<code>restart webproxy</code>	Restarts the webproxy service.
<code>show webproxy blacklist categories</code>	Displays all categories defined in the installed squidGuard database.
<code>show webproxy blacklist domains</code>	Displays all domains listed in the installed database.
<code>show webproxy blacklist log</code>	Displays the log for blacklisted URLs.
<code>show webproxy blacklist search &lt;filter&gt;</code>	Displays domains and/or URLs matching search text.
<code>show webproxy blacklist urls</code>	Displays all URLs in squidGuard database categories.
<code>show webproxy log</code>	Displays the web proxy log.

---

<code>show webproxy vyatguard categories</code>	Displays all categories defined in the installed vyatguard database.
<code>show webproxy vyatguard search &lt;filter&gt;</code>	Displays domains and/or URLs in the vyatguard database matching search text.
<code>update webproxy blacklists</code>	Updates the squidGuard database.
<code>update webproxy vyatguard</code>	Updates the vyatguard database.

---

## restart webproxy

Restarts the webproxy service.

---

### Syntax

```
restart webproxy
```

---

### Command Mode

Operational mode.

---

### Parameters

None.

---

### Usage Guidelines

Use this command to restart the webproxy service.

---

### Examples

[Example 2-11](#) displays output for `restart webproxy`.

Example 2-11 Restarting the webproxy service

---

```
vyatta@R1> restart webproxy
Restarting Squid HTTP Proxy 3.0: squid3 Waiting.....done.
.
vyatta@R1>
```

## service webproxy append-domain <domain>

Appends a domain name to URLs not containing a dot.

---

### Syntax

```
set service webproxy append-domain domain
delete service webproxy append-domain
show service webproxy append-domain
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
    webproxy {
        append-domain domain
    }
}
```

---

### Parameters

<i>domain</i>	Optional. The domain name to be appended to the domain name within URLs.
---------------	--

---

### Default

None.

---

### Usage Guidelines

Use this command to specify a domain name to be appended to domain names within URLs that do not include a dot (“.”). For example, if the domain to be appended is set to “vyatta.com” and the URL received is “www/xyz.htm,” the system renders the final URL as “www.vyatta.com/xyz.htm.”

Use the **set** form of this command to specify a domain name to be appended.

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

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

## service webproxy default-port <port>

Sets the default port for web proxy listen addresses.

---

### Syntax

```
set service webproxy default-port port
delete service webproxy default-port
show service webproxy default-port
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    default-port port
  }
}
```

---

### Parameters

<i>port</i>	Optional. The port number to use for the web proxy service. The range is 1025 to 65535. The default is 3128.
-------------	--

---

### Default

Port 3128 is used for web proxy listen addresses.

---

### Usage Guidelines

Use this command to specify the port on which the web proxy service is to listen for web requests from clients. This port is used by default on web proxy listen addresses.

Use the **set** form of this command to specify the default listening port.

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

Use the **show** form of this command to view web proxy listening port configuration.

## service webproxy domain-block <domain>

Specifies a domain to block.

---

### Syntax

```
set service webproxy domain-block domain
delete service webproxy domain-block domain
show service webproxy domain-block
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    domain-block domain
  }
}
```

---

### Parameters

<i>domain</i>	Multi-node. The domain to block.
---------------	----------------------------------

---

### Default

None

---

### Usage Guidelines

Use this command to block access to the specified domain. For example, specifying “facebook.com” will block all access to facebook.com, and specifying “.cn” will block all access to Chinese sites.

Use the **set** form of this command to specify the domain to block.

Use the **delete** form of this command to restore access to the domain.

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

**NOTE** This command has limitations filtering HTTPS traffic in transparent mode. For details, see [“Transparent Mode for HTTPS Traffic” on page 79](#).

## service webproxy domain-noncache <domain>

Specifies a domain that is not to be cached.

---

### Syntax

```
set service webproxy domain-noncache domain
delete service webproxy domain-noncache domain
show service webproxy domain-noncache
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    domain-noncache domain
  }
}
```

---

### Parameters

<i>domain</i>	Multi-node. The domain that is not to be cached.
---------------	--

---

### Default

All domains are cached.

---

### Usage Guidelines

Use this command to allow access to sites in a domain without caching them. For example, specifying “facebook.com” will allow access to “facebook.com” but the pages accessed will not be cached. This is useful when data on certain sites is sensitive and it caching it on the Vyatta system’s disk poses a security risk. It is also useful for working around problems with “If-Modified-Since” checking at certain sites.

Use the **set** form of this command to specify the domain that should not be cached.

Use the **delete** form of this command to restore caching of sites on the domain.

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



## service webproxy listen-address <ipv4>

Specifies a web proxy listening address.

---

### Syntax

```
set service webproxy listen-address ipv4
delete service webproxy listen-address ipv4
show service webproxy listen-address ipv4
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    listen-address ipv4 {
    }
  }
}
```

---

### Parameters

<i>ipv4</i>	Multi-node. The IP address the web proxy service listens on.  You can set the system to listen for client web requests at more than one IP address by creating multiple <b>listen-address</b> configuration nodes.
-------------	--

---

### Default

None.

---

### Usage Guidelines

Use this command to specify the web proxy listen address. This must be the primary address on whatever interface it is configured on.

The listen address is the IP address on which the web proxy service listens for client web requests. For security, a listen address should only be used on internal/trusted networks, since a proxy can be used to hide the client's true IP address.

Use the **set** form of this command to set a specific listening address for the web proxy service.

Use the **delete** form of this command to restore the web proxy's use of the configured default listen address.

Use the **show** form of this command to view web proxy listen address configuration.

## service webproxy listen-address <ipv4> disable-transparent

Disables web proxy transparent mode at a listening address.

---

### Syntax

```
set service webproxy listen-address ipv4 disable-transparent
delete service webproxy listen-address ipv4 disable-transparent
show service webproxy listen-address ipv4
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    listen-address ipv4 {
      disable-transparent
    }
  }
}
```

---

### Parameters

<i>ipv4</i>	The IP address on which the web proxy service is listening for web requests.
<b>disable-transparent</b>	Disables transparent mode.

---

### Default

Transparent mode is enabled.

---

### Usage Guidelines

Use this command to disable web proxy transparent mode for the specified listen address.

In transparent mode, all traffic arriving on port 80 and destined for the Internet is automatically forwarded through the web proxy. This allows immediate proxy forwarding without configuring client browsers.

Non-transparent proxying requires that the client browsers be configured with the proxy settings before requests are redirected. The advantage of this is that the client web browser can detect that a proxy is in use and can behave accordingly. In addition, web-transmitted malware can sometimes be blocked by a non-transparent web proxy, since they are not aware of the proxy settings.

Use the **set** form of this command to disable web proxy transparent mode for the specified listening address.

Use the **delete** form of this command to re-enable transparent mode.

Use the **show** form of this command to view the configuration for the specified listening address.

## service webproxy listen-address <ipv4> port <port>

Sets the listening port for a listening address.

---

### Syntax

```
set service webproxy listen-address ipv4 port port
delete service webproxy listen-address ipv4 port
show service webproxy listen-address ipv4 port
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    listen-address ipv4 {
      port port
    }
  }
}
```

---

### Parameters

<i>ipv4</i>	An IP address the web proxy service is listening on.
<i>port</i>	The port on which the web proxy service is to listen. The default is the value configured as the default listening port.

---

### Default

The default listening port is specified using the `service webproxy default-port <port>` command.

---

### Usage Guidelines

Use this command to specify the listening port for a listening address.

By default, the web proxy service listens on the port defined as the default listening port, which is configurable using the **service webproxy default-port <port>** command. This command allows you to override the default listening port for a specific IP address. In general, changing the port that the web proxy listens on is only of use when in non-transparent mode.

Use the **set** form of this command to specify the listening port for a listening address.

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

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

## service webproxy proxy-bypass <address>

Specifies an address or subnet to bypass the webproxy service.

---

### Syntax

```
set service webproxy proxy-bypass address
delete service webproxy proxy-bypass address
show service webproxy proxy-bypass
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
    webproxy {
        proxy-bypass address
    }
}
```

---

### Parameters

<i>address</i>	Multi-node. An IPv4 address or IPv4 subnet to bypass the webproxy.
----------------	--

---

### Default

All addresses and subnets are processed by the webproxy when it is enabled.

---

### Usage Guidelines

Use this command to allow packets destined for a specific address or subnet to bypass the webproxy when it is enabled. When the webproxy is enabled, all traffic bound for port 80 is redirected to port 3128 and is processed by the webproxy. This command allows specific traffic to bypass this redirection.

Proxy bypass applies only if the webproxy is in transparent mode.

Use the **set** form of this command to specify the IPv4 address or subnet that should bypass the webproxy.

Use the **delete** form of this command to remove an IPv4 address or subnet from bypassing the webproxy.

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



**NOTE** *This command has limitations filtering HTTPS traffic in transparent mode. For details, see [“Transparent Mode for HTTPS Traffic” on page 79](#).*

## service webproxy reply-block-mime <mime-type>

Specifies a mime type to block.

---

### Syntax

```
set service webproxy reply-block-mime mime-type
delete service webproxy reply-block-mime mime-type
show service webproxy reply-block-mime mime-type
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    reply-block-mime mime-type {
    }
  }
}
```

---

### Parameters

<i>mime-type</i>	Mime type to block. Mime type are specified in a “type/sub-type” format. For example, the mime type for Quicktime video is “video/quicktime”, the mime type for .pdf files is “application/pdf”, and the mime type for .wav files is “audio/wav”.
------------------	---

---

### Default

None.

---

### Usage Guidelines

Use this command to specify the mime type to block.

Use the **set** form of this command to specify the mime type to block.

Use the **delete** form of this command to allow the mime type.

Use the **show** form of this command to view the mime type.

## service webproxy reply-body-max-size <size>

Specifies the maximum size of a reply body.

---

### Syntax

```
set service webproxy reply-body-max-size size
delete service webproxy reply-body-max-size size
show service webproxy reply-body-max-size
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
    webproxy {
        reply-body-max-size size
    }
}
```

---

### Parameters

<i>size</i>	The maximum size (in K bytes) of a web reply. The range is 0 to 4294967295. The default is unlimited.
-------------	---

---

### Default

All reply sizes are accepted.

---

### Usage Guidelines

Use this command to limit the size of replies accepted. This provides the capability to limit download sizes.

Use the **set** form of this command to specify the maximum reply body size.

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

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

## service webproxy url-filtering disable

Disables web filtering without discarding configuration.

---

### Syntax

```
set service webproxy url-filtering disable
delete service webproxy url-filtering disable
show service webproxy url-filtering
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

None.

---

### Default

None.

---

### Usage Guidelines

Use this command to disable web filtering without discarding configuration.

Use the **set** form of this command to disable web filtering.

Use the **delete** form of this command to enable web filtering.

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

**NOTE** This command has limitations filtering HTTPS traffic in transparent mode. For details, see [“Transparent Mode for HTTPS Traffic” on page 79](#).

## 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 web filtering.

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

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

## service webproxy url-filtering squidguard allow-category <category>

Allows web content by squidGuard database category.

---

### Syntax

```
set service webproxy url-filtering squidguard allow-category category
delete service webproxy url-filtering squidguard allow-category category
show service webproxy url-filtering squidguard allow-category
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>category</i>	Multi-node. The database category to allow, or the keyword <b>all</b> to allow all categories.  You can allow more than one category by creating multiple <b>allow-category</b> configuration nodes.
-----------------	--

---

### Default

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

---

### Usage Guidelines

Use this command to specify database categories to allow.

The categories available will vary with the specific database. To view the categories defined in the installed database, issue [show webproxy blacklist categories command](#).

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

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

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



## service webproxy url-filtering squidguard allow-ipaddr-url

Specifies that direct IP address requests should be allowed.

---

### Syntax

```
set service webproxy url-filtering squidguard allow-ipaddr-url
delete service webproxy url-filtering squidguard allow-ipaddr-url
show service webproxy url-filtering squidguard allow-ipaddr-url
```

---

### Availability

Vyatta Plus - VyattaGuard

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

None.

---

### Default

Direct IP address requests are blocked.

---

### Usage Guidelines

By default, all accesses made directly to an IP address are blocked. Use this command to specify that direct IP address requests should not be blocked.

Use the **set** form of this command to allow direct IP address requests.

Use the **delete** form of this command to restore the default and block direct IP address requests.

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

## service webproxy url-filtering squidguard auto-update update-hour <hour>

Sets the hour of the day at which to check for squidGuard database updates.

---

### Syntax

```
set service webproxy url-filtering squidguard auto-update update-hour hour
delete service webproxy url-filtering squidguard auto-update update-hour
show service webproxy url-filtering squidguard auto-update update-hour
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>hour</i>	The hour of the day (using a 24 hour clock) at which the web proxy service will check for database updates. Possible values range from 0 (12:00am) to 23 (11:00pm).
-------------	---

---

---

### Default

The system will not check for database updates.

### Usage Guidelines

Use this command to specify the hour of the day at which the system should check for database updates.

Use the **set** form of this command to set the hour of the day to check for database updates.

Use the **delete** form of this command to stop the system from checking for updates.

Use the **show** form of this command to view update hour 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 category
      }
    }
  }
}
```

---

### Parameters

<i>category</i>	Multi-node. The database category to block, or the keyword <b>all</b> to block all categories.  You can block more than one category by creating multiple <b>block-category</b> 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 `show webproxy blacklist categories command`.

Use the **set** form of this command to block 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 default-action <action>

Specifies the default action to take for all traffic passing through the webproxy.

---

### Syntax

```
set service webproxy url-filtering squidguard default-action action
delete service webproxy url-filtering squidguard default-action
show service webproxy url-filtering squidguard default-action
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>action</i>	The default action to take on all traffic passing through the webproxy. <b>allow:</b> Allow all traffic through by default. <b>block:</b> Block all traffic by default.
---------------	---

---

---

### Default

Traffic is allowed through the webproxy.

---

### Usage Guidelines

Use this command to specify the default action to take on traffic passing through the webproxy.

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

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

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



## service webproxy url-filtering squidguard enable-safe-search

Enables Safe Search on many popular search engines.

---

### Syntax

```
set service webproxy url-filtering squidguard enable-safe-search
delete service webproxy url-filtering squidguard enable-safe-search
show service webproxy url-filtering squidguard
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        enable-safe-search
      }
    }
  }
}
```

---

### Parameters

None.

---

### Default

Safe Search is not enabled on any search engines.

---

### Usage Guidelines

Use this command to modify requests to many popular search engines to perform Safe Search in order to filter out objectionable content. The search engines that are currently supported include: Google, Yahoo, MSN, and Bing.

Use the set form of this command to enable Safe Search on many popular search engines.

Use the **delete** form of this command to return URL filtering to its default (non-Safe Search) behavior.

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

**NOTE** *This command has limitations filtering HTTPS traffic in transparent mode. For details, see [“Transparent Mode for HTTPS Traffic” on page 79](#).*

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

Specifies an IP address or domain 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 address
      }
    }
  }
}
```

---

### Parameters

<i>address</i>	Multi-node. An IP address or domain to be blocked. Do not prefix the domain name with http://.
	You can block a number of IP addresses and/or domains by creating multiple <b>local-block</b> configuration nodes.

---

### Default

None.

---

### Usage Guidelines

Use this command to specify an IP address or domain 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 domain.

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

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

## service webproxy url-filtering squidguard local-block-keyword <keyword>

Specifies a URL substring within a URL to be blocked.

---

### Syntax

```
set service webproxy url-filtering squidguard local-block-keyword keyword
delete service webproxy url-filtering squidguard local-block-keyword keyword
show service webproxy url-filtering squidguard local-block-keyword
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>keyword</i>	Multi-node. A substring or regular expression (regex) matching a URL to be blocked.  You can block a number of URLs by creating multiple <b>local-block-keyword</b> configuration nodes.
----------------	--

---

---

### Default

None.

---

### Usage Guidelines

Use this command to specify a substring or regular expression matching a URL to be blocked. This allows you to block sites not belonging to a database category.

**NOTE** Use this command with caution as specifying a non-specific substring can match unintended URLs. In addition, this command is CPU intensive and can degrade performance.

Use the **set** form of this command to specify the substring or regular expression to match.

Use the **delete** form of this command to remove the substring or regular expression from the configuration.

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

**NOTE** This command has limitations filtering HTTPS traffic in transparent mode. For details, see [“Transparent Mode for HTTPS Traffic” on page 79](#).

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

Specifies a URL to be blocked.

---

### Syntax

```
set service webproxy url-filtering squidguard local-block-url url
delete service webproxy url-filtering squidguard local-block-url url
show service webproxy url-filtering squidguard local-block-url
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>url</i>	Multi-node. A URL to be blocked. Do not prefix the URL with http://.
	You can block a number of URLs by creating multiple <b>local-block-url</b> configuration nodes.

---

### Default

None.

---

### Usage Guidelines

Use this command to specify a URL to be blocked. This allows you to block sites not belonging to a database category. It differs from **local-block** in that **local-block** only accepts an IP address or a top-level domain.

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

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

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

**NOTE** *This command has limitations filtering HTTPS traffic in transparent mode. For details, see “Transparent Mode for HTTPS Traffic” on page 79.*



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

Specifies an IP address or domain 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 address
      }
    }
  }
}
```

---

### Parameters

<i>address</i>	Multi-node. An IP address or domain to allow. Do not prefix a domain name with http://.  You can allow a number of IP address or domains by creating multiple <b>local-ok</b> configuration nodes.
----------------	--

---

---

### Default

None.

---

### Usage Guidelines

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

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

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

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

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

Specifies a URL to be allowed.

---

### Syntax

```
set service webproxy url-filtering squidguard local-ok-url url
delete service webproxy url-filtering squidguard local-ok-url url
show service webproxy url-filtering squidguard local-ok-url
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>url</i>	Multi-node. A URL to be allowed. Do not prefix the URL with <code>http://</code> .  You can allow a number of URLs by creating multiple <code>local-ok-url</code> configuration nodes.
------------	--

---

---

### Default

None.

---

### Usage Guidelines

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

Use the **set** form of this command to allow a specific URL.

Use the **delete** form of this command to stop a URL from being allowed.

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

**NOTE** *This command has limitations filtering HTTPS traffic in transparent mode. For details, see [“Transparent Mode for HTTPS Traffic” on page 79](#).*

## 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 category
      }
    }
  }
}
```

---

### Parameters

<i>category</i>	Multi-node. The squidGuard database category to log, or the keyword <b>all</b> to log all categories.
-----------------	---

---

### Default

Web proxy web 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 url
      }
    }
  }
}
```

---

### Parameters

<i>url</i>	The URL to which to redirect users when the user attempts to access a blacklisted URL. By default, users are redirected to a pre-defined site.
------------	--

---

### Default

Users attempting to access a blacklisted site are redirected to a pre-defined site.

---

### Usage Guidelines

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

**NOTE** It is important to make sure that the redirect URL specified is not a blocked site. For example, if the **default-action** is set to **block** and the **redirect-url** is not included in the **local-ok** list then it will not be able to redirect the user as expected.

**NOTE** In order to enter a “?” in the URL string, it must be preceded by a <Ctrl>+v or it will be interpreted as a command completion character. Another option is to use the “set terminal key query-help disable” command.

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.

**NOTE** This command has limitations filtering HTTPS traffic in transparent mode. For details, see [“Transparent Mode for HTTPS Traffic” on page 79](#).



## service webproxy url-filtering squidguard rule <rule-num>

Specifies a web filtering rule.

---

### Syntax

```
set service webproxy url-filtering squidguard rule rule-num
delete service webproxy url-filtering squidguard rule rule-num
show service webproxy url-filtering squidguard rule rule-num
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>rule-num</i>	Multi-node. Defines a web filtering rule. The rule number specifies the order in which the rule is evaluated. Each rule must have a unique rule number. The range is 1 to 1024.
-----------------	---

---

---

### Default

None.

---

### Usage Guidelines

Use this command to define a web filtering rule. web filtering rules are evaluated in a sequence according to rule number.

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

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

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

## service webproxy url-filtering squidguard rule <rule-num> allow-category <category>

Allows web content by squidGuard database category within the rule.

---

### Syntax

```
set service webproxy url-filtering squidguard rule rule-num allow-category category
delete service webproxy url-filtering squidguard rule rule-num allow-category
category
show service webproxy url-filtering squidguard rule rule-num allow-category
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        rule rule-num {
          allow-category category
        }
      }
    }
  }
}
```

---

### Parameters

<i>rule-num</i>	Multi-node. Defines a web filtering rule. The rule number specifies the order in which the rule is evaluated. Each rule must have a unique rule number. The range is 1 to 1024.
<i>category</i>	Multi-node. The database category to allow, or the keyword <b>all</b> to allow all categories.  You can allow more than one category by creating multiple <b>allow-category</b> configuration nodes.

---

### Default

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

---

### Usage Guidelines

Use this command to specify database categories to allow within the rule.

The categories available will vary with the specific database. To view the categories defined in the installed database, issue [show webproxy blacklist categories command](#).

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

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

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

## service webproxy url-filtering squidguard rule <rule-num> allow-ipaddr-url

Specifies that direct IP address requests should be allowed within the rule.

---

### Syntax

```
set service webproxy url-filtering squidguard rule rule-num allow-ipaddr-url
delete service webproxy url-filtering squidguard rule rule-num allow-ipaddr-url
show service webproxy url-filtering squidguard rule rule-num allow-ipaddr-url
```

---

### Availability

Vyatta Plus - VyattaGuard

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        rule rule-num {
          allow-ipaddr-url
        }
      }
    }
  }
}
```

---

### Parameters

---

<i>rule-num</i>	Multi-node. Defines a web filtering rule. The rule number specifies the order in which the rule is evaluated. Each rule must have a unique rule number. The range is 1 to 1024.
-----------------	---

---

---

### Default

Direct IP address requests are blocked.

---

### Usage Guidelines

Use this command to specify that direct IP address requests should not be blocked. By default, all accesses made directly to an IP address are blocked.

Use the **set** form of this command to allow direct IP address requests.

Use the **delete** form of this command to restore the default and block direct IP address requests.

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

## service webproxy url-filtering squidguard rule <rule-num> block-category <category>

Blocks web content by squidGuard database category within the rule.

---

### Syntax

```
set service webproxy url-filtering squidguard rule rule-num block-category category
delete service webproxy url-filtering squidguard rule rule-num block-category
category
show service webproxy url-filtering squidguard rule rule-num block-category
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        rule rule-num {
          block-category category
        }
      }
    }
  }
}
```

---

### Parameters

<i>rule-num</i>	Multi-node. Defines a web filtering rule. The rule number specifies the order in which the rule is evaluated. Each rule must have a unique rule number. The range is 1 to 1024.
<i>category</i>	Multi-node. The database category to block, or the keyword <b>all</b> to block all categories.  You can block more than one category by creating multiple <b>block-category</b> configuration nodes.

---

### Default

When the **rule** is defined with no block categories, all categories are blocked within the rule.

---

### Usage Guidelines

Use this command to specify database categories to block within the rule.

The categories available will vary with the specific database. To view the categories defined in the installed database, issue [show webproxy blacklist categories command](#).

Use the **set** form of this command to block 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 rule <rule-num> default-action <action>

Specifies the default action to take for traffic within the rule.

---

### Syntax

```
set service webproxy url-filtering squidguard rule rule-num default-action action
delete service webproxy url-filtering squidguard rule rule-num default-action
show service webproxy url-filtering squidguard rule rule-num default-action
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        rule rule-num {
          default-action action
        }
      }
    }
  }
}
```

---

### Parameters

<i>rule-num</i>	Multi-node. Defines a web filtering rule. The rule number specifies the order in which the rule is evaluated. Each rule must have a unique rule number. The range is 1 to 1024.
<i>action</i>	The default action to take on all traffic passing through the webproxy. <b>allow:</b> Allow all traffic through by default. <b>block:</b> Block all traffic by default.

---

---

### Default

Traffic is allowed through the webproxy.

---

### Usage Guidelines

Use this command to specify the default action to take on traffic within the rule.

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

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

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

## service webproxy url-filtering squidguard rule <rule-num> description <desc>

Specifies a brief description for a web filtering policy rule.

---

### Syntax

```
set service webproxy url-filtering squidguard rule rule-num description desc
delete service webproxy url-filtering squidguard rule rule-num description
show service webproxy url-filtering squidguard rule rule-num description
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        rule rule-num {
          description desc
        }
      }
    }
  }
}
```

---

### Parameters

<i>rule-num</i>	Multi-node. Defines a web filtering rule. The rule number specifies the order in which the rule is evaluated. Each rule must have a unique rule number. The range is 1 to 1024.
<i>desc</i>	A description of the policy rule. If the description contains spaces then it must be contained in double quotes.

---

---

### Default

None.

---

### Usage Guidelines

Use this command to specify a description of a web filtering policy rule.

Use the **set** form of this command to specify the description of the policy rule specified by the policy rule number.

Use the **delete** form of this command to remove the description of the policy rule specified by the policy rule number.

Use the **show** form of this command to view the description of the policy rule specified by the policy rule number.

## service webproxy url-filtering squidguard rule <rule-num> enable-safe-search

Enables Safe Search on many popular search engines for a web filtering policy rule.

---

### Syntax

```
set service webproxy url-filtering squidguard rule rule-num enable-safe-search
delete service webproxy url-filtering squidguard rule rule-num enable-safe-search
show service webproxy url-filtering squidguard rule rule-num
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        rule rule-num {
          enable-safe-search
        }
      }
    }
  }
}
```

---

### Parameters

<i>rule-num</i>	Multi-node. Defines a web filtering rule. The rule number specifies the order in which the rule is evaluated. Each rule must have a unique rule number. The range is 1 to 1024.
-----------------	---

---

---

### Default

Safe Search is not enabled on any search engines.

---

### Usage Guidelines

Use this command to modify requests to many popular search engines to perform Safe Search in order to filter out objectionable content for this policy rule. The search engines that are currently supported include: Google, Yahoo, MSN, and Bing.

Use the **set** form of this command to enable Safe Search on many popular search engines for this policy rule.

Use the **delete** form of this command to return URL filtering for this policy rule to its default (non-Safe Search) behavior.

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

**NOTE** *This command has limitations filtering HTTPS traffic in transparent mode. For details, see [“Transparent Mode for HTTPS Traffic” on page 79](#).*

## service webproxy url-filtering squidguard rule <rule-num> local-block <address>

Defines a specific IP address or URL to be blocked within the rule.

---

### Syntax

```
set service webproxy url-filtering squidguard rule rule-num local-block address
delete service webproxy url-filtering squidguard rule rule-num local-block address
show service webproxy url-filtering squidguard rule rule-num local-block
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        rule rule-num {
          local-block address
        }
      }
    }
  }
}
```

---

### Parameters

<i>rule-num</i>	Multi-node. Defines a web filtering rule. The rule number specifies the order in which the rule is evaluated. Each rule must have a unique rule number. The range is 1 to 1024.
<i>address</i>	Multi-node. An IP address or URL to be blocked within the rule.  You can block a number of IP addresses and/or URLs by creating multiple <b>local-block</b> configuration nodes.

---

---

**Default**

None.

---

**Usage Guidelines**

Use this command to specify an IP address or URL to be blocked within the rule. 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 rule <rule-num> local-block-keyword <keyword>

Defines a URL substring within a URL to be blocked within the rule.

---

### Syntax

```
set service webproxy url-filtering squidguard rule rule-num local-block-keyword  
keyword
```

```
delete service webproxy url-filtering squidguard rule rule-num local-block-keyword  
keyword
```

```
show service webproxy url-filtering squidguard rule rule-num local-block-keyword
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {  
  webproxy {  
    url-filtering {  
      squidguard {  
        rule rule-num {  
          local-block-keyword keyword  
        }  
      }  
    }  
  }  
}
```

---

### Parameters

---

<i>rule-num</i>	Multi-node. Defines a web filtering rule. The rule number specifies the order in which the rule is evaluated. Each rule must have a unique rule number. The range is 1 to 1024.
<i>keyword</i>	Multi-node. A substring or regular expression (regex) matching a URL to be blocked within the rule.  You can block a number of URLs by creating multiple <code>local-block-keyword</code> configuration nodes.

---

---

## Default

None.

---

## Usage Guidelines

Use this command to specify a substring or regular expression matching a URL to be blocked within the rule. This allows you to block sites not belonging to a database category.

**NOTE** Use this command with caution as specifying a non-specific substring can match unintended URLs. In addition, this command is CPU intensive and can degrade performance.

Use the **set** form of this command to specify the substring or regular expression to match..

Use the **delete** form of this command to remove the substring or regular expression from the configuration.

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

**NOTE** This command has limitations filtering HTTPS traffic in transparent mode. For details, see [“Transparent Mode for HTTPS Traffic” on page 79](#).

## service webproxy url-filtering squidguard rule <rule-num> local-block-url <url>

Specifies a URL to be blocked within the rule.

---

### Syntax

```
set service webproxy url-filtering squidguard rule rule-num local-block-url url
delete service webproxy url-filtering squidguard rule rule-num local-block-url url
show service webproxy url-filtering squidguard rule rule-num local-block-url
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        rule rule-num {
          local-block-url url
        }
      }
    }
  }
}
```

---

### Parameters

<i>rule-num</i>	Multi-node. Defines a web filtering rule. The rule number specifies the order in which the rule is evaluated. Each rule must have a unique rule number. The range is 1 to 1024.
<i>url</i>	Multi-node. A URL to be blocked. Do not prefix the URL with http://.  You can block a number of URLs by creating multiple <b>local-block-url</b> configuration nodes.

---

---

### Default

None.

---

### Usage Guidelines

Use this command to specify a URL to be blocked within the rule. This allows you to block sites not belonging to a database category. It differs from **local-block** in that **local-block** only accepts an IP address or a top-level domain.

Use the **set** form of this command to block a specific URL within the rule.

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

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

**NOTE** This command has limitations filtering HTTPS traffic in transparent mode. For details, see [“Transparent Mode for HTTPS Traffic” on page 79](#).

## service webproxy url-filtering squidguard rule <rule-num> local-ok <address>

Specifies an IP address or URL to allow within the rule.

---

### Syntax

```
set service webproxy url-filtering squidguard rule rule-num local-ok address
delete service webproxy url-filtering squidguard rule rule-num local-ok address
show service webproxy url-filtering squidguard rule rule-num local-ok
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        rule rule-num {
          local-ok address
        }
      }
    }
  }
}
```

---

### Parameters

<i>rule-num</i>	Multi-node. Defines a web filtering rule. The rule number specifies the order in which the rule is evaluated. Each rule must have a unique rule number. The range is 1 to 1024.
<i>address</i>	Multi-node. An IP address or URL to allow within the rule.

---

---

### Default

None.

---

### Usage Guidelines

Use this command to allow an IP address or URL (within the rule) 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 rule <rule-num> local-ok-url <url>

Specifies a URL to be allowed within the rule.

---

### Syntax

```
set service webproxy url-filtering squidguard rule rule-num local-ok-url url
delete service webproxy url-filtering squidguard rule rule-num local-ok-url url
show service webproxy url-filtering squidguard rule rule-num local-ok-url
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        rule rule-num {
          local-ok-url url
        }
      }
    }
  }
}
```

---

### Parameters

<i>rule-num</i>	Multi-node. Defines a web filtering rule. The rule number specifies the order in which the rule is evaluated. Each rule must have a unique rule number. The range is 1 to 1024.
<i>url</i>	Multi-node. A URL to be allowed. Do not prefix the URL with http://.  You can allow a number of URLs by creating multiple <b>local-ok-url</b> configuration nodes.

---

---

### Default

None.

---

### Usage Guidelines

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

Use the **set** form of this command to allow a specific URL within the rule.

Use the **delete** form of this command to stop a URL from being allowed.

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

**NOTE** *This command has limitations filtering HTTPS traffic in transparent mode. For details, see “Transparent Mode for HTTPS Traffic” on page 79.*



## service webproxy url-filtering squidguard rule <rule-num> log <category>

Enables logging for a squidGuard database category within the rule.

---

### Syntax

```
set service webproxy url-filtering squidguard rule rule-num log category
delete service webproxy url-filtering squidguard rule rule-num log category
show service webproxy url-filtering squidguard rule rule-num log
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        rule rule-num {
          log category
        }
      }
    }
  }
}
```

---

### Parameters

<i>rule-num</i>	Multi-node. Defines a web filtering rule. The rule number specifies the order in which the rule is evaluated. Each rule must have a unique rule number. The range is 1 to 1024.
<i>category</i>	Multi-node. The squidGuard database category to log, or the keyword <b>all</b> to log all categories within the rule.

---

---

### Default

Web proxy web filtering is not logged.

---

### Usage Guidelines

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

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 rule <rule-num> redirect-url <url>

Specifies a URL to redirect users to when a blacklisted URL is requested within the rule.

---

### Syntax

```
set service webproxy url-filtering squidguard rule rule-num redirect-url url
delete service webproxy url-filtering squidguard rule rule-num redirect-url
show service webproxy url-filtering squidguard rule rule-num redirect-url
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        rule rule-num {
          redirect-url url
        }
      }
    }
  }
}
```

---

### Parameters

<i>rule-num</i>	Multi-node. Defines a web filtering rule. The rule number specifies the order in which the rule is evaluated. Each rule must have a unique rule number. The range is 1 to 1024.
<i>url</i>	The URL to which to redirect users when the user attempts to access a blacklisted URL.

---

---

## Default

Users attempting to access a blacklisted site are redirected to the global redirect URL.

---

## Usage Guidelines

Use this command to specify a redirect URL for users attempting to access a filtered URL within the rule. If no redirect URL is specified within the rule then the global redirect URL is used.

**NOTE** *It is important to make sure that the redirect URL specified is not a blocked site. For example, if the **default-action** is set to **block** and the **redirect-url** is not included in the **local-ok** list then it will not be able to redirect the user as expected.*

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.

**NOTE** *This command has limitations filtering HTTPS traffic in transparent mode. For details, see [“Transparent Mode for HTTPS Traffic” on page 79](#).*

## service webproxy url-filtering squidguard rule <rule-num> source-group <group-name>

Specifies the source group to be used for the web filtering rule.

---

### Syntax

```
set service webproxy url-filtering squidguard rule rule-num source-group  
group-name  
delete service webproxy url-filtering squidguard rule rule-num source-group  
show service webproxy url-filtering squidguard rule rule-num source-group
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {  
  webproxy {  
    url-filtering {  
      squidguard {  
        rule rule-num {  
          source-group group-name  
        }  
      }  
    }  
  }  
}
```

---

### Parameters

<i>rule-num</i>	Multi-node. Defines a web filtering rule. The rule number specifies the order in which the rule is evaluated. Each rule must have a unique rule number. The range is 1 to 1024.
<i>group-name</i>	Mandatory. The source group to be used for the web filtering rule.

---

---

### Default

None.

---

### Usage Guidelines

Use this command to specify the source group to be used for the web filtering rule. A source group must be specified. Source groups must be pre-defined using [service webproxy url-filtering squidguard source-group <group-name> command](#) before they can be specified here.

Use the **set** form of this command to specify the source group to use for the web filtering rule.

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

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

## service webproxy url-filtering squidguard rule <rule-num> time-period <period-name>

Specifies the time period to be used for the web filtering rule.

---

### Syntax

```
set service webproxy url-filtering squidguard rule rule-num time-period period-name
delete service webproxy url-filtering squidguard rule rule-num time-period
show service webproxy url-filtering squidguard rule rule-num time-period
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        rule rule-num {
          time-period period-name
        }
      }
    }
  }
}
```

---

### Parameters

<i>rule-num</i>	Multi-node. Defines a web filtering rule. The rule number specifies the order in which the rule is evaluated. Each rule must have a unique rule number. The range is 1 to 1024.
<i>period-name</i>	The time period to be used for the web filtering rule.

---

### Default

The web filtering rule is valid at all times.

---

### Usage Guidelines

Use this command to specify the time period to be used for the web filtering rule. Time periods must be pre-defined using `service webproxy url-filtering squidguard time-period <period-name> command` before they can be specified here. Use “!” to negate the time period (i.e. include all times not specified in the time period definition).

Use the **set** form of this command to specify the time period to use for the web filtering rule.

Use the **delete** form of this command to remove the time period and make the web filtering rule valid at all times.

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



## service webproxy url-filtering squidguard source-group <group-name>

Specifies a web filtering source group.

---

### Syntax

```
set service webproxy url-filtering squidguard source-group group-name
delete service webproxy url-filtering squidguard source-group group-name
show service webproxy url-filtering squidguard source-group group-name
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        source-group group-name {
        }
      }
    }
  }
}
```

---

### Parameters

<i>group-name</i>	Multi-node. A web filtering source group.
-------------------	---

---

### Default

None.

---

### Usage Guidelines

Use this command to define a web filtering source group to provide a way to filter traffic from a specific set of addresses or subnets rather than filtering all traffic.

Use the **set** form of this command to specify a web filtering source group.

Use the **delete** form of this command to remove the web filtering source group.

Use the **show** form of this command to view the web filtering source group configuration.

## service webproxy url-filtering squidguard source-group <group-name> address <addr>

Specifies an IPv4 address or subnet to include in the web filtering source group.

---

### Syntax

```
set service webproxy url-filtering squidguard source-group group-name address addr
delete service webproxy url-filtering squidguard source-group group-name address
addr
show service webproxy url-filtering squidguard source-group group-name address
addr
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        source-group group-name {
          address addr
        }
      }
    }
  }
}
```

---

### Parameters

<i>group-name</i>	Multi-node. A web filtering source group.
<i>addr</i>	Multi-node. An IPv4 address or subnet that is part of the source group.

---

### Default

None.

---

### Usage Guidelines

Use this command to specify an IPv4 address or subnet to include in the source group.

Use the **set** form of this command to specify an IPv4 address or subnet.

Use the **delete** form of this command to remove the IPv4 address or subnet from the source group.

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

## service webproxy url-filtering squidguard source-group <group-name> description <desc>

Specifies a brief description for a web filtering source group.

---

### Syntax

```
set service webproxy url-filtering squidguard source-group group-name description  
desc
```

```
delete service webproxy url-filtering squidguard source-group group-name  
description
```

```
show service webproxy url-filtering squidguard source-group group-name  
description
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {  
  webproxy {  
    url-filtering {  
      squidguard {  
        source-group group-name {  
          description desc  
        }  
      }  
    }  
  }  
}
```

---

### Parameters

---

<i>group-name</i>	Multi-node. A web filtering source group.
<i>desc</i>	A description of the source group. If the description contains spaces then it must be contained in double quotes.

---

---

### Default

None.

---

### Usage Guidelines

Use this command to specify a description of a web filtering source group.

Use the **set** form of this command to specify a description of a web filtering source group.

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

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

## service webproxy url-filtering squidguard source-group <group-name> domain <domain>

Specifies a domain to include in a web filtering source group.

---

### Syntax

```
set service webproxy url-filtering squidguard source-group group-name domain domain
```

```
delete service webproxy url-filtering squidguard source-group group-name domain domain
```

```
show service webproxy url-filtering squidguard source-group group-name domain
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {  
  webproxy {  
    url-filtering {  
      squidguard {  
        source-group group-name {  
          domain domain  
        }  
      }  
    }  
  }  
}
```

---

### Parameters

<i>group-name</i>	Multi-node. A web filtering source group.
<i>domain</i>	A domain to be included in the source group (e.g. company.com).

---

### Default

None.

---

### Usage Guidelines

Use this command to specify a domain to include in a web filtering source group.

Use the **set** form of this command to specify a domain to include in a web filtering source group.

Use the **delete** form of this command to remove the a domain to include in a web filtering source group.

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



## service webproxy url-filtering squidguard time-period <period-name>

Specifies a time period to be used in a web filtering rule.

---

### Syntax

```
set service webproxy url-filtering squidguard time-period period-name
delete service webproxy url-filtering squidguard time-period period-name
show service webproxy url-filtering squidguard time-period period-name
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        time-period period-name {
        }
      }
    }
  }
}
```

---

### Parameters

<i>period-name</i>	The time period to be used in a web filtering rule.
--------------------	---

---

### Default

The rule is active at all times.

---

### Usage Guidelines

Use this command to define a time period to be used in a web filtering rule. The web filtering rule is valid during the times specified.

Use the **set** form of this command to specify a time period to be used in a web filtering rule.

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

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

## service webproxy url-filtering squidguard time-period <period-name> days <day> time <time>

Specifies a day and time included in the time period.

---

### Syntax

```
set service webproxy url-filtering squidguard time-period period-name days day time time
```

```
delete service webproxy url-filtering squidguard time-period period-name days day [time]
```

```
show service webproxy url-filtering squidguard time-period period-name days day [time]
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {  
  webproxy {  
    url-filtering {  
      squidguard {  
        time-period period-name {  
          days day {  
            time time  
          }  
        }  
      }  
    }  
  }  
}
```

---

### Parameters

---

<i>period-name</i>	The time period to be used in a web filtering rule.
--------------------	---

---

---

<i>day</i>	<p>A day (or days) within a time period specification. Supported values are as follows:</p> <p><b>Mon:</b> The rule is valid on Mondays.</p> <p><b>Tue:</b> The rule is valid on Tuesdays.</p> <p><b>Wed:</b> The rule is valid on Wednesdays.</p> <p><b>Thu:</b> The rule is valid on Thursdays.</p> <p><b>Fri:</b> The rule is valid on Fridays.</p> <p><b>Sat:</b> The rule is valid on Saturdays.</p> <p><b>weekdays:</b> The rule is valid on weekdays.</p> <p><b>weekends:</b> The rule is valid on weekends.</p> <p><b>all:</b> The rule is valid on all days.</p>
<i>time</i>	<p>The time range (using 24 hour time representation) within the day specified. The format is <b>hh:mm-hh:mm</b>. Multiple ranges are supported. When multiple ranges are specified they must be separated by commas and enclosed in double quotes (e.g. "09:00-14:00, 18:00-24:00").</p>

---

### Default

None.

### Usage Guidelines

Use this command to specify a day (or days) and a time range within the time period definition.

**NOTE** To filter url requests based on time period, at least one rule and one source group are required.

Use the **set** form of this command to specify a day (or days) and a time range.

Use the **delete** form of this command to remove the day and/or time configuration.

Use the **show** form of this command to view the day and/or time configuration.

## service webproxy url-filtering squidguard time-period <period-name> description <desc>

Specifies a brief description for the time period.

---

### Syntax

```
set service webproxy url-filtering squidguard time-period period-name description desc
delete service webproxy url-filtering squidguard time-period period-name description
show service webproxy url-filtering squidguard time-period period-name description
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        time-period period-name {
          description desc
        }
      }
    }
  }
}
```

---

### Parameters

---

<i>period-name</i>	The time period to be used.
<i>desc</i>	A description of the policy rule. If the description contains spaces then it must be contained in double quotes.

---

---

**Default**

None.

---

**Usage Guidelines**

Use this command to specify a description of a time period.

Use the **set** form of this command to specify the description of the time period.

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

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

## service webproxy url-filtering squidguard vyattaguard https-filter

Specifies that transparent mode HTTPS filtering is to be used.

---

### Syntax

```
set service webproxy url-filtering squidguard vyattaguard https-filter
delete service webproxy url-filtering squidguard vyattaguard https-filter
show service webproxy url-filtering squidguard vyattaguard
```

---

### Availability

Vyatta Plus - VyattaGuard

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        vyattaguard {
          https-filter
        }
      }
    }
  }
}
```

---

### Parameters

None.

---

### Default

None.

---

### Usage Guidelines

Use this command to specify that transparent mode HTTPS filtering is to be used. See [“Transparent Mode for HTTPS Traffic” on page 79](#) for more information on using this command.

Use the **set** form of this command to specify that transparent mode HTTPS filtering is to be used.

Use the **delete** form of this command to remove transparent mode HTTPS filtering.

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



## service webproxy url-filtering squidguard vyattaguard mode

Specifies how the VyattaGuard system will operate.

---

### Syntax

```
set service webproxy url-filtering squidguard vyattaguard mode [local-only | net-only | normal]
```

```
delete service webproxy url-filtering squidguard vyattaguard mode
```

```
show service webproxy url-filtering squidguard vyattaguard mode
```

---

### Availability

Vyatta Plus - VyattaGuard

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
service {
  webproxy {
    url-filtering {
      squidguard {
        vyattaguard {
          mode {
            local-only
            net-only
            normal
          }
        }
      }
    }
  }
}
```

---

### Parameters

<b>local-only</b>	Use only the local vyattaguard database to classify a URL.
-------------------	--

---

---

<b>net-only</b>	Use only the remote vyattaguard database to classify a URL.
<b>normal</b>	Use the local vyattaguard database to classify a URL. If no classification can be made, try the remote vyattguard database. This is the default.

---

---

### Default

Both the local database and network classification are used.

---

### Usage Guidelines

Use this command to specify how the vyattaguard system will operate. The **net-only** mode would typically be used on systems with disk space constraints that would not be able to house the entire vyattaguard database. Vyatta recommends 2GB of additional disk space to house the vyattguard database.

Use the **set** form of this command to specify how the vyattaguard system will operate.

Use the **delete** form of this command to return the vyattaguard fetaure to its default operational mode.

Use the **show** form of this command to view the vyattaguard mode 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 2-12](#) displays categories for a squidGuard database.

Example 2-12 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 2-13](#) shows the first few domains displayed from an installed database.

Example 2-13 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 [http [summary] | https [summary] | summary]
```

---

### Command Mode

Operational mode.

---

### Parameters

<b>http</b>	Show only HTTP entries in the blacklist log.
<b>http summary</b>	Show a summary of HTTP entries in the blacklist log.
<b>https</b>	Show only HTTPS entries in the blacklist log.
<b>https summary</b>	Show a summary of HTTPS entries in the blacklist log.
<b>summary</b>	Show a summary of HTTP and HTTPS entries in the blacklist log.

---

### Usage Guidelines

Use this command to display the system's record of URLs that have been filtered. With no optional parameters specified the complete log is displayed.

**NOTE** *Transparent mode HTTPS requests that get dropped will show up multiple times in the log as the client will retry the request multiple times before giving up. This occurs because HTTPS filtering in transparent mode is not able to redirect blocked attempts. It is only able to drop them.*

---

### Examples

[Example 2-14](#) shows sample output of `show webproxy blacklist log`.

Example 2-14 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>
```

---

[Example 2-15](#) shows sample output of `show webproxy blacklist log summary`.

Example 2-15 Displaying the blacklist log summary

---

```
vyatta@R1> show webproxy blacklist log summary
Blocked category                               Count
-----
Social_Networking-1                          284
Finance-1                                     164
local-block-1                                 93
=====
                                              541

Top 10 sites                                   Count
-----
wwwao-prodtx.bankofamerica.com                45
10.1.27.3                                     41
www.wellsfargo.com                            40
171.159.228.173                               39
74.121.135.225                                35
a0.twimg.com                                  33
a1.twimg.com                                  29
www-11-01-ash2.facebook.com                  24
63.135.80.49                                  23
profile.myspace.com                           22
--
Total sites: 33

Top 10 Requestors                             Blocks
-----
172.16.139.10/-                               393
172.16.139.160/-                              148
--
Total users: 2
```

---

## show webproxy blacklist search <filter>

Displays domains and/or URLs matching search text.

---

### Syntax

```
show webproxy blacklist search filter
```

---

### Command Mode

Operational mode.

---

### Parameters

---

<i>filter</i>	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 2-16](#) lists the IP addresses in the installed database that begin with “206.132.42”.

Example 2-16 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 2-17](#) shows the first few entries of sample output of **show webproxy blacklist urls**.

Example 2-17 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/h0tp001man
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 2-18](#) displays a portion of the web proxy log.

Example 2-18 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_page/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
:
:
```

## show webproxy vyattaguard categories

Displays all categories defined in the installed vyattaguard database.

---

### Availability

Vyatta Plus

---

### Syntax

```
show webproxy vyattaguard categories
```

---

### Command Mode

Operational mode.

---

### Parameters

None

---

### Usage Guidelines

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

---

## show webproxy vyattaguard search <filter>

Displays domains and/or URLs in the vyattaguard database matching search text.

---

### Availability

Vyatta Plus

---

### Syntax

```
show webproxy vyattaguard search filter
```

---

### Command Mode

Operational mode.

---

### Parameters

<i>filter</i>	The filter text.
---------------	------------------

---

### Usage Guidelines

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

---

## update webproxy blacklists

Updates the squidGuard database.

---

### Syntax

```
update webproxy blacklists
```

---

### Command Mode

Operational mode.

---

### Parameters

None.

---

### Usage Guidelines

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

---

### Examples

[Example 2-19](#) shows the system interaction for downloading a first squidGuard database.

#### Example 2-19 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...
```

# update webproxy vyattaguard

Updates the vyattaguard database.

---

## Availability

Vyatta Plus

---

## Syntax

```
update webproxy vyattguard [net-only]
```

---

## Command Mode

Operational mode.

---

## Parameters

---

<b>net-only</b>	Update the remote vyattaguard database.
-----------------	---

---

## Usage Guidelines

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

Use the **net-only** option if the system uses the remote vyattaguard database rather than a local copy. This is defined using the [service webproxy url-filtering squidguard vyattaguard mode](#) command.

# Glossary of Acronyms

ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
AMI	Amazon Machine Image
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
AWS	Amazon Web Services
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CCMP	AES in counter mode with CBC-MAC
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6



---

DLCI	data-link connection identifier
DMI	desktop management interface
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
EBS	Amazon Elastic Block Storage
EC2	Amazon Elastic Compute Cloud
EGP	Exterior Gateway Protocol
ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Output
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
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
KVM	Kernel-Based Virtual Machine
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol
LLDP	Link Layer Discovery Protocol
MAC	medium access control
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
P2P	peer-to-peer
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PAT	Port Address Translation

---

---

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
RHEL	Red Hat Enterprise Linux
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
Rx	receive
S3	Amazon Simple Storage Service
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SSH	Secure Shell
SSID	Service Set Identifier
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service

---

---

Tx	transmit
UDP	User Datagram Protocol
vif	virtual interface
VLAN	virtual LAN
VPC	Amazon virtual private cloud
VPN	Virtual Private Network
VRRP	Virtual Router Redundancy Protocol
WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access

---